IJST (2015) 39A1: 69-78
Iranian Journal of Science & Technology

http://ijsts.shirazu.ac.ir

The block LSMR method: a novel efficient algorithm for solving non-symmetric linear systems with multiple right-hand sides

F. Toutounian[1, 2], M. Mojarrab[1, 3]*

[1]Department of Applied Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad, Iran
[2]The Center of Excellence on Modeling and Control Systems, Ferdowsi University of Mashhad, Iran
[3]Department of Mathematics, University of Sistan and Bluchestan, Zahedan, Iran
E-mail: Maryam.Modjarrab@gmail.com

Abstract

It is well known that if the coefficient matrix in a linear system is large and sparse or sometimes not readily available, then iterative solvers may become the only choice. The block solvers are an attractive class of iterative solvers for solving linear systems with multiple right-hand sides. In general, the block solvers are more suitable for dense systems with preconditioner. In this paper, we present a novel block LSMR (least squares minimal residual) algorithm for solving non-symmetric linear systems with multiple right-hand sides. This algorithm is based on the block bidiagonalization and LSMR algorithm and derived by minimizing the 2-norm of each column of normal equation. Then, we give some properties of the new algorithm. In addition, the convergence of the stated algorithm is studied. In practice, we also observe that the Frobenius norm of residual matrix decreases monotonically. Finally, some numerical examples are presented to show the efficiency of the new method in comparison with the traditional LSMR method.

***Keywords**:* LSMR method; bidiagonalization; block methods; iterative methods; multiple right-hand sides

1. Introduction

Many applications require the solution of several sparse systems of equations

$$Ax^{(i)} = b^{(i)}, \quad i = 1,2, \dots, s, \qquad (1)$$

with the same coefficient matrix and different right-hand sides. When all the $b^{(i)}$'s are available simultaneously, Eq. (1) can be written as

$$AX = B, \qquad (2)$$

where A is an $n \times n$ nonsingular and nonsymmetric real matrix, B and X are $n \times s$ rectangular matrices whose columns are $b^{(1)}, b^{(2)}, \dots, b^{(s)}$ and $x^{(1)}, x^{(2)}, \dots, x^{(s)}$, respectively. In practice, s is of moderate size $s \ll n$. Instead of applying an iterative method to each linear system, it is more efficient to use a method for all the systems simultaneously. In the last years, generalizations of the classical Krylov subspace methods have been developed.

One class of solvers for solving the problem (2) is the block solvers which are much more efficient when the matrix A is relatively dense and preconditioners are used. The first block solvers are block conjugate gradient (Bl-CG) algorithm and block biconjugate gradient (Bl-BCG) algorithm proposed in (Oleary, 1980). Variable Bl-CG algorithms for symmetric positive definite problems are implemented on parallel computers (Haase and Reitzinger, 2005; Nikishin and Yeremin, 1995). If the matrix is only symmetric, an adaptive block Lanczos algorithm and a block version of Minres method are devised in (Dai, 2000). For nonsymmetric problems, the Bl-BCG algorithm (Oleary, 1980; Simoncini, 1997), the block generalized minimal residual (Bl-GMRES) algorithm (Darnel et al., 2008; Gu and Cao, 2001; Gutknecht, 2007; Liu and Zhong, 2008; Morgan, 2005; Robbe and Sadkane, 2006; Simoncini and Gallopoulos, 1995; Simoncini and Gallopoulos, 1996; Vital, 1990), the block quasi minimum residual (Bl-QMR) algorithm (Freund and Malhotra, 1997), the block BiCGStab (Bl-BICGSTAB) algorithm (Guennouni et al., 2003), the block Lanczos method (Guennouni et al., 2004) and the block least squares (Bl-LSQR) algorithm (Karimi and Toutounian, 2006) have been developed.

Another class is the global methods, which are based on the use of a global projection process onto a matrix Krylov subspace, including global FOM and GMRES methods (Bellalij et al., 2008; Jbilou et al., 1999), global conjugate gradient type

*Corresponding author

Received: 28 December 2013 / Accepted: 8 October 2014

methods (Salkuyeh, 2006), global BCG and BiCGStab methods (Jbilou et al., 1997; Jbilou et al., 2005), global CGS algorithm (Zhang and Dai, 2008; Zhang et al., 2010), Gl-LSQR algorithm (Toutounian and Karimi, 2006), Gl-BCR and Gl-CRS algorithms (Zhang et al., 2011), global Hessenberg and CMRH methods (Heyouni, 2001;Lin, 2005), global SCD algorithm (Gu and Yang, 2007), and weighted global methods (Heyouni and Essai, 2005). Global methods are more suitable for sparse linear systems (Heyouni and Essai, 2005).

The other class is the seed methods, which consist of selecting a single system as the seed system and generating the corresponding Krylov subspace and then projecting all the residuals of the other linear systems onto the same Krylov subspace to find new approximate solutions as initial approximations. References on this class include (Abdel-Rehim et al., 2008; Chan and Wang, 1997; Joly, 1991; Saad, 1987; Simoncini and Gallopoulos, 1995; Smith et al., 1989; Van Der Vorst, 1987).

In this paper, based on the block bidiagonalization, we derive a simple recurrence formula for generating a sequence of approximations $\{X_k\}$ such that the $\| \mathrm{col}_j(A^T R_k) \|_2$ decreases monotonically, where $R_k = B - AX_k$ and $\mathrm{col}_j(A^T R_k)$ represents the jth column of $A^T R_k$.

Throughout this paper, the following notations are used. 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 $\langle\cdot\rangle_2$ 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 outline of this paper is as follows. In Section 2, we give a quick overview of 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, 2011) 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 recall some fundamental properties of LSMR algorithm (Chin-Lung Fong and Saunders, 2011), which is an iterative method for solving real linear systems of the form $Ax = b$, where A is a nonsymmetric matrix of order n and $x, b \in \mathbb{R}^n$.

LSMR algorithm uses an algorithm of Golub and Kahan (Gloub and Kahan, 1965), which is stated as procedure Bidiag 1 in (Paige and Saunders, 1982), to reduce $(b\, A)$to the upper-diagonal form $(\beta_1 e_1\ B_k)$. The procedure Bidiag 1 can be described as follows.

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

$$\left.\begin{array}{l} \beta_1 u_1 = b, \quad \alpha_1 v_1 = A^T u_1, \\ \beta_{i+1} u_{i+1} = A v_i - \alpha_i u_i, \\ \alpha_{i+1} v_{i+1} = A^T u_{i+1} - \beta_{i+1} v_i, \end{array}\right\} i = 1,2, ... \tag{3}$$

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

$$\begin{array}{l} U_k \equiv [u_1, u_2, \quad \ldots \ , u_k], \\ V_k \equiv [v_1, v_2, \quad \ldots \ , v_k], \\ B_k \equiv \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}, \\ L_{k+1} = (B_k \alpha_{k+1} e_{k+1}), \quad V_{k+1} = (V_k \quad v_{k+1}), \end{array}$$

the recurrence relations (3) may be rewritten as

$$\begin{array}{l} U_{k+1}(\beta_1 e_1) = b, \\ AV_k = U_{k+1} B_k, \\ A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T = V_{k+1} L_{k+1}^T. \\ A^T A V_k = A^T U_{k+1} B_k = V_{k+1} L_{k+1}^T B_k \\ = V_{k+1} \begin{pmatrix} B_k^T \\ \alpha_{k+1} e_{k+1}^T \end{pmatrix} B_k \\ = V_{k+1} \begin{pmatrix} B_k^T B_k \\ \alpha_{k+1} \beta_{k+1} e_k^T \end{pmatrix}. \end{array}$$

This is equivalent to what would be generated by the symmetric Lanczos process with matrix $A^T A$ and starting vector $A^T b$.

Hence, by using the 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, $\min_{y_k} \| A^T r_k \|$. The main steps of the LSMR algorithm can be summarized as follows.

Algorithm 1 LSMR algorithm

```
Set $\beta_1 u_1 = b,\ \alpha_1 v_1 = A^T u_1,\ \overline{\alpha}_1 = \alpha_1,\ \overline{\zeta_1} = \alpha_1 \beta_1,$
$\rho_0 = 1,\ \overline{\rho}_0 = 1,\ \overline{c_0} = 1,\ \overline{s_0} = 0,\ h_1 = v_1,\ \overline{h}_0 = 0,$
$x_0 = 0$
For $k = 1,2, ...,$ until convergence Do:
    $\beta_{k+1} u_{k+1} = A v_k - \alpha_k u_k$
    $\alpha_{k+1} v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k$
    $\rho_k = (\overline{\alpha}_k^2 + \beta_{k+1}^2)^{\frac{1}{2}}$
```

$$
\begin{aligned}
c_k &= \frac{\bar{\alpha}_k}{\rho_k}\\
s_k &= \frac{\beta_{k+1}}{\rho_k}\\
\theta_{k+1} &= s_k\alpha_{k+1}\\
\bar{\alpha}_{k+1} &= c_k\alpha_{k+1}\\
\bar{\theta}_k &= \bar{s}_{k-1}\rho_k\\
\bar{\rho}_k &= ((\bar{c}_{k-1}\rho_k)^2 + \theta_{k+1}^2)^{\frac{1}{2}}\\
\bar{c}_k &= \frac{\bar{c}_{k-1}\rho_k}{\bar{\rho}_k}\\
\bar{s}_k &= \frac{\theta_{k+1}}{\bar{\rho}_k}\\
\zeta_k &= \bar{c}_k\bar{\zeta}_k\\
\bar{\zeta}_{k+1} &= -\bar{s}_k\bar{\zeta}_k\\
\bar{h}_k &= h_k - \left(\frac{\bar{\theta}_k\rho_k}{(\rho_{k-1}\bar{\rho}_{k-1})}\right)\bar{h}_{k-1}\\
x_k &= x_{k-1} + \left(\frac{\zeta_k}{(\rho_k\bar{\rho}_k)}\right)\bar{h}_k\\
h_{k+1} &= v_{k+1} - \left(\frac{\theta_{k+1}}{\rho_k}\right)h_k.
\end{aligned}
$$

If $|\bar{\zeta}_{k+1}|$ is small enough then stop
End Do.

More details about the LSMR algorithm can be found in (Chin-Lung Fong and Saunders, 2011).

3. The block LSMR method

In this section, we propose a new method for solving the linear equation (2). This method is based on the LSMR method. We use the block Bidiag 1 (Karimi and Toutounian, 2006), based on Bidiag 1, for reducing A to the block lower bidiagonal form.

The block Bidiag 1 procedure constructs the sets of the $n \times n$ block vectors $V_1, V_2, \dots$ and U_1, U_2, 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).

$$
\begin{aligned}
&U_1B_1 = B,\quad V_1A_1 = A^TU_1,\\
&\left.\begin{aligned}
U_{i+1}B_{i+1} &= AV_i - U_iA_i^T,\\
V_{i+1}A_{i+1} &= A^TU_{i+1} - V_iB_{i+1}^T,
\end{aligned}\right\} i = 1,2,\dots,k,
\end{aligned}
\tag{4}
$$

where $U_i, V_i \in \mathbb{R}^{n\times s}; B_i, A_i \in \mathbb{R}^{s\times s}$, and $U_1B_1, V_1A_1, U_{i+1}B_{i+1}$, $V_{i+1}A_{i+1}$ are the QR decompositions of the matrices $B, A^TU_1, AV_i - U_iA_i^T$, $A^TU_{i+1} - V_iB_{i+1}^T$, respectively. With the definitions

$$\bar{U}_k \equiv [U_1, \quad U_2, \quad \dots, \quad U_k],$$

$$\bar{V}_k \equiv [V_1, \quad V_2, \quad \dots, \quad V_k],$$

$$
T_k \equiv \begin{bmatrix}
A_1^T & & & \\
B_2 & A_2^T & & \\
 & \ddots & \ddots & \\
 & & B_k & A_k^T\\
 & & & B_{k+1}
\end{bmatrix},
$$

the recurrence relations (4) may be rewritten as:

$$
\begin{aligned}
&\bar{U}_{k+1}E_1B_1 = B,\\
&A\bar{V}_k = \bar{U}_{k+1}T_k,\\
&A^T\bar{U}_{k+1} = \bar{V}_kT_k^T + V_{k+1}A_{k+1}E_{k+1}^T,
\end{aligned}
$$

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

$$\bar{L}_{k+1} \equiv [T_k \quad E_{k+1}A_{k+1}^T],$$

then

$$A^T\bar{U}_{k+1} = \bar{V}_{k+1}\bar{L}_{k+1}^T,$$

$$
\begin{aligned}
&A^TA\bar{V}_k = A^T\bar{U}_{k+1}T_k = \bar{V}_{k+1}\bar{L}_{k+1}^TT_k =\\
&\bar{V}_{k+1}\begin{bmatrix} T_k^T \\ A_{k+1}E_{k+1}^T \end{bmatrix}T_k\\
&= \bar{V}_{k+1}\begin{bmatrix} T_k^TT_k \\ A_{k+1}E_{k+1}^TT_k \end{bmatrix}.
\end{aligned}
\tag{5}
$$

At iteration k we seek an approximate solutionX_k of the form

$$X_k = \bar{V}_kY_k, \tag{6}$$

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

$$
\begin{aligned}
&A^TR_k = A^TB - A^TAX_k\\
&= V_1A_1B_1 - A^TA\bar{V}_kY_k,
\end{aligned}
$$

we have

$$
\begin{aligned}
&A^TR_k = V_1\bar{B}_1 - \bar{V}_{k+1}\begin{bmatrix} T_k^TT_k \\ A_{k+1}E_{k+1}^TT_k \end{bmatrix}Y_k =\\
&\bar{V}_{k+1}(E_1\bar{B}_1 - \begin{bmatrix} T_k^TT_k \\ \bar{B}_{k+1}\bar{E}_k^T \end{bmatrix}Y_k)
\end{aligned}
\tag{7}
$$

where $\bar{E}_k$ is the $ks \times s$ matrix, which is zero except for kth s rows, which are the $s \times s$ identity matrix. Now, we use the QR decomposition (Golub and Van Loan, 1983), where a unitary matrix Q_{k+1} is determined so that

$$Q_{k+1}T_k = \begin{bmatrix} \bar{R}_k \\ 0 \end{bmatrix}, \bar{R}_k = \begin{bmatrix} \rho_1 & \theta_2 & & & \\ & \rho_2 & \theta_3 & & \\ & & \ddots & \ddots & \\ & & & \rho_{k-1} & \theta_k \\ & & & & \rho_k \end{bmatrix} \qquad (8)$$

where ρ_l and θ_l are the s × s matrices. As (Karimi and Toutounian, 2006), the matrix Q_{k+1} is updated from the previous iteration by setting

$$Q_{k+1} = \begin{bmatrix} I_{(k-1)s} & 0 \\ 0 & Q(a_k, b_k, c_k, d_k) \end{bmatrix} \begin{bmatrix} Q_k & 0 \\ 0 & I_s \end{bmatrix},$$

where

$$Q(a_k, b_k, c_k, d_k) = \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix}$$

is an $2s \times 2s$ unitary matrix written as four $s \times s$ blocks. The unitary matrix $Q(a_k, b_k, c_k, d_k)$ is computed such that

$$Q(a_k, b_k, c_k, d_k) \begin{bmatrix} \hat{\hat{\rho}}_k & 0 \\ B_{k+1} & A_{k+1}^T \end{bmatrix} = \begin{bmatrix} \rho_k & \theta_{k+1} \\ 0 & \hat{\hat{\rho}}_{k+1} \end{bmatrix}. \qquad (9)$$

From (7), we have

$$A^T R_k = \bar{V}_{k+1}\left(E_1\bar{B}_1 - \begin{bmatrix} T_k^T T_k \\ \bar{B}_{k+1}\bar{E}_k^T \end{bmatrix} Y_k\right)$$
$$= \bar{V}_{k+1}(E_1\bar{B}_1 - \begin{bmatrix} \bar{R}_k^T \bar{R}_k \\ \bar{B}_{k+1}\bar{E}_k^T \end{bmatrix} Y_k) \qquad (10)$$

If we define $F_k \equiv \bar{R}_k Y_k$, we have

$$A^T R_k = \bar{V}_{k+1}(E_1\bar{B}_1 - \begin{bmatrix} \bar{R}_k^T \\ \bar{B}_{k+1}\bar{E}_k^T \bar{R}_k^{-1} \end{bmatrix} F_k) =$$
$$\bar{V}_{k+1}(E_1\bar{B}_1 - \begin{bmatrix} \bar{R}_k^T \\ A_{k+1}B_{k+1}\rho_k^{-1}\bar{E}_k^T \end{bmatrix} F_k). \qquad (11)$$

On the other hand, from (9), we have $\theta_{k+1} = b_k A_{k+1}^T$ and

$$\begin{bmatrix} \hat{\hat{\rho}}_k & 0 \\ B_{k+1} & A_{k+1}^T \end{bmatrix} = Q^T(a_k, b_k, c_k, d_k) \begin{bmatrix} \rho_k & \theta_{k+1} \\ 0 & \hat{\hat{\rho}}_{k+1} \end{bmatrix}.$$

This implies that $B_{k+1} = b_k^T \rho_k$. So,

$$\theta_{k+1} = \rho_k^{-T} B_{k+1}^T A_{k+1}^T \qquad (12)$$

and

$$A^T R_k = \bar{V}_{k+1}(E_1\bar{B}_1 - \begin{bmatrix} \bar{R}_k^T \\ \theta_{k+1}^T \bar{E}_k^T \end{bmatrix} F_k). \qquad (13)$$

Then we perform a second QR factorization

$$\bar{Q}_{k+1} \begin{bmatrix} \bar{R}_k^T & E_1\bar{B}_1 \\ \theta_{k+1}^T \bar{E}_k^T & 0 \end{bmatrix} = \begin{bmatrix} \bar{\bar{R}}_k & z_k \\ 0 & \bar{\zeta}_{k+1} \end{bmatrix},$$

$$\bar{\bar{R}}_k = \begin{bmatrix} \bar{\rho}_1 & \bar{\theta}_2 & & & \\ & \bar{\rho}_2 & \bar{\theta}_3 & & \\ & & \ddots & \ddots & \\ & & & \bar{\rho}_{k-1} & \bar{\theta}_k \\ & & & & \bar{\rho}_k \end{bmatrix} \qquad (14)$$

where $\bar{\rho}_l$ and $\bar{\theta}_l$ are the s × s matrices. The matrix $\bar{Q}_{k+1}$ is updated from the previous iteration by setting

$$\bar{Q}_{k+1} = \begin{bmatrix} I_{(k-1)s} & 0 \\ 0 & Q(\bar{a}_k, \bar{b}_k, \bar{c}_k, \bar{d}_k) \end{bmatrix} \begin{bmatrix} \bar{Q}_k & 0 \\ 0 & I_s \end{bmatrix} \qquad (15)$$

where

$$Q(\bar{a}_k, \bar{b}_k, \bar{c}_k, \bar{d}_k) = \begin{bmatrix} \bar{a}_k & \bar{b}_k \\ \bar{c}_k & \bar{d}_k \end{bmatrix}$$

is a $2s \times 2s$ unitary matrix written as four $s \times s$ blocks. The unitary matrix $Q(\bar{a}_k, \bar{b}_k, \bar{c}_k, \bar{d}_k)$ is computed such that

$$Q(\bar{a}_k, \bar{b}_k, \bar{c}_k, \bar{d}_k) \begin{bmatrix} \tilde{\bar{\rho}}_k & 0 \\ \theta_{k+1}^T & \rho_{k+1}^T \end{bmatrix} = \begin{bmatrix} \bar{\rho}_k & \bar{\theta}_{k+1} \\ 0 & \tilde{\bar{\rho}}_{k+1} \end{bmatrix}.$$

Combining what we have with (13) gives

$$A^T R_k = \bar{V}_{k+1}\bar{Q}_{k+1}^T(\begin{bmatrix} z_k \\ \bar{\zeta}_{k+1} \end{bmatrix} - \begin{bmatrix} \bar{\bar{R}}_k \\ 0 \end{bmatrix} F_k). \qquad (16)$$

In the block LSMR algorithm we would like to choose $Y_k \in \mathbb{R}^{ks\times s}$ and correspondingly $F_k = \bar{R}_k Y_k$ such that $\|col_j(A^T R_k)\|_2$ is a minimum independent for $j = 1,2, \ldots, s$. Since $\bar{Q}_{k+1}$ is a unitary matrix and $\bar{V}_{k+1}$ is orthonormal, we get

$$\min_{Y_k}\|col_j(A^T R_k)\|$$
$$= \min_{F_k}\left\|col_j(\begin{bmatrix} z_k \\ \bar{\zeta}_{k+1} \end{bmatrix} - \begin{bmatrix} \bar{\bar{R}}_k \\ 0 \end{bmatrix} F_k)\right\|. \qquad (17)$$

The subproblem is solved by choosing F_k from $\bar{\bar{R}}_k F_k = z_k$. Therefore,

$$A^T R_k = \bar{V}_{k+1}\bar{Q}_{k+1}^T \begin{bmatrix} 0 \\ \bar{\zeta}_{k+1} \end{bmatrix}, \qquad (18)$$

and the approximate solution is given by

$$X_k = \bar{V}_k \bar{R}_k^{-1} \bar{\bar{R}}_k^{-1} z_k.$$

Letting

$$\bar{P}_k \equiv \bar{V}_k \bar{R}_k^{-1} \bar{\bar{R}}_k^{-1} \equiv [P_1 \quad P_2 \quad \ldots \quad P_k],$$

then

$$X_k = \bar{P}_k z_k.$$

The n × s matrix P_k, the last block column of $\bar{P}_k$, can be computed from the previous P_{k-1}, P_{k-2} and V_k, by the update

$$P_k = (V_k - P_{k-2}\bar{\theta}_{k-1}\theta_k \\ -P_{k-1}\left(\bar{\rho}_{k-1}\theta_k + \bar{\theta}_k\rho_k\right))(\bar{\rho}_k\rho_k)^{-1}. \qquad (19)$$

Also note that,

$$z_k = \begin{bmatrix} z_{k-1} \\ \zeta_k \end{bmatrix}$$

in which

$$\zeta_k = \bar{a}_k\bar{\zeta}_k.$$

Thus, X_k can be updated at each step, via the relation

$$X_k = X_{k-1} + P_k\zeta_k. \qquad (20)$$

The equation $\overline{RES}_k = \max_j\left\|\mathrm{col}_j(A^T R_k)\right\|_2$ is computed directly from the quantity $\bar{\zeta}_{k+1}$ as $\overline{RES}_k = \max_j\left\|\mathrm{col}_j(\bar{\zeta}_{k+1})\right\|_2$.
From (20), the residual R_k is given by

$$R_k = R_{k-1} - AP_k\zeta_k, \qquad (21)$$

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

$$AP_k = (AV_k - AP_{k-2}\bar{\theta}_{k-1}\theta_k - AP_{k-1}\left(\bar{\rho}_{k-1}\theta_k + \bar{\theta}_k\rho_k\right))(\bar{\rho}_k\rho_k)^{-1}.$$

In addition, we show that the $\|\ \mathrm{col}_j(R_k)\ \|_2$ can be estimated by a simple formula. We transform $\bar{\bar{R}}_k^T$ to block upper-bidiagonal form using a third QR factorization: $\widetilde{R}_k = \widetilde{Q}_k\bar{\bar{R}}_k^T$ with $\widetilde{Q}_k = \widetilde{P}_{k-1}\dots\widetilde{P}_1$. By defining

$$\tilde{F}_k = \widetilde{Q}_k F_k, \qquad \widetilde{B}_k = \begin{bmatrix} \widetilde{Q}_k & \\ & I_s \end{bmatrix} Q_{k+1}E_1B_1, \qquad (22)$$

we have

$$\begin{aligned} R_k &= B - AX_k \\ &= U_1B_1 - A\bar{V}_kY_k \\ &= \bar{U}_{k+1}(E_1B_1 - T_kY_k) \\ &= \bar{U}_{k+1}\left(E_1B_1 - Q_{k+1}^T\begin{bmatrix} \bar{R}_k \\ 0_s \end{bmatrix} Y_k\right) \\ &= \bar{U}_{k+1}\left(E_1B_1 - Q_{k+1}^T\begin{bmatrix} F_k \\ 0_s \end{bmatrix}\right) \\ &= \bar{U}_{k+1}\left(Q_{k+1}^T\begin{bmatrix} \widetilde{Q}_k^T & \\ & I_s \end{bmatrix}\widetilde{B}_k - Q_{k+1}^T\begin{bmatrix} \widetilde{Q}_k^T\tilde{F}_k \\ 0_s \end{bmatrix}\right) \\ &= \bar{U}_{k+1}Q_{k+1}^T\begin{bmatrix} \widetilde{Q}_k^T & \\ & I_s \end{bmatrix}\left(\widetilde{B}_k - \begin{bmatrix} \tilde{F}_k \\ 0_s \end{bmatrix}\right). \end{aligned}$$

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

$$\|\ \mathrm{col}_j(R_k)\ \|_2 = \|\left(\widetilde{B}_k - \begin{bmatrix} \tilde{F}_k \\ 0_s \end{bmatrix}\right) e_j\ \|_2. \qquad (23)$$

The matrices $\widetilde{B}_k$ and $\tilde{F}_k$ can be written in the form

$$\widetilde{B}_k = [\tilde{\beta}_1^T \quad \tilde{\beta}_{k-1}^T\dot{\beta}_k^T\ddot{\beta}_{k+1}^T]^T, \tilde{F}_k = [\tilde{\tau}_1^T \dots \tilde{\tau}_{k-1}^T\dot{\tau}_k^T]^T \qquad (24)$$

The following Lemma shows that we can estimate $\|\ \mathrm{col}_j(R_k)\ \|_2$ from just the last two blocks of $\widetilde{B}_k$ and the last block of $\tilde{F}_k$.

Lemma 1. In (23) and (24), $\tilde{\beta}_i = \tilde{\tau}_i$ for $i = 1,2,\dots,k-1$.

Proof: Appendix A proves the lemma by induction.

The main steps of Bl-LSMR algorithm can be summarized as follows.

Algorithm 2 Bl-LSMR algorithm

```
Set $X_0 = 0$, $\rho_0 = I_s$, $\bar{\rho}_0 = I_s$, $d_0 = I_s$, $\bar{d}_0 = I_s$, $\bar{\theta}_0 = 0$, $\bar{\theta}_1 = 0$, $\theta_1 = 0$
$U_1B_1 = B$, $V_1A_1 = A^TU_1$, (QR decomposition of B and $A^TU_1$)
Set $\bar{B}_1 = A_1B_1$, $\hat{\hat{\rho}}_1 = A_1^T$, $\bar{\zeta}_1 = \bar{B}_1$, $P_{-1} = P_0 = 0_{n\times s}$
For $i = 1,2,\dots$ until convergence, do:
    $\bar{W}_i = AV_i - U_iA_i^T$
    $U_{i+1}B_{i+1} = \bar{W}_i$, (QR decomposition of $\bar{W}_i$)
    $\bar{S}_i = A^TU_{i+1} - V_iB_{i+1}^T$
    $V_{i+1}A_{i+1} = \bar{S}_i$, (QR decomposition of $\bar{S}_i$)
    Compute a unitary matrix $Q(a_i, b_i, c_i, d_i)$ such that
        $Q(a_i, b_i, c_i, d_i)\begin{bmatrix} \hat{\hat{\rho}}_i \\ B_{i+1} \end{bmatrix} = \begin{bmatrix} \rho_i \\ 0 \end{bmatrix}$
        $\theta_{i+1} = b_iA_{i+1}^T$
        $\hat{\hat{\rho}}_{i+1} = d_iA_{i+1}^T$
        $\tilde{\bar{\rho}}_i = \bar{d}_{i-1}\rho_i^T$
    Compute a unitary matrix $Q(\bar{a}_i, \bar{b}_i, \bar{c}_i, \bar{d}_i)$ such that
        $Q(\bar{a}_i, \bar{b}_i, \bar{c}_i, \bar{d}_i)\begin{bmatrix} \tilde{\bar{\rho}}_i \\ \theta_{i+1}^T \end{bmatrix} = \begin{bmatrix} \bar{\rho}_i \\ 0 \end{bmatrix}$
        $\bar{\theta}_{i+1} = \bar{b}_i\rho_{i+1}^T$
        $\zeta_i = \bar{a}_i\bar{\zeta}_i$
        $\bar{\zeta}_{i+1} = \bar{c}_i\bar{\zeta}_i$
    $P_i = \left(V_i - P_{i-2}\bar{\theta}_{i-1}\theta_i - P_{i-1}\left(\bar{\rho}_{i-1}\theta_i + \bar{\theta}_i\rho_i\right)\right)\rho_i^{-1}\bar{\rho}_i^{-1}$
```

$$X_i = X_{i-1} + P_i\zeta_i$$

If max $\| \mathrm{col}_j(\overline{\zeta}_{i+1}) \|_2$ is small enough then stop
End Do.

The algorithm is a generalization of the LSMR algorithm. It reduces to the classical LSMR when $s = 1$. The Bl-LSMR algorithm will breakdown at step i, if $\overline{\rho}_i$ or ρ_i are singular. This happens when the matrix $\begin{bmatrix}\hat{\hat{\rho}}_i \\ B_{i+1}\end{bmatrix}$ or matrix $\begin{bmatrix}\tilde{\overline{\rho}}_i \\ \theta_{i+1}^T\end{bmatrix}$ is not full rank. So the Bl-LSMR algorithm will not breakdown at step i, if B_{i+1} and θ_{i+1} are nonsingular. We will not treat the problem of breakdown in this paper and we assume that the matrices B_i's and θ_{i+1}'s produced by the Bl-LSMR algorithm are nonsingular.

4. The convergence of the Bl-LSMR algorithm

In this section, the convergence of the Bl-LSMR algorithm is studied. In order to get this goal, we first give the following lemma.

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 (2). Then, we have

(a) $P_i^TA^TAA^TAP_j = \begin{cases} I_s, & i = j \\ 0_s, & i \neq j \end{cases}$

(b) $(A^TR_{k-1})^TA^TAP_k = \zeta_k^T$

Proof: (a) It is enough to show that

$$\overline{P}_k^TA^TAA^TA\overline{P}_k = I_{ks},$$

where I_{ks} is a $ks \times ks$ identity matrix and $\overline{P}_k = [P_1, P_2, \dots, P_k]$ is an $n \times ks$ matrix whose columns are defined by (19). By using $\overline{P}_k = \overline{V}_k\overline{R}_k^{-1}\overline{\overline{R}}_k^{-1}$ and equation (5), we have

$$\begin{aligned}
A^TA\overline{P}_k &= A^TA\overline{V}_k\overline{R}_k^{-1}\overline{\overline{R}}_k^{-1} \\
&= \overline{V}_{k+1}\begin{bmatrix}T_k^TT_k \\ A_{k+1}E_{k+1}^TT_k\end{bmatrix}\overline{R}_k^{-1}\overline{\overline{R}}_k^{-1} \\
&= \overline{V}_{k+1}\begin{bmatrix}\overline{R}_k^T\overline{R}_k \\ \overline{B}_{k+1}\overline{E}_k^T\end{bmatrix}\overline{R}_k^{-1}\overline{\overline{R}}_k^{-1} && (\text{from}(8)) \\
&= \overline{V}_{k+1}\begin{bmatrix}\overline{R}_k^T \\ A_{k+1}B_{k+1}\rho_k^{-1}\overline{E}_k^T\end{bmatrix}\overline{\overline{R}}_k^{-1} && (\text{from}(8)) \\
&= \overline{V}_{k+1}\begin{bmatrix}\overline{R}_k^T \\ \theta_{k+1}^T\overline{E}_k^T\end{bmatrix}\overline{\overline{R}}_k^{-1} && (\text{from}(12)) \\
&= \overline{V}_{k+1}\overline{Q}_{k+1}^T\begin{bmatrix}\overline{\overline{R}}_k \\ 0\end{bmatrix}\overline{\overline{R}}_k^{-1} && (\text{from}(14)) \\
&= \overline{V}_{k+1}\overline{Q}_{k+1}^T\begin{bmatrix}I_{ks} \\ 0\end{bmatrix}. && (25)
\end{aligned}$$

Since $\overline{V}_{k+1}$ is orthonormal and $\overline{Q}_{k+1}$ is a unitary matrix, we get
$\overline{P}_k^TA^TAA^TA\overline{P}_k = I_{ks}$.
(b) From (18), (25), and (15), we have

$$(A^TR_{k-1})^TA^TAP_k = (\overline{V}_k\overline{Q}_k^T\begin{bmatrix}0 \\ \overline{\zeta}_k\end{bmatrix})^T\overline{V}_{k+1}\overline{Q}_{k+1}^T\begin{bmatrix}I_{ks} \\ 0\end{bmatrix}\overline{E}_k$$

$$= \begin{bmatrix}0 & \overline{\zeta}_k^T\end{bmatrix}\overline{Q}_k\overline{V}_k^T\begin{bmatrix}\overline{V}_k & V_{k+1}\end{bmatrix}\begin{bmatrix}\overline{Q}_k^T & 0 \\ 0 & I_s\end{bmatrix}\begin{bmatrix}I_{(k-1)s} & 0 \\ 0 & Q^T(\overline{a}_k, \overline{b}_k, \overline{c}_k, \overline{d}_k)\end{bmatrix}\begin{bmatrix}0_{(k-1)s} \\ I_s \\ 0_s\end{bmatrix}$$

$$= \begin{bmatrix}0 & \overline{\zeta}_k^T\end{bmatrix}\begin{bmatrix}I_{ks} & 0_s\end{bmatrix}\begin{bmatrix}0_{(k-1)s} \\ \overline{a}_k^T \\ \overline{b}_k^T\end{bmatrix} = \overline{\zeta}_k^T\overline{a}_k^T = \zeta_k^T.$$

Theorem 3. Let X_k be the approximate solution of (2), obtained from the Bl-LSMR algorithm. Then
$\| A^TR_k \|_F \leq \| A^TR_{k-1} \|_F,$
where $R_k = B - AX_k$.

Proof: We have

$(A^TR_k)^T(A^TR_k) =$
$(A^TR_{k-1} - A^TAP_k\zeta_k)^T(A^TR_{k-1} - A^TAP_k\zeta_k)$
$= (A^TR_{k-1})^T(A^TR_{k-1}) - 2(A^TR_{k-1})^TA^TAP_k\zeta_k$
$+\zeta_k^TP_k^TA^TAA^TAP_k\zeta_k.$

By using Lemma 2, we get

$$(A^TR_k)^T(A^TR_k) = (A^TR_{k-1})^T(A^TR_{k-1}) - \zeta_k^T\zeta_k.$$

Thus
$\mathrm{trace}((A^TR_k)^T(A^TR_k))$
$= \mathrm{trace}((A^TR_{k-1})^T(A^TR_{k-1})) - \mathrm{trace}(\zeta_k^T\zeta_k),$
$\| A^TR_k \|_F^2 = \| A^TR_{k-1} \|_F^2 - \| \zeta_k \|_F^2,$
$\| A^TR_k \|_F \leq \| A^TR_{k-1} \|_F.$

5. Numerical examples

In this section, we give some experimental results.

Our examples have been coded in Matlab with double precision. For all the experiments, the initial guess was $X_0 = 0$ and $B = \mathrm{rand}(n,s)$, where function rand creates an $n \times s$ random matrix with coefficients uniformly distributed in $[0,1]$. No preconditioning has been used in any of the test problems. All the tests were stopped as soon as, $\| A^T R_k \|_F = \| \bar{\zeta}_{k+1} \|_F \leq 10^{-10}$.

We use some matrices from the University of Florida Sparse Matrix Collection (Davis [7]). These matrices with their properties are shown in Table 1. In Table 2, for $s = 10,20,30$, and 40, we give the ratio $t(s)/t(1)$, 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, in our experiments, $t(1)$ is the average of the times needed for the s right-hand sides using LSMR. We note that Bl-LSMR is effective if the indicator $t(s)/t(1)$ is less than s. Table 2 shows that the Bl-LSMR algorithm is effective and less expensive than the LSMR algorithm applied to each right-hand side. To show that the Frobenius norm of residual matrix decreases monotonically, we display the convergence history in Fig. 1 for the systems corresponding to the matrices of Table 2 and Bl-LSMR algorithm, respectively. In this figure, the vertical axis and horizontal axis are the logarithm 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.

Table 1. Test problems information

Matrix\ Property	Order	Sym	nnz	id
Rajat/ rajat03	7602	no	32653	1187
HB/ sherman4	1104	no	3786	245
FEMLAB/ ns3Da	20414	no	1659799	925
Simon/ appu	14000	no	1853104	811
Bai/ pde2961	2961	no	14585	324

Table 2. Effectiveness of BL-LSMR algorithm measured by $t(s)/t(1)$

Matrix\S	10	20	30	40
Rajat03	1.98	6.31	3.54	18.98
Sherman4	0.15	0.16	0.18	0.45
Ns3Da	2.15	4.50	7.00	10.25
appu	0.93	5.96	0.98	15.10
Pde2961	0.70	1.04	0.81	0.87

6. Conclusion

We have proposed in this paper a new block LSMR algorithm for solving nonsymmetric linear systems with multiple right-hand sides. To define this new algorithm, we used the block Bidiag 1 procedure (Karimi and Toutounian, 2006) and derived a simple recurrence formulas for generating the sequences of approximations $\{X_k\}$ such that $\left\|\mathrm{col}_j(A^T R_k)\right\|_2$ decreases monotonically. In practice we also observed that the Frobenius norm of residual matrix decreased monotonically. Experimental results showed that the proposed methods are effective and less expensive than the LSMR algorithm applied to each right-hand side.

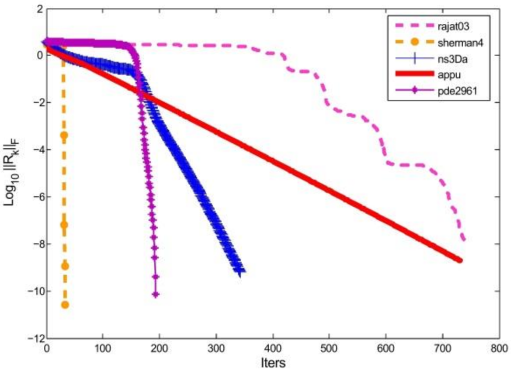

Fig. 1. Convergence history of the **Bl-LSMR** algorithm with s=10

Appendix A. Proof of Lemma 1.

Let $Q_k = \widehat{P}_k \widehat{P}_{k-1} \ldots \widehat{P}_1$, $\overline{Q}_k = \overline{P}_k \overline{P}_{k-1} \ldots \overline{P}_1$, and

$$\widetilde{R}_k = \begin{bmatrix} \tilde{\rho}_1 & & \tilde{\theta}_2 & \\ & \ddots & \ddots & \\ & & \tilde{\rho}_{k-1} & \tilde{\theta}_k \\ & & & \dot{\rho}_k \end{bmatrix} \qquad (26)$$

Then, with $\ddot{\beta}_1 = B_1$, $\tilde{\bar{\rho}}_= \rho_1^T$, $\dot{\rho}_1 = \bar{\rho}_1^T$, $\dot{\beta}_1 = \hat{\beta}_1$, $\hat{\hat{\rho}}_1 = A_1^T$, the effects of the rotations $\widehat{P}_k$, $\overline{P}_k$ and $\widetilde{P}_{k-1}$ can be summarized as

$$\begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \hat{\hat{\rho}}_k & 0 & \ddot{\beta}_k \\ B_{k+1} & A_{k+1}^T & 0 \end{bmatrix} = \begin{bmatrix} \rho_k & \theta_{k+1} & \hat{\beta}_k \\ 0 & \hat{\hat{\rho}}_{k+1} & \ddot{\beta}_{k+1} \end{bmatrix}, \qquad (27)$$

$$\begin{bmatrix} \bar{a}_k & \bar{b}_k \\ \bar{c}_k & \bar{d}_k \end{bmatrix} \begin{bmatrix} \bar{\bar{\rho}}_k & 0 \\ \theta_{k+1}^T & \rho_{k+1}^T \end{bmatrix} = \begin{bmatrix} \bar{\rho}_k & \bar{\theta}_{k+1} \\ 0 & \bar{\bar{\rho}}_{k+1} \end{bmatrix}, \qquad (28)$$

$$\begin{bmatrix} \tilde{a}_{k-1} & \tilde{b}_{k-1} \\ \tilde{c}_{k-1} & \tilde{d}_{k-1} \end{bmatrix} \begin{bmatrix} \dot{\rho}_{k-1} & 0 & \dot{\beta}_{k-1} \\ \bar{\theta}_k^T & \bar{\rho}_k^T & \hat{\beta}_k \end{bmatrix} = \begin{bmatrix} \tilde{\rho}_{k-1} & \tilde{\theta}_k & \tilde{\beta}_{k-1} \\ 0 & \dot{\rho}_k & \dot{\beta}_k \end{bmatrix}, \qquad (29)$$

where $a_k, b_k, c_k, d_k, \bar{a}_k, \bar{b}_k, \bar{c}_k$ and $\bar{d}_k$ are defined in section 3. From (27), (28), and (29), with $d_0 = I$ we have

$$
\begin{aligned}
&\text{(i) } a_k = \rho_k^{-T}\hat{\hat{\rho}}_k^T = \rho_k^{-T}A_k d_{k-1}^T,\\
&\text{(ii) } c_k = -\hat{\hat{\rho}}_{k+1}^{-T}\theta_{k+1}^T a_k\\
&\qquad = -d_k^{-T}A_{k+1}^{-1}\theta_{k+1}^T\rho_k^{-T}A_k d_{k-1}^T,\\
&\text{(iii) } \bar{a}_k = \left(\bar{\rho}_k - \bar{b}_k\theta_{k+1}^T\right)\tilde{\bar{\rho}}_k^{-1}\\
&\qquad = (\bar{\rho}_k - \bar{\theta}_{k+1}\rho_{k+1}^{-T}\theta_{k+1}^T)\rho_k^{-T}\bar{d}_{k-1}^{-1},\\
&\text{(iv) } \bar{c}_k = -\bar{d}_k\theta_{k+1}^T\tilde{\bar{\rho}}_k^{-1} = -\bar{d}_k\theta_{k+1}^T\rho_k^{-T}\bar{d}_{k-1}^{-1},\\
&\text{(v) } \tilde{a}_k = (\tilde{\rho}_k)^{-T}\dot{\rho}_k^T = (\tilde{\rho}_k)^{-T}\bar{\rho}_k\tilde{d}_{k-1}^T,\\
&\text{(vi) } \tilde{b}_k = (\tilde{\rho}_k)^{-T}\bar{\theta}_{k+1},\\
&\text{(vii) } \tilde{\theta}_k = \tilde{b}_{k-1}\bar{\rho}_k^T\\
&\text{(viii) } \tilde{b}_k^T\tilde{a}_k = -\tilde{d}_k^T\tilde{c}_k\\
&\text{(ix) } \tilde{b}_k^T\tilde{b}_k = I - \tilde{d}_k^T\tilde{d}_k
\end{aligned}
$$

From the definitions of F_k, $\tilde{F}_k$, $\widetilde{R}_k$, and z_k, we have

$$\widetilde{R}_k^T\tilde{F}_k = \bar{\bar{R}}_k F_k = z_k = [I_{ks}0_{ks,s}]\bar{Q}_{k+1}E_1\bar{B}_1. \tag{30}$$

So, the matrix $\tilde{F}_k$ can be computed by the block forward substitution from $\widetilde{R}_k^T\tilde{F}_k = [I_{ks}0_{ks,s}]\bar{Q}_{k+1}E_1\bar{B}_1$. By defining $c^{(k)} = c_k c_{k-1}\dots c_2c_1$ and $\bar{c}^{(k)} = \bar{c}_k\bar{c}_{k-1}\dots\bar{c}_2\bar{c}_1$ and expanding (30) and (16), we have

$$\widetilde{R}_k^T\tilde{F}_k = \begin{bmatrix}\bar{a}_1\\ \bar{a}_2\bar{c}_1\\ \vdots\\ \bar{a}_k\bar{c}^{(k-1)}\end{bmatrix}\bar{B}_1,$$

$$\widetilde{B}_k = \begin{bmatrix}\widetilde{Q}_k & \\ & I_s\end{bmatrix}\begin{bmatrix}a_1\\ a_2c_1\\ \vdots\\ a_kc^{(k-1)}\\ c^{(k)}\end{bmatrix}B_1.$$

These imply that

$$\tilde{\tau}_1 = (\tilde{\rho}_1^T)^{-1}\bar{a}_1\bar{B}_1 \tag{A.1}$$
$$\tilde{\tau}_{k-1} = (\tilde{\rho}_{k-1}^T)^{-1}(\bar{a}_{k-1}\bar{c}^{(k-2)}\bar{B}_1 - \tilde{\theta}_{k-1}^T\tilde{\tau}_{k-2}) \tag{A.2}$$
$$\dot{\tau}_k = (\dot{\rho}_k^T)^{-1}(\bar{a}_k\bar{c}^{(k-1)}\bar{B}_1 - \tilde{\theta}_k^T\tilde{\tau}_{k-1}) \tag{A.3}$$
$$\dot{\beta}_1 = \hat{\beta}_1 = a_1B_1 \tag{A.4}$$
$$\dot{\beta}_k = \tilde{c}_{k-1}\dot{\beta}_{k-1} + \tilde{d}_{k-1}a_kc^{(k-1)}B_1 \tag{A.5}$$
$$\tilde{\beta}_k = \tilde{a}_k\dot{\beta}_k + \tilde{b}_ka_{k+1}c^{(k)}B_1 \tag{A.6}$$

Now, by induction, we show that $\tilde{\tau}_i = \tilde{\beta}_i$ for all i. When $i = 1$, (A.6) and (A.4) give

$$
\begin{aligned}
\tilde{\beta}_1 &= \left(\tilde{a}_1a_1 + \tilde{b}_1a_2c_1\right)B_1\\
&= (\tilde{\rho}_1^T)^{-1}(\bar{\rho}_1a_1 + \bar{\theta}_2a_2c_1)B_1 \quad (\text{from(v)and(vi)})\\
&= (\tilde{\rho}_1^T)^{-1}(\bar{\rho}_1\rho_1^{-T}A_1\\
&\quad -\bar{\theta}_2\rho_2^{-T}\theta_2^T\rho_1^{-T}A_1)B_1 \quad (\text{from(i)and(ii)})\\
&= (\tilde{\rho}_1^T)^{-1}(\bar{\rho}_1 - \bar{\theta}_2\rho_2^{-T}\theta_2^T)\rho_1^{-T}A_1\\
&= (\tilde{\rho}_1^T)^{-1}\bar{a}_1\bar{B}_1 \quad (\text{from(iii)})\\
&= \tilde{\tau}_1.
\end{aligned}
$$

From (i), (ii), and (iv) we have

$$
\begin{aligned}
&(\bar{\rho}_ka_k + \bar{\theta}_{k+1}a_{k+1}c_k)c^{(k-1)} = \big(\bar{\rho}_k\rho_k^{-T}A_kd_{k-1}^T -\\
&\bar{\theta}_{k+1}\rho_{k+1}^{-T}\theta_{k+1}^T\rho_k^{-T}A_kd_{k-1}^T\big)c^{(k-1)}\\
&= \left(\bar{\rho}_k - \bar{\theta}_{k+1}\rho_{k+1}^{-T}\theta_{k+1}^T\right)\rho_k^{-T}A_kd_{k-1}^Tc^{(k-1)}\\
&= \bar{a}_k\bar{d}_{k-1}A_kd_{k-1}^Tc^{(k-1)}\\
&= (-1)^{k-1}\bar{a}_k\bar{d}_{k-1}A_kd_{k-1}^T(d_{k-1}^{-T}A_k^{-1}\theta_k^T\rho_{k-1}^{-T}A_{k-1}d_{k-2}^T)\\
&\quad \dots(d_1^{-T}A_2^{-1}\theta_2^T\rho_1^{-T}A_1)\\
&= (-1)^{k-1}\bar{a}_k(\bar{d}_{k-1}\theta_k^T\rho_{k-1}^{-T}\bar{d}_{k-2}^{-1})(\bar{d}_{k-2}\theta_{k-1}^T\rho_{k-2}^{-T}\bar{d}_{k-3}^{-1})\\
&\quad \dots(\bar{d}_1\theta_2^T\rho_1^{-T})A_1\\
&= \bar{a}_k\bar{c}^{(k-1)}A_1.
\end{aligned} \tag{A.7}
$$

Suppose $\tilde{\tau}_{k-1} = \tilde{\beta}_{k-1}$. Applying this induction hypothesis on $\tilde{\tau}_k = (\tilde{\rho}_k^T)^{-1}(\bar{a}_k\bar{c}^{(k-1)}\bar{B}_1 - \tilde{\theta}_k^T\tilde{\tau}_{k-1})$ gives

$$
\begin{aligned}
\tilde{\tau}_k &= (\tilde{\rho}_k^T)^{-1}(\bar{a}_k\bar{c}^{(k-1)}\bar{B}_1 - \tilde{\theta}_k^T\tilde{\beta}_{k-1})\\
&= (\tilde{\rho}_k^T)^{-1}(\bar{a}_k\bar{c}^{(k-1)}\bar{B}_1 - \tilde{\theta}_k^T(\tilde{a}_{k-1}\dot{\beta}_{k-1}\\
&\quad +\tilde{b}_{k-1}a_kc^{(k-1)}B_1)) \quad (\text{from(A.6)})\\
&= (\tilde{\rho}_k^T)^{-1}\bar{a}_k\bar{c}^{(k-1)}\bar{B}_1 - (\tilde{\rho}_k^T)^{-1}\bar{\rho}_k\tilde{b}_{k-1}^T(\tilde{a}_{k-1}\dot{\beta}_{k-1}\\
&\quad +\tilde{b}_{k-1}a_kc^{(k-1)}B_1) \quad (\text{from(vii)})\\
&= (\tilde{\rho}_k^T)^{-1}\bar{a}_k\bar{c}^{(k-1)}\bar{B}_1\\
&\quad +(\tilde{\rho}_k^T)^{-1}\bar{\rho}_k(\tilde{d}_{k-1}^T\tilde{c}_{k-1}\dot{\beta}_{k-1} + (\tilde{d}_{k-1}^T\tilde{d}_{k-1}\\
&\quad -I)a_kc^{(k-1)}B_1) \quad (\text{from(viii)and(ix)})\\
&= (\tilde{\rho}_k^T)^{-1}\bar{a}_k\bar{c}^{(k-1)}\bar{B}_1 + \tilde{a}_k\tilde{c}_{k-1}\dot{\beta}_{k-1}\\
&\quad +\tilde{a}_k\tilde{d}_{k-1}a_kc^{(k-1)}B_1\\
&\quad -(\tilde{\rho}_k^T)^{-1}\bar{\rho}_ka_kc^{(k-1)}B_1 \quad (\text{from(v)})\\
&= \tilde{a}_k\dot{\beta}_k + (\tilde{\rho}_k^T)^{-1}(\bar{a}_k\bar{c}^{(k-1)}\bar{B}_1\\
&\quad -\bar{\rho}_ka_kc^{(k-1)}B_1) \quad (\text{from(A.5)})\\
&= \tilde{a}_k\dot{\beta}_k\\
&\quad +(\tilde{\rho}_k^T)^{-1}\left(\bar{\theta}_{k+1}a_{k+1}c_kc^{(k-1)}\right) \quad (\text{from(A.7)})\\
&= \tilde{a}_k\dot{\beta}_k + \tilde{b}_ka_{k+1}c^{(k)}B_1 \quad \left(\text{from(vi)}\right)\\
&= \tilde{\beta}_k.
\end{aligned}
$$

Therefore by induction, we know that $\tilde{\tau}_i = \tilde{\beta}_i$ for $i = 1,2,\dots k-1$. From (24), we see that at iteration k, the first $k-1$ blocks of $\widetilde{B}_k$ and $\tilde{F}_k$ are equal.

References

Abdel-Rehim, A. M., Morgan, R. B., & Wilcox, W. (2008). Improved seed methods for symmetric positive definite linear equations with multiple right-hand sides. http://arxiv.org/abs/0810.0330 {math-ph}.

Bellalij, M., Jbilou, K., & Sadok, H. (2008). New convergence results on the global GMRES method for diagonalizable matrices. *Journal of Computational and Applied Mathematics*, 219, 350–358.

Chan, T. F., & Wang, W. (1997). Analysis of projection methods for solving linear systems with multiple right-hand sides. *SIAM Journal on Scientific Computing*, 18, 1698–1721.

Chin-Lung Fong, D., & Saunders, M. (2011). LSMR: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33, 2950–2971.

Dai, H. (2000). Two algorithms for symmetric linear systems with multiple right-hand sides. *Numerical Mathematics a Journal of Chinese Universities*, 9, 91–110.

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

Davis, T. A. (2011). University of Florida Sparse Matrix Collection, http://www.cise.ufl.edu/research/sparse/ matrices.

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

Golub, G. H., & Van Loan, C. F. (1983). *Matrix Computations,* Johns Hopkins University Press, Baltimore, MD.

Golub, G. H., & Kahan, W. (1965). Algorithm LSQR is based on the Lanczos process and bidiagonalization procedure. *SIAM Journal on Numerical Analysis,* 2, 205–224.

Gu, G., & Cao, Z. (2001). A block GMRES method augmented with eigenvectors. *Applied Mathematics and Computation,* 121, 271–289.

Gu, C., & Yang, Z. (2007). Global SCD algorithm for real positive definite linear systems with multiple right-hand sides. *Applied Mathematics and Computation,*189, 59–67.

Guennouni, A. El., Jbilou, K., & Sadok, H. (2003). A block version of BICGSTAB for linear systems with multiple right-hand sides. *Electronic Transactions on Numerical Analysis,* 16, 129–142.

Guennouni, A. El., Jbilou, K., & Sadok, H. (2004). The block Lanczos method for linear systems with multiple right-hand sides. *Applied Numerical Mathematics,* 51, 243–256.

Gutknecht, M. H. (2007). *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* (pp. 420–447). Anamaya Publishers, New Delhi, India.

Haase, G., & Reitzinger, S. (2005). Cache issues of algebraic multigrid methods for linear systems with multiple right-hand sides. *SIAM Journal on Scientific Computing,* 27, 1–18.

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

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

Jbilou, K., Messaoudi, A., & Sadok, H. (1999). Global FOM and GMRES algorithms for matrix equations. *Applied Numerical Mathematics,* 31, 49–63.

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

Jbilou, K., Sadok, H., & Tinzefte, A. (2005). Oblique projection methods for linear systems with multiple right-hand sides. *Electronic Transactions on Numerical Analysis,* 20, 119–138.

Joly, P. (1991). *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.

Karimi, S., & Toutounian, F. (2006). The block least squares method for solving nonsymmetric linear systems with multiple right-hand sides. *Applied Mathematics and Computation,* 177, 852–862.

Lin, Y. (2005). Implicitly restarted global FOM and GMRES for nonsymmetric matrix equations and Sylvester equations. *Applied Mathematics and Computations,* 167, 1004–1025.

Liu, H. -L., & Zhong, B. -J. (2008). Simpler block GMRES for nonsymmetric systems with multiple right-hand sides. *Electronic Transactions Numerical Analysis,* 30, 1–9.

Morgan, R. B. (2005). Restarted block- GMRES with deflation of eignvalues. *Applied Numerical Mathematics,* 54, 222–236.

Nikishin, A., & Yeremin, A. (1995). Variable block CG algorithms for solving large sparse symmetric positive definite linear systems on parallel computers I: general iterative scheme. *SIAM Journal of Matrix Analysis,* 16, 1135–1153.

Oleary, D. (1980). The block conjugate gradient algorithm and related methods. *Linear Algebra and its Applications*, 29, 293–322.

Paige, C. C., & Saunders, M. A. (1982). LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematics Software,* 8, 43–71.

Paige, C. C., & Saunders, M. A. (1975). Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis,*12, 617–629.

Robbe, M., & Sadkane, M. (2006). Exact and inexact breakdowns in the block GMRES method. *Linear Algebra and its Applications*, 419, 265–285.

Saad, Y. (1987). On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Mathematics and Computation,* 48, 651–662.

Salkuyeh, D. K. (2006). CG-type algorithms to solve symmetric matrix equations. *Applied Mathematics and Computations*,172, 985–999.

Simoncini, V. (1997). A stabilized QMR version of block BICG. *SIAM Journal on Matrix Analysis and Applications,* 18, 419–434.

Simoncini, V., & Gallopoulos, E. (1995). An iterative method for nonsymmetric systems with multiple right-hand sides. *SIAM Journal on Scientific Computing,*16, 917–933.

Simoncini, V., & Gallopoulos, E. (1996). Convergence properties of block GMRES and matrix polynomials. *Linear Algebra and its Applications,* 247, 97–119.

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

Toutounian, F., & Karimi, S. (2006). Global least squares method (Gl-LSQR) for solving general linear systems with several right-hand sides. *Applied Mathematics and Computations,* 178, 452–460.

Van Der Vorst, H. (1987). An iterative solution method for solving f(A) = b, using Krylov subspace information obtained for the symmetric positive definite matrix A. *Journal of Computational and Applied Mathematics,* 18, 249–263.

Vital, B. (1990). *Etude de quelques methodes de resolution de problemes linears de grande taille sur* *multiprocesseur*. Ph.D. Thesis, University de Rennes.

Zhang, J., & Dai, H. (2008). Global CGS algorithm for linear systems with multiple right-hand sides. *Numerical Mathematics a Journal of Chinese Universities*, 30, 390–399.

Zhang, J., Dai, H., & Zhao, J. (2010). Generalized global conjugate gradient squared algorithm. *Applied Mathematics and Computation,* 216, 3694–3706.

Zhang, J., Dai, H., & Zhao, J. (2011). A new family of global methods for linear systems with multiple right-hand sides. *Jornal of Computational Applied Mathematics,* 236, 1562–1575.