

A New Iterative Solution Method for Solving Multiple Linear Systems

Saeed Karimi

Department of Mathematics, Persian Gulf University, Bushehr, Iran
Email: karimi@pgu.ac.ir

Received July 25, 2012; revised August 25, 2012; accepted September 8, 2012

ABSTRACT

In this paper, a new iterative solution method is proposed for solving multiple linear systems $A^{(i)}x^{(i)} = b^{(i)}$, for $1 \leq i \leq s$, where the coefficient matrices $A^{(i)}$ and the right-hand sides $b^{(i)}$ are arbitrary in general. The proposed method is based on the global least squares (GL-LSQR) method. A linear operator $\mathcal{L} : \mathbb{R}^{n \times s} \rightarrow \mathbb{R}^{n \times s}$ is defined to connect all the linear systems together. To approximate all numerical solutions of the multiple linear systems simultaneously, the GL-LSQR method is applied for the operator \mathcal{L} and the approximate solutions are obtained recursively. The presented method is compared with the well-known LSQR method. Finally, numerical experiments on test matrices are presented to show the efficiency of the new method.

Keywords: Iterative Method; Multiple Linear Systems; LSQR Method; GL-LSQR Method; Projection Method

1. Introduction

We want to solve, using global least squares (GL-LSQR) method, the following linear systems:

$$A^{(i)}x^{(i)} = b^{(i)}, \quad 1 \leq i \leq s \quad (1)$$

where $A^{(i)}$ are arbitrary matrices of order n , and in general $A^{(i)} \neq A^{(j)}$ and $b^{(i)} \neq b^{(j)}$ for $i \neq j$. In spite of that, in many practical application the coefficient matrices and the right-hand sides are not arbitrary, and often there is information that can be shared among the coefficient matrices and right-hand sides. Multiple linear systems (1) arise in many problems in scientific computing and engineering application, including recursive least squares computations [1,2], wave scattering problems [3,4], numerical methods for integral equations [4,5], and image restorations [6].

Many authors, see [7,8], have researched to approximate the solutions of the multiple linear systems (1) with the same coefficient matrix but different right-hand sides, *i.e.*,

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

Recently, S. Karimi and F. Toutounian proposed an iterative method for solving the linear systems (2) with the some advantages over the existing methods, see [9-11], for more details. In [12], Tony F. chan and Michael K. Ng presented the Galerkin projection method for solving linear systems (1). They focused on the seed

projection method which generates a Krylov subspace from a set of direction vectors obtained by solving one of the systems, called the seed system, by the CG method and then projects the residuals of other systems onto the generated Krylov subspace to get the approximate solutions. Note that the coefficient matrices in this method are real symmetric positive definite.

In this paper, we propose a new method to solve the linear systems (1) simultaneously, where the coefficient matrices and right-hand sides are arbitrary. We define a linear operator $\mathcal{L} : \mathbb{R}^{n \times s} \rightarrow \mathbb{R}^{n \times s}$ to connect all the linear systems (1) together. Then we apply the GL-LSQR method [13] for the linear operator \mathcal{L} and obtain recursively the approximate solutions simultaneously. In the new method, the linear operator \mathcal{L} will be reduced to a lower global bidiagonal, namely \mathcal{L} -Bidiag, matrix form. We obtain a recurrence formula for generating the sequence of approximate solutions. Our new method has certain advantages over the existent methods. In the new method, the coefficient matrices $A^{(i)}$ and right-hand sides $b^{(i)}$ are arbitrary. Also we do not need to store the basis vectors, we do not need to predetermine a subspace dimension and the approximate solutions and residuals are cheaply computed at every stage of the algorithm because they are updated with short-term recurrence.

The remainder of the paper is organized as follows. Section 2 is devoted to a short review of the global least squares (GL-LSQR) method. In section 3, we present a new method, namely \mathcal{L} -GL-LSQR, to solve the multi-

ple linear systems (1). we also show how to reduce the low-rank approximate solutions to linear operator \mathcal{L} . The section 4 is devoted to some numerical experiments and comparing the new method with the well known LSQR method, when it is applied to s linear systems independently. Finally, we make some concluding remarks in section 5.

We use the following notations. For X and Y two matrices in $\mathbb{R}^{n \times s}$, we consider the following inner product $\langle X, Y \rangle_F = \text{tr}(X^T Y)$, where $\text{tr}(\cdot)$ denotes the trace of a matrix. The associated norm is the Frobenious norm denoted by $\|\cdot\|_F$. The notation $X \perp_F Y$ means that $\langle X, Y \rangle_F = 0$ and $X(:, i)$ means that the i th column of X . Finally, we use the notation $*$ for the following product:

$$\mathcal{V}^* y = \sum_{j=1}^m y_j V_j, \quad (3)$$

where $\mathcal{V} = [V_1, V_2, \dots, V_m]$, $V_j \in \mathbb{R}^{n \times s}$ for $1 \leq j \leq m$, and $y \in \mathbb{R}^m$.

By the same way, we define

$$\mathcal{V}^* T = [\mathcal{V}^* T(:, 1), \mathcal{V}^* T(:, 2), \dots, \mathcal{V}^* T(:, m)], \quad (4)$$

where T is the $m \times m$ matrix. It is easy to show that the following relations are satisfied:

$$\mathcal{V}^*(y+z) = \mathcal{V}^* y + \mathcal{V}^* z, \quad (\mathcal{V}^* T)^* y = \mathcal{V}^*(Ty), \quad (5)$$

where y and z are two vectors of \mathbb{R}^m .

2. The Global Least Squares (GL-LSQR) Method

In this section, we recall some fundamental properties of GL-LSQR algorithm [13], which is an iterative method for solving the multiple linear systems (2). When all the $b^{(i)}$'s are available simultaneously, the multiple linear systems (2) can be written as

$$AX = B, \quad (6)$$

where A is an $n \times n$ nonsingular arbitrary matrix, B and X are an $n \times s$ rectangular matrices whose columns are $b^{(1)}, b^{(2)}, \dots, b^{(s)}$ and $x^{(1)}, x^{(2)}, \dots, x^{(s)}$, respectively.

For solving the matrix Equation (6), the GL-LSQR method uses a procedure, namely Global-Bidiag procedure, to reduce A to the global lower bidiagonal form. The Global-Bidiag procedure can be described as follows.

Global-Bidiag (starting matrix B ; reduction to global lower bidiagonal form):

$$\left. \begin{aligned} \beta_1 U_1 &= B, \alpha_1 V_1 = A^T U_1 \\ \beta_{i+1} U_{i+1} &= AV_i - \alpha_i U_i \\ \alpha_{i+1} V_{i+1} &= A^T U_{i+1} - \beta_{i+1} V_i \end{aligned} \right\}, \quad i = 1, 2, \dots, \quad (7)$$

where $U_i, V_i \in \mathbb{R}^{n \times s}$. The scalars $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen so that $\|U_i\|_F = \|V_i\|_F = 1$.

With the definitions

$$\begin{aligned} \mathcal{U}_k &\equiv [U_1, U_2, \dots, U_k], \\ \mathcal{V}_k &\equiv [V_1, V_2, \dots, V_k], \\ T_k &\equiv \begin{bmatrix} \alpha_1 & & & & \\ & \beta_2 & \alpha_2 & & \\ & & \ddots & \ddots & \\ & & & \beta_k & \alpha_k \\ & & & & \beta_{k+1} \end{bmatrix}, \end{aligned}$$

the recurrence relations (7) may be rewritten as:

$$\mathcal{U}_{k+1} * (\beta_1 e_1) = B, \quad (8)$$

$$A \mathcal{V}_k = \mathcal{U}_{k+1} * T_k, \quad (9)$$

$$A^T \mathcal{U}_{k+1} = \mathcal{V}_k * T_k^T + \alpha_{k+1} V_{k+1} * e_{k+1}^T. \quad (10)$$

For the Global-Bidiag procedure, we have the following propositions.

Proposition 1 [13]. Suppose that k step of the Global-Bidiag procedure have been taken, then the $n \times s$ block vectors V_1, V_2, \dots, V_k and $U_1, U_2, \dots, U_k, U_{k+1}$ are F-orthonormal basis of the Krylov subspaces $\mathcal{K}_k(A^T A, V_1)$ and $\mathcal{K}_{k+1}(AA^T, U_1)$, respectively.

Proposition 2 [13]. The Global-Bidiag procedure will be stopped at step m if and only if $m = \min\{\mu, \lambda\}$, where μ and λ are the grades of V_1 and U_1 with respect to $A^T A$ and AA^T , respectively.

Proposition 3 [13]. Let \mathcal{U}_k be the matrix defined by $\mathcal{U}_k \equiv [U_1, U_2, \dots, U_k]$, where the $n \times s$ matrices U_i , $i = 1, \dots, k$, are generated by the Global-Bidiag procedure. Then

$$\|\mathcal{U}_k * \eta\|_F = \|\eta\|_2.$$

By using the Global-Bidiag procedure, the GL-LSQR algorithm constructs an approximate solution of the form $X_k = V_k * y_k$, where $y_k = [y_k^{(1)}, \dots, y_k^{(k)}]^T \in \mathbb{R}^k$, which solves the least-squares problem,

$$\min_X \|B - AX\|_F.$$

The main steps of the GL-LSQR algorithm can be summarized as follows.

Algorithm 1: GI-LSQR algorithm

- 1) Set $X_0 = 0$
- 2) $\beta_1 = \|B\|_F$, $U_1 = B/\beta_1$, $\alpha_1 = \|A^T U_1\|_F$,
 $V_1 = A^T U_1/\alpha_1$.
- 3) Set $W_1 = V_1$, $\bar{\phi}_1 = \beta_1$, $\bar{\rho}_1 = \alpha_1$
- 4) For $i = 1, 2, \dots$ until convergence, Do:
- 5) $\bar{W}_i = AV_i - \alpha_i U_i$
- 6) $\beta_{i+1} = \|\bar{W}_i\|_F$

- 7) $U_{i+1} = \bar{W}_i / \beta_{i+1}$
- 8) $\bar{S}_i = A^T U_{i+1} - \beta_{i+1} V_i$
- 9) $\alpha_{i+1} = \|\bar{S}_i\|_F$
- 10) $V_{i+1} = \bar{S}_i / \alpha_{i+1}$
- 11) $\rho_i = (\bar{\rho}_i^2 + \beta_{i+1}^2)^{1/2}$
- 12) $c_i = \bar{\rho}_i / \rho$
- 13) $s_i = \beta_{i+1} / \rho_i$
- 14) $\theta_{i+1} = s_i \alpha_{i+1}$
- 15) $\bar{\rho}_{i+1} = c_i \alpha_{i+1}$
- 16) $\phi_i = a_i \bar{\phi}_i$
- 17) $\phi_{i+1} = c_i \bar{\phi}_i$
- 18) $\phi_i = c_i \bar{\phi}_i$
- 19) $\phi_{i+1} = -s_i \phi_i$
- 20) $X_i = X_{i-1} + (\phi_i / \rho_i) W_i$
- 21) $W_{i+1} = V_{i+1} - (\theta_{i+1} / \rho_i) W_i$
- 22) If $|\bar{\phi}_{i+1}|$ is small enough then stop
- 23) EndDo.

More details about the GL-LSQR algorithm can be found [13].

3. The GL-LSQR-Like Operator Method

In this section, we propose a new method for solving the linear systems (1). For this mean, we define the following linear operator

$$\mathcal{L} : \mathbb{R}^{n \times s} \rightarrow \mathbb{R}^{n \times s} \quad (11)$$

$$\mathcal{L}(X) = [A^{(1)} X(:,1), \dots, A^{(s)} X(:,s)], \quad (12)$$

where $A^{(j)}$, $j=1, \dots, s$ are the coefficient matrices of the multiple linear systems (1). Therefore, the linear systems (1) is written as:

$$\mathcal{L}(X) = B, \quad (13)$$

where B is an $n \times s$ rectangular matrix whose columns are $b^{(1)}, b^{(2)}, \dots, b^{(s)}$ the right hand sides of the linear systems (1).

Definition 1: Let \mathcal{L} be linear operator (11). Then

$$\mathcal{L}^T(X) = [A^{(1)T} X(:,1), \dots, A^{(s)T} X(:,s)].$$

Consider the block operator (13), similar to the well-known block Krylov subspace

$$\mathcal{K}_m(A; R) = \text{span}\{R, AR, A^2 R, \dots, A^{m-1} R\},$$

where R is the residual of the operator equation (13), we define the following block Krylov-like subspace.

Definition 2: Let \mathcal{L} be linear operator (11). Then

$$\mathcal{K}_m(\mathcal{L}; R) = \text{span}\{R, \mathcal{L}(R), \dots, \mathcal{L}^{m-1}(R)\},$$

where $\mathcal{L}^i = \overbrace{\mathcal{L} \circ \mathcal{L} \circ \dots \circ \mathcal{L}}^{i \text{ time}}$ and \circ is the combination of two operators.

Definition 3: Let \mathcal{L} be linear operator (11) and $\mathcal{V}_m = [V_1, V_2, \dots, V_m] \in \mathbb{R}^{n \times ms}$. Then

$$\mathcal{L}(\mathcal{V}_m) = [\mathcal{L}(V_1), \mathcal{L}(V_2), \dots, \mathcal{L}(V_m)] \in \mathbb{R}^{n \times ms}.$$

where $V_j \in \mathbb{R}^{n \times s}$, $j=1, 2, \dots, s$.

To approximate the solution of the block operator Equation (13), we present a new algorithm, will be referred to \mathcal{L} -GL-LSQR algorithm, which is based on the Global-Bidiag-like procedure, will be referred to \mathcal{L} -Bidiag. The \mathcal{L} -Bidiag procedure reduces the linear operator \mathcal{L} to the lower bidiagonal matrix form. This procedure can be described as follows.

\mathcal{L} -Bidiag (starting matrix B ; reduction to lower bidiagonal matrix form):

$$\beta_1 U_1 = B, \alpha_1 V_1 = \mathcal{L}^T(U_1)$$

$$\left. \begin{aligned} \beta_{i+1} U_{i+1} &= \mathcal{L}(V_i) - \alpha_i U_i \\ \alpha_{i+1} V_{i+1} &= \mathcal{L}^T(U_{i+1}) - \beta_{i+1} V_i \end{aligned} \right\}, \quad i=1, 2, \dots, \quad (14)$$

where $U_i, V_i \in \mathbb{R}^{n \times s}$. The scalars $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen so that $\|U_i\|_F = \|V_i\|_F = 1$.

Similar to Global-Bidiag procedure, we define

$$\mathcal{U}_m \equiv [U_1, U_2, \dots, U_m],$$

$$\mathcal{V}_m \equiv [V_1, V_2, \dots, V_m],$$

$$T_m \equiv \begin{bmatrix} \alpha_1 & & & & & \\ & \beta_2 & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \beta_m & \alpha_m \\ & & & & & \beta_{m+1} \end{bmatrix}.$$

According to notation $*$ and by using the definition 3, the recurrence relations (14) can be rewritten as:

$$\mathcal{U}_{m+1} * (\beta_1 e_1) = B, \quad (15)$$

$$\mathcal{L}(\mathcal{V}_m) = \mathcal{U}_{m+1} * T_m, \quad (16)$$

$$\mathcal{L}^T(\mathcal{U}_{m+1}) = \mathcal{V}_m * T_k^T + \alpha_{m+1} V_{m+1} * e_{m+1}^T. \quad (17)$$

Proposition 4. Suppose that k step of the \mathcal{L} -Bidiag procedure have been taken, then the block vectors V_1, \dots, V_m and U_1, \dots, U_{m+1} are F-orthonormal basis of the Krylov-like subspace $\mathcal{K}_m(\mathcal{L}^T \mathcal{L}; V_1)$ and $\mathcal{K}_{m+1}(\mathcal{L} \mathcal{L}^T; U_1)$, respectively.

The proof of this proposition is similar to that given in [13].

The quantities generated from the linear operator \mathcal{L} and B by the \mathcal{L} -Bidiag process will now be used to solve the block least squares problem,

$$\min_X \|B - \mathcal{L}(X)\|_F.$$

Let the quantities

$$X_m = \mathcal{V}_m * y_m, \quad (18)$$

$$R_m = B - \mathcal{L}(X_m), \quad (19)$$

be defined, where $y_m \in \mathbb{R}^m$. According to linearity of the operator, it is easy to show that

$$\mathcal{L}(X_m) = \mathcal{L}(\mathcal{V}_m) * y_m.$$

Also it readily follows from (15), (16) and properties of product $*$ the equation

$$\begin{aligned} R_m &= B - \mathcal{L}(\mathcal{V}_m) * y_m \\ &= \mathcal{U}_{m+1} * (\beta_1 e_1) - (\mathcal{U}_{m+1} * T_m) * y_m \\ &= \mathcal{U}_{m+1} * (\beta_1 e_1 - T_m y_m), \end{aligned}$$

holds to working accuracy.

To minimize the m th residual $\|R_m\|_F$, since \mathcal{U}_{m+1} is F-orthonormal and by using the proposition 3, we choose y_m so that

$$\|R_m\|_F = \|\beta_1 e_1 - T_m y_m\|_2, \quad (20)$$

is minimum. This minimization problem is carried out by applying the QR decomposition [13], where a unitary matrix Q_m is determined so that

$$\begin{aligned} Q_m [T_m \quad \beta_1 e_1] &= \begin{bmatrix} \mathcal{R}_m & f_m \\ 0 & \bar{\phi}_{m+1} \end{bmatrix} \\ &= \begin{bmatrix} \rho_1 & \theta_1 & & & \phi_1 \\ & \rho_2 & \theta_3 & & \phi_2 \\ & & \ddots & \ddots & \vdots \\ & & & \rho_{m-1} & \theta_m & \phi_{m-1} \\ & & & & \rho_m & \phi_m \\ & & & & 0 & \phi_{m+1} \end{bmatrix}, \end{aligned}$$

where ρ_i , θ_i and ϕ_i are scalars. The above QR factorization is determined by constructing the m th plane rotation $Q_{m,m+1}$ to operate on rows m and $m+1$ of the transformed $[T_m \quad \beta_1 e_1]$ to annihilate β_{m+1} . This gives the following simple recurrence relation:

$$\begin{bmatrix} c_m & s_m \\ -s_m & c_m \end{bmatrix} \begin{bmatrix} \bar{\rho}_m & 0 & \bar{\phi}_m \\ \beta_{m+1} & \alpha_{m+1} & 0 \end{bmatrix} = \begin{bmatrix} \rho_m & \theta_{m+1} & \phi_m \\ 0 & \bar{\rho}_{m+1} & \bar{\phi}_{m+1} \end{bmatrix},$$

where $\bar{\rho}_1 \equiv \alpha_1$, $\bar{\phi}_1 \equiv \beta_1$ and the scalars c_m and s_m are the nontrivial elements of $Q_{m,m+1}$. The quantity $\bar{\rho}_m$ and $\bar{\phi}_m$ are intermediate scalars that are subsequently replaced by ρ_m and ϕ_m .

With setting

$$y_m = \mathcal{R}_m^{-1} * f_m,$$

the approximate solution is given by

$$X_m = \mathcal{V}_m * (\mathcal{R}_m^{-1} * y_m), \quad (21)$$

$$= (\mathcal{V}_m * \mathcal{R}_m^{-1}) * f_m. \quad (22)$$

Letting

$$\mathcal{P}_m \equiv \mathcal{V}_m * \mathcal{R}_m^{-1} \equiv [P_1 P_2 \cdots P_m],$$

then

$$X_m = \mathcal{P}_m * f_m.$$

The $n \times s$ matrix P_m , the last block column of \mathcal{P}_m , can be computed from the previous P_{m-1} and V_m , by the simple update

$$P_m = (V_m - \theta_m P_{m-1}) / \rho_m, \quad (23)$$

also note that,

$$f_m = \begin{bmatrix} f_{m-1} \\ \phi_m \end{bmatrix},$$

in which

$$\phi_m = c_m \bar{\phi}_m.$$

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

$$X_m = X_{m-1} + \phi_m P_m.$$

The residual norm $\|R_m\|_F$ is computed directly from the quantity $\bar{\phi}_{m+1}$ as

$$\|R_m\|_F = |\bar{\phi}_{m+1}|.$$

Some of the work in (23) can be eliminated by using matrices $W_m = \rho_m P_m$ in place of P_m . The main steps of the \mathcal{L} -GL-LSQR algorithm can be summarized as follows.

Algorithm 2: \mathcal{L} -GL-LSQR algorithm

- 1) Set $X_0 = 0$
- 2) $\beta_1 = \|B\|_F$, $U_1 = B / \beta_1$, $\alpha_1 = \|\mathcal{L}^T(U_1)\|_F$,
 $V_1 = \mathcal{L}^T(U_1) / \alpha_1$.
- 3) Set $W_1 = V_1$, $\bar{\phi}_1 = \beta_1$, $\bar{\rho}_1 = \alpha_1$
- 4) For $i = 1, 2, \dots$ until convergence, Do:
 - 5) $\hat{W}_i = \mathcal{L}(V_i) - \alpha_i U_i$
 - 6) $\beta_{i+1} = \|\hat{W}_i\|_F$
 - 7) $U_{i+1} = \hat{W}_i / \beta_{i+1}$
 - 8) $\hat{S}_i = \mathcal{L}^T(U_{i+1}) - \beta_{i+1} V_i$
 - 9) $\alpha_{i+1} = \|\hat{S}_i\|_F$
 - 10) $V_{i+1} = \hat{S}_i / \alpha_{i+1}$
 - 11) $\rho_i = (\bar{\rho}_i^2 + \beta_{i+1}^2)^{1/2}$
 - 12) $c_i = \bar{\rho}_i / \rho_i$

- 13) $s_i = \beta_{i+1}/\rho_i$
- 14) $\theta_{i+1} = s_i \alpha_{i+1}$
- 15) $\bar{\rho}_{i+1} = c_i \alpha_{i+1}$
- 16) $\phi_i = a_i \bar{\phi}_i$
- 17) $\bar{\phi}_{i+1} = c_i \bar{\phi}_i$
- 18) $\phi_i = c_i \bar{\phi}_i$
- 19) $\bar{\phi}_{i+1} = -s_i \phi_i$
- 20) $X_i = X_{i-1} + (\phi_i/\rho_i)W_i$
- 21) $W_{i+1} = V_{i+1} - (\theta_{i+1}/\rho_i)W_i$
- 22) If $|\bar{\phi}_{i+1}|$ is small enough then stop
- 23) EndDo.

As we observe, the \mathcal{L} -GL-LSQR algorithm has certain advantages, we obtain simultaneously the approximate solution of the multiple linear systems (1). Also the residual norm is cheaply computed at every stage of the algorithm.

As a application of the new method applying it to solve the Sylvester equation in a special case. Consider the Sylvester equation

$$AX - XB = C,$$

where $A \in \mathbb{R}^{n \times n}$, $C \in \mathbb{R}^{n \times s}$ and $B \in \mathbb{R}^{s \times s}$ are known and $X \in \mathbb{R}^{n \times s}$ is unknown. By using the Schur decomposition for the symmetric matrix B , the above Sylvester equation is returned to the s linear systems as follows. There is a unitary matrix Q such that $B = Q\Lambda Q^T$, where Λ is a diagonal matrix which diagonal elements are the eigenvalues of B . So we have

$$AXQ - XQ\Lambda = CQ = \hat{C},$$

by taking $\hat{X} = XQ$, the Sylvester equation is converted to the following s linearsystems

$$\hat{A}^{(j)} \hat{x}^{(j)} = \hat{c}^{(j)}, \quad j = i, \dots, s,$$

where $\hat{A}^{(j)} = A - \lambda_j I$, $\hat{x}^{(j)}$ and $\hat{c}^{(j)}$ are the j -th column of \hat{X} and \hat{C} , respectively.

4. Numerical Experiments

In this section, all the numerical experiments were computed in double precision with some MATLAB codes. For all the examples the initial guess X_0 was taken to be zero matrix. We consider two sets of numerical experiments. The first set of numerical experiments contains matrices of the form

$$A^{(i)} = A - \lambda_i I, \quad i = 1, 2, \dots, s,$$

obtained from the Sylvester equation, explained in previous section, where

$$A = \text{tridiag}\left(-1 + \frac{10}{n+1}, 4, -1 - \frac{10}{n+1}\right) \text{ and } \lambda_i \text{ is the ei-}$$

genvalue of the matrix

$$B = \text{tridiag}\left(-1 + \frac{1}{s+1}, 2, -1 + \frac{1}{s+1}\right).$$

The right-hand side matrix C is taken $\text{rand}(n, s)$, where the function rand creates an $n \times s$ random matrix. The matrices of the second set of experiments arise from the three-point centered discretization of the operator $-\frac{d}{dx}\left(a(x)\frac{du}{dx}\right)$

in $[0,1]$ where the function $a(x)$ is given by $a(x) = c + dx$, where c and d are two parameters. The discretization is performed using a grid size of $h = \frac{1}{65}$, yielding matrices

of size 64 with the values of $c_k = 0.1551 \times (0.9524)^k$ and $d_k = 7.7566 \times (0.9524)^k, k = 1, 2, \dots, s$. The right-hand sides of these systems are generated randomly with their 2-norms being 1. All the tests were stopped as soon as, $|\bar{\phi}| \leq 10^{-7}$.

We display the convergence history in **Figure 1** and **Figure 2** for the systems corresponding to the matrices of the first set of matrices and second set of matrices, respectively. **Figure 1** shows that the \mathcal{L} -GL-LSQR algorithm converges, however, **Figure 2** shows slowly convergence which can be remedied by using a reliable preconditioner. But we did not deal with the preconditioner techniques in this paper.

5. Conclusion

We proposed a new method for solving multiple linear systems $A^{(i)}x^{(i)} = b^{(i)}$, for $1 \leq i \leq s$, where the coefficient matrices $A^{(i)}$ and the right-hand sides $b^{(i)}$ are arbitrary in general. This method has certain advantages which is applied to the arbitrary coefficient matrices $A^{(i)}$ and right-hand sides $b^{(i)}$. Also It is not needed to

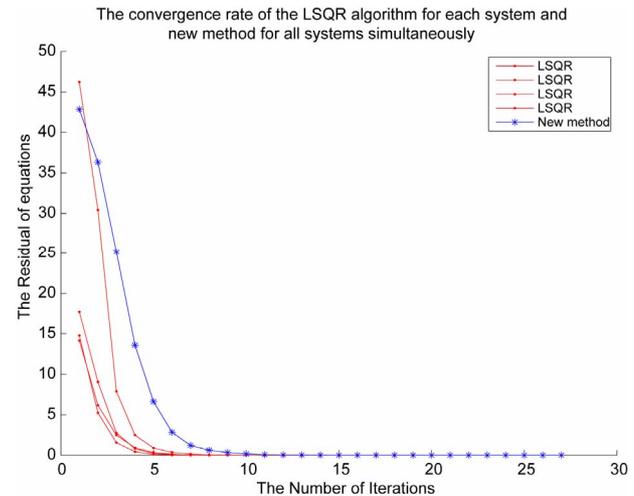


Figure 1. Convergence history of the LSQR algorithm and the new algorithm for the first set matrices with $s = 4$ and $n = 3000$.

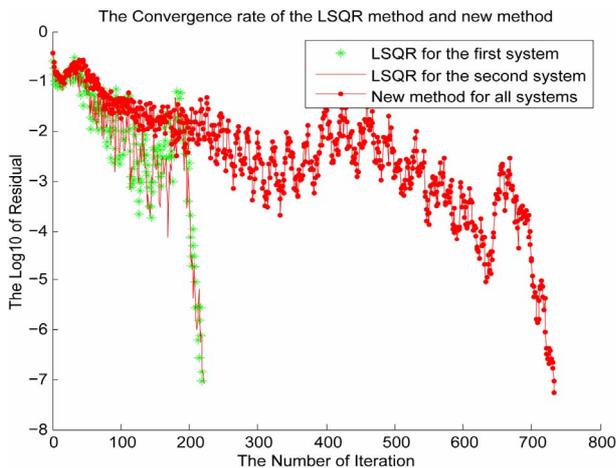


Figure 2. Convergence history of the LSQR algorithm and the new algorithm for the second set matrices with $s = 2$.

store the basis vectors, it is not needed to predetermine a subspace dimension and the approximate solutions and residuals are cheaply computed at every stage of the algorithm simultaneously because they are updated with short-term recurrence. Applying a reliable preconditioner for the linear systems of Equation (1) may increase the convergence rate, which has not been discussed in this paper.

6. Acknowledgements

The author would like to thank the anonymous referees for their helpful suggestions, which would greatly improve the article.

This work is financially supported by the research committee of Persian Gulf University.

REFERENCES

- [1] C. C. Paige and M. A. Saunders, "LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares," *ACM Transactions on Mathematical*, Vol. 8, No. 1, 1982, pp. 43-71. [doi:10.1145/355984.355989](https://doi.org/10.1145/355984.355989)
- [2] R. Plemmons, "FFT-Based RLS in Signal Processing," *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Minneapolis, 27-30 April 1993, pp. 571-574.
- [3] W. E. Boyse and A. A. Seidl, "A Block QMR Method for Computing Multiple Simultaneous Solutions to Complex Symmetric Systems," *SIAM Journal on Scientific Computing*, Vol. 17, No. 1, 1996, pp. 263-274.
- [4] R. Kress, "Linear Integral Equations," Springer-Verlag, New York, 1989. [doi:10.1007/978-3-642-97146-4](https://doi.org/10.1007/978-3-642-97146-4)
- [5] D. O'Leary, "The Block Conjugate Gradient Algorithm and Related Methods," *Linear Algebra and Its Applications*, Vol. 29, 1980, pp. 293-322. [doi:10.1016/0024-3795\(80\)90247-5](https://doi.org/10.1016/0024-3795(80)90247-5)
- [6] A. Jain, "Fundamentals of Digital Image Processing," Prentice-Hall, Englewood Cliffs, 1989.
- [7] S. Karimi and F. Toutounian, "The Block Least Squares Method for Solving Nonsymmetric Linear Systems with Multiple Right-Hand Sides," *Applied Mathematics and Computation*, Vol. 177, No. 2, 2006, pp. 852-862. [doi:10.1016/j.amc.2005.11.038](https://doi.org/10.1016/j.amc.2005.11.038)
- [8] G. Golub and C. Loan, "Matrix Computations," 2nd Edition, Johns Hopkins Press, Baltimore, 1989.
- [9] A. El Guennouni, K. Jbilou and H. Sadok, "The Block Lanczos Method for Linear Systems with Multiple Right-Hand Sides," *Applied Numerical Mathematics*, Vol. 51, No. 2-3, 2004, pp. 243-256.
- [10] A. El Guennouni, K. Jbilou and H. Sadok, "A Block Bi-CGSTAB Algorithm for Multiple Linear Systems," *Electronic Transactions on Numerical Analysis*, Vol. 16, 2003, pp. 129-142.
- [11] S. Karimi and F. Toutounian, "On the Convergence of the BL-LSQR Algorithm for Solving Matrix Equations," *International Journal of Computer Mathematics*, Vol. 88, No. 1, 2011, pp. 171-182. [doi:10.1080/00207160903365883](https://doi.org/10.1080/00207160903365883)
- [12] T. F. Chan and M. K. Ng, "Galerkin Projection Methods for Solving Multiple Linear Systems," *SIAM Journal on Scientific Computing*, Vol. 21, No. 3, 2012, pp. 836-850. [doi:10.1137/S1064827598310227](https://doi.org/10.1137/S1064827598310227)
- [13] F. Toutounian and S. Karimi, "Global Least Squares (GL-LSQR) Method for Solving General Linear Systems with Several Right-Hand Sides," *Applied Mathematics and Computation*, Vol. 178, No. 2, 2006, pp. 452-460. [doi:10.1016/j.amc.2005.11.065](https://doi.org/10.1016/j.amc.2005.11.065)