Journal of Mathematical Modeling
Vol. 1, No. 1, 2013, pp. 15-27

JMM

Global conjugate gradient method for solving large general Sylvester matrix equation

Saeed Karimi*

Department of Mathematics, Persian Gulf University, Bushehr, Iran
Emails: karimi@pgu.ac.ir, karimi214148@gmail.com

Abstract. In this paper, an iterative method is proposed for solving large general Sylvester matrix equation $AXB+CXD=E$, where $A \in \mathbb{R}^{n\times n}, C \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{s\times s}$ and $D \in \mathbb{R}^{s\times s}$ are given matrices and $X \in \mathbb{R}^{n\times s}$ is the unknown matrix. We present a global conjugate gradient (GL-CG) algorithm for solving linear system of equations with multiple right-hand sides. By defining a linear matrix operator and imposing some conditions on this operator, we demonstrate how to employ the GL-CG algorithm for solving large general Sylvester matrix equation. Finally, some numerical experiments are given to illustrate the efficiency of the method.

Keywords: Iterative method, General Sylvester matrix equation, CG method, Linear matrix operator.
AMS Subject Classification: 65F10, 65F50, 65Y20.

1 Introduction

We consider the large general Sylvester matrix equation

$$AXB + CXD = E, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}, C \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{s\times s}$ and $D \in \mathbb{R}^{s\times s}$ are given matrices and $X \in \mathbb{R}^{n\times s}$ is the unknown matrix. Matrix equations of the form (1) arise in

*Corresponding author.
Received: 16 October 2013 / Revised: 12 December 2013 / Accepted: 16 December 2013.

 http://research.guilan.ac.ir/jmm

many problems on scientific computing and engineering applications. For instances, in linear control and filtering theory for continuous or discrete-time large-scale dynamical systems, image restoration and other problems (see [2, 10, 11, 13, 14] and the references therein).

Several methods have been proposed to compute an approximate solution to Eq. (1) (for example, see [3, 4, 5, 9, 6, 7]). It is easy to verify that Eq. (1) is equivalent to the Kronecker form

$$(B^T \otimes A + D^T \otimes C)\text{vec}(X) = \text{vec}(E), \tag{2}$$

where $\otimes$ stands for the Kronecker product and $\text{vec}(Z) = (z_1^T, z_2^T, \ldots, z_m^T)^T$ for $Z = (z_1, z_2, \ldots, z_m) \in \mathbb{R}^{m\times n}$. The Kronecker equivalent form (2) can be solved by an iterative method.

Evidently, the size of the linear system (2) would be huge even for moderate values of n and s. Therefore, it is more preferable to employ an iterative method for solving the original system (1) instead of the linear system (2). Toutounian and Karimi [12] have proposed a global least squares(GL-LSQR) method for solving standard Sylvester equation. Recently, Beik and Salkuyeh [1] presented two global Krylov subspace methods for solving general coupled linear matrix equations. They defined a new inner product and its corresponding matrix norm for deriving a new version of the global FOM (GL-FOM) and global GMRES (GL-GMRES) methods.

For solving the large general Sylvester matrix Eq. (1), we define the following linear operator

$$\mathcal{L} : \mathbb{R}^{m\times n} \longrightarrow \mathbb{R}^{m\times n},$$
$$\mathcal{L}(X) = AXB + CXD.$$

It is well-known that its transpose is defined as $\mathcal{L}^T(X) = A^TXB^T + C^TXD^T$. Hence, the large general Sylvester matrix equation (1) is rewritten as the following linear operator equation

$$\mathcal{L}(X) = E. \tag{3}$$

In this paper, we present a new iterative method to compute an approximate solution of the linear matrix operator equation (3). This method is based on global conjugate gradient method, namely GL-CG method. At first, we propose GL-CG algorithm to solve the following linear system of equations with multiple right-hand sides

$$Ax_i = b_i, \quad i = 1, 2, \ldots, s. \tag{4}$$

Then we demonstrate how to employ the GL-CG method for solving the linear matrix equation (3).

Throughout this paper, we use the following notations. For two matrices X and Y in $\mathbb{R}^{n\times s}$, we consider the following inner product $< X, Y >_F = \mathrm{tr}(X^T Y)$, where tr(.) denotes the trace of a matrix. The associated norm is the Frobenious norm $\|.\|_F$.

The paper is organized as follows. In Section 2, we present GL-CG method to solve the multiple linear equations (4). Section 3 is devoted to implementation of GL-CG method for solving the linear matrix equation (3). Numerical examples are given in Section 4 and concluding remarks are presented in Section 5.

2 GL-CG method

The conjugate gradient (CG) algorithm is one of the best known iterative techniques for solving sparse symmetric positive definite (SPD) linear system of equations

$$Ax = b.$$

Described in one sentence, the CG method is a realization of an orthogonal projection technique onto the Krylov subspace

$$\mathcal{K}_m(r_0, A) = \mathrm{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\},$$

where r_0 is the initial residual. It is therefore mathematically equivalent to FOM method [8]. However, since A is symmetric, some simplifications resulting form the three-term Lanczos recurrence will lead to the following well-known version of CG. For more details about the CG method, one can refer to [8].

Algorithm 1. CG algorithm

1. Compute $r_0 = b - Ax_0,\ p_0 = r_0$
2. For $j = 0, 1, 2, \dots$ until convergence Do:
3. $\alpha_j = (r_j^T r_j)/(p_j^T A p_j)$
4. $x_{j+1} = x_j + \alpha_j p_j$
5. $r_{j+1} = r_j - \alpha_j A p_j$
6. $\beta_j = (r_{j+1}^T r_{j+1})/(r_j^T r_j)$
7. $p_{j+1} = r_{j+1} + \beta_j p_j$
8. EndDo

The multiple linear equations (4) can be rewritten as the matrix equation form

$$AX = B, \tag{5}$$

where $A \in \mathbb{R}^{n\times n}, X = [x_1, x_2, \ldots, x_s] \in \mathbb{R}^{n\times s}$ and $B = [b_1, b_2, \ldots, b_s] \in \mathbb{R}^{n\times s}$. In the sequel present the GL-CG algorithm for solving the sparse SPD linear matrix equation (5). The GL-CG method is a realization of an orthogonal projection technique onto block Krylov subspace

$$\mathbb{K}_m(R_0, A) = \text{span}\{R_0, AR_0, \ldots, A^{m-1}R_0\},$$

where R_0 is the initial residual $R_0 = B - AX_0$. This method is mathematically equivalent to global FOM method [10]. By using the Frobenious inner product $< ., . >_F$ and analogous to the CG algorithm, we have the following algorithm.

Algorithm 2. GL-CG algorithm

1. Compute $R_0 = B - AX_0$, and set $P_0 = R_0$
2. For $j = 0, 1, 2, \ldots,$ until convergence Do:
3. $\alpha_j = \frac{<R_j,R_j>_F}{<AP_j,P_j>_F}$
4. $X_{j+1} = X_j + \alpha_j P_j$
5. $R_{j+1} = R_j - \alpha_j AP_j$
6. $\beta_j = \frac{<R_{j+1},R_{j+1}>_F}{<R_j,R_j>_F}$
7. $P_{j+1} = R_{j+1} + \beta_j P_j$
8. EndDo

In step 3 of the GL-CG algorithm, we have

$$< AP_j, P_j >_F = \text{tr}(P_j^T AP_j) = \sum_{i=1}^{s} p_{j_i}^T A p_{j_i},$$

where p_{j_i} is the ith column of P_j. Also in step 6, we have

$$< R_j, R_j >_F = \text{tr}(R_j^T R_j) = \sum_{i=1}^{s} r_{j_i}^T r_{j_i},$$

where r_{j_i} is the ith column of R_j. Therefore, when the direct matrix P_j and the residual matrix R_j are different from zero matrix, the GL-CG algorithm has no breakdown because A is SPD.

3 Application of GL-CG method

In this section, we demonstrate how to employ the GL-CG method to get the approximate solution of the linear matrix operator equation (3). We begin with the following definition.

Definition 1. Let $\mathcal{L}$ be the following linear matrix operator

$$\mathcal{L} : \mathbb{R}^{n\times s} \longrightarrow \mathbb{R}^{n\times s}$$
$$\mathcal{L}(X) = AXB + CXD,$$

where $A, C \in \mathbb{R}^{n\times n}$ and $B, D, X \in \mathbb{R}^{n\times s}$. Then $\mathcal{L}$ is SPD whenever $\mathcal{L}$ is symmetric, i.e. $\mathcal{L} = \mathcal{L}^T$, and the following condition is satisfied

$$< \mathcal{L}(X), X >_F > 0, \quad 0 \neq X \in \mathbb{R}^{n\times s}.$$

Remark 1. Let A, B, C and D be symmetric matrices. Then the linear matrix operator (3) is symmetric.

Note that any nonsymmetric linear matrix operator (3) can be reduced to a symmetric one. Suppose $\mathcal{L}$ be nonsymmetric linear matrix operator, thus $\mathcal{L}^T \circ \mathcal{L}$ is a symmetric linear matrix operator, where $\circ$ is the combination of two operators. Therefore, we have the following remark. For convenience, we denote $\mathcal{L}^T \circ \mathcal{L}$ by $\mathcal{L}^T\mathcal{L}$.

Remark 2. Let $\mathcal{L}$ be the nonsymmetric linear matrix operator $\mathcal{L}(X) = AXB+CXD$. Then the following new linear matrix operator is symmetric

$$\hat{\mathcal{L}}(X) = \mathcal{L}^T\mathcal{L}(X).$$

According to the Remark 2, the nonsymmetric linear matrix operator equation (3) is equivalent to the symmetric linear matrix equation

$$\hat{\mathcal{L}}(X) = \hat{E}, \tag{6}$$

where $\hat{\mathcal{L}}(X) = A^TAXBB^T + A^TCXDB^T + C^TAXBD^T + C^TCXBB^T$ and $\hat{E} = A^TEB^T + C^TED^T$. In the special case of the general Sylvester matrix equation (1), when $B = I_s, C = I_n$, where I_k is identity matrix of size k, then the equation (3) will be in the form of

$$\mathcal{L}(X) = AX + XD = E. \tag{7}$$

Proposition 1. Suppose that both of the matrices A and D in Eq. (7) are symmetric positive semi-definite. If one of these matrices is also positive definite, then the linear matrix operator (7) is SPD.

Proof. Suppose X be an arbitrary matrix in $\mathbb{R}^{n\times s}$. Then we have $\mathrm{tr}(X^T\mathcal{L}(X)) = \mathrm{tr}(X^TAX)+\mathrm{tr}(XDX^T)$. Since at least one of the matrices A, D is positive definite, then $\mathrm{tr}(X^TAX)+\mathrm{tr}(XDX^T)$ is positive. Therefore, $\mathcal{L}$ is SPD. □

Proposition 2. *Let $\mathcal{L}$ be the general Sylvester matrix operator equation* (3). *Then $\mathcal{L}$ is SPD if and only if the coefficient matrix of the Kronecker equivalent form* (2) *is SPD.*

Proof. Let $\mathcal{L}$ be a SPD operator. According to definition 1, $\mathcal{L} = \mathcal{L}^T$ and $< \mathcal{L}(X), X >_F > 0$, for all $0 \neq X \in \mathbb{R}^{n\times s}$. If $X = [x_1, x_2, \ldots, x_s] \in \mathbb{R}^{n\times s}$,

$$< \mathcal{L}(X), X >_F = \mathrm{tr}(X^TAXB+X^TCXD) = \sum_{j=1}^{s}\sum_{i=1}^{s} x_j^T(b_{ji}A+d_{ji}C)x_i, \quad (8)$$

where b_{kl} and d_{kl}, for $k, l = 1, 2, \ldots, s$ are the components of the matrices B and D, respectively. One can easily verify that the last equation (8) is equal to

$$\mathrm{vec}(X)^T(B \otimes A + D \otimes C)\mathrm{vec}(X). \quad (9)$$

Therefore, from Remark 1 the proof is completed. □

Now we want to implement the GL-CG algorithm for solving the linear matrix operator equation (3). Without loss of generality, we assume that $\mathcal{L}$ is symmetric. Also we assume that $\mathcal{L}$ is positive definite. The same as the GL-CG algorithm, we have the following algorithm to solve the SPD operator equation (3).

Algorithm 3.

1. Choose X_0 from $\mathbb{R}^{n\times s}$
2. Compute $R^{(0)} = E - \mathcal{L}(X_0),\ P_0 = R_0$
3. For $j = 0, 1, 2, \ldots$ until convergence Do:
4. $\quad \alpha_j = \frac{<R_j,R_j>_F}{<\mathcal{L}(P_j),P_j>_F}$
5. $\quad X_{j+1} = X_j + \alpha_j P_j$
6. $\quad R_{j+1} = R_j - \alpha_j \mathcal{L}(P_j)$
7. $\quad \beta_j = \frac{<R_{j+1},R_{j+1}>_F}{<R_j,R_j>_F}$
8. $\quad P_{j+1} = R_{j+1} + \beta_j P_j$
9. EndDo

Consider the following multiple linear matrix equations

$$\sum_{j=1}^{k} A_{ij} X_i B_{ij} = C_i, \quad i = 1, 2, \ldots, p. \tag{10}$$

We define the following linear matrix operators

$$\mathcal{L}_i(X_i) = \sum_{j=1}^{k} A_{ij} X_i B_{ij}, \quad i = 1, 2, \ldots, p. \tag{11}$$

Therefore, the multiple linear matrix equation (10) can be written as follows

$$\mathcal{L}_i(X_i) = C_i, \quad i = 1, 2, \ldots, k. \tag{12}$$

We assume that $\mathcal{L}_i,\ i = 1, 2, \ldots, k$, are SPD operators, then we can apply in parallel the GL-CG method to obtain the approximate solutions of the multiple linear matrix operator equations (12). We note that the general Sylvester matrix equation (1) is a special case of the multiple linear matrix equations (10), when $p = 1$ and $k = 2$. Also the following coupled Lyapunov equation is the special case of (10)

$$\begin{cases} AX + XA^T + BB^T = 0, \\ A^T Y + YA + C^T C = 0, \end{cases} \tag{13}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times s}$ and $C \in \mathbb{R}^{r\times n}$ $(r, s << n)$ are given and $X, Y \in \mathbb{R}^{n\times n}$ are unknown. This coupled equation can be rewritten to the coupled operator form

$$\begin{cases} \mathcal{L}(X) = -BB^T, \\ \mathcal{L}^T(Y) = -C^T C. \end{cases}$$

We can apply the GL-CG method in parallel to solve the previous equation. As we mentioned in Section 1, one can apply the CG method. However, it may be too expensive especially when the coefficient matrices are dense.

4 Numerical experiments

In this section, some numerical examples are presented to illustrate the effectiveness of the GL-CG algorithm to solve (1). All the numerical experiments were computed in double precision with some MATLAB codes codes on a Pentium 4 PC, with a 3 GHz CPU and 4 GB of RAM.. For

Table 1: Numerical results of the GL-CG method for Example 1.

M-category	It	CPU-time	Err	cond(A)	cond(B)
(a)	418	2.17	9.89×10^{-8}	1.35×10^{5}	1.30×10^{3}
(b)	12	0.19	9.22×10^{-8}	7.59	2.71
(c)	1601	10.86	9.85×10^{-8}	7.59	1.30×10^{3}

all the examples the initial guess X_0 is the zero matrix and the right-hand side matrix is taken such that the exact solution is `ones(n,s)`, where the function `ones` creates a matrix of appropriate size of all ones. For all the test the stopping

$$\frac{||R_j||_F}{||R_0||_F} < \delta,$$

is used, where R_j and R_0 are the jth and initial residuals of the GL-CG algorithm, respectively. We always use $\delta = 10^{-7}$, except for the second case of Example 5, where we set $\delta = 10^{-10}$. In all tables, It, CPU-time, Err and cond(S) denote the number of iterations, iteration time, relative residual norm and condition number of matrix S, respectively.

Example 1. In this example we have considered the general Sylvester matrix equation (1) where

$$\begin{aligned}
&(a) \quad A = C = \text{pentadiag}(-2, -1, 6, -1, -2), \quad && B = D = \text{tridiag}(-1, 2, -1),\\
&(b) \quad A = C = \text{pentadiag}(-2, -1, 6, 1, 2), && B = D = \text{tridiag}(-1, 2, 1),\\
&(c) \quad A = C = \text{pentadiag}(-2, -1, 6, 1, 2), && B = D = \text{tridiag}(-1, 2, -1).
\end{aligned}$$

The pentadiagonal matrices are 900×900 and the tridiagonal matrices are 50×50. By Proposition 2, we find that the operator $\mathcal{L}$ is SPD, in the case of (a), and we can apply the GL-CG algorithm to obtain the approximate solution. However, the operator $\mathcal{L}$ is not SPD, in the case of (b) and (c), and therefore, we should use the transpose operator $\mathcal{L}^T$ and solve Eq. (6). The numerical results are given in Table 1. In this table, the M-category column refers to the cases (a), (b) and (c).

Example 2. Consider the following elliptic operator

$$L(u) = -\Delta u + 2\nu u_x + 2\nu u_y,$$

with Dirichlet boundary conditions [12]. The operator was discretized using central finite differences on $[0, 1] \times [0, 1]$, with mesh size $h = \frac{1}{n+1}$ in the x-direction and $k = \frac{1}{s+1}$ in the y-direction. This yields a linear system of

Table 2: Numerical results of the GL-CG algorithm for Example 2.

ν	It	CPU-time	Err	cond(A)
10	926	16.65	9.87×10^{-8}	2.07×10^{6}
50	226	4.31	9.65×10^{-8}	4.69×10^{5}

Table 3: Numerical results of the GL-CG and Kron-CG methods for Example 3.

Method	It	CPU-time	Err
GL-CG	706	0.29	8.68×10^{-8}
Kron-CG	20170	12.80	9.12×10^{-8}

algebraic equations that can be written as an standard Sylvester equation (7), in which A and D are tridiagonal matrices of dimensions $n = 3600$ and $s = 25$ as the following

$$A = \text{tridiag}(-1 - \nu h, 2, -1 + \nu h) \quad \text{and} \quad D = \text{tridiag}(-1 - \nu k, 2, -1 + \nu k).$$

We note that with this choices of A and D, the $\mathcal{L}$ operator is not SPD. In this case, we have solved (6), by the GL-CG algorithm. The numerical results are given in Table 2 for parameter $\nu = 10, 50$.

As we mentioned in the previous section, the GL-CG method for solving Eq. (1) is more robust than applied CG method for solving Kronecker equivalent form (2), when the coefficient matrices are dense. To illustrate this we present the next example.

Example 3. Let

$$A = C = \text{rand}(50), \qquad \text{and} \quad B = D = \text{rand}(10),$$

where the function rand(m) generates a random matrix of order m. We consider these matrices as the coefficient matrices of the matrix equation (1). Therefore, the coefficient matrices are rather dense. Numerical results are given in Table 3. In this table, Kron-CG is the information of the CG method for solving the Kronecker equivalent form of (6). The results show the robustness of the GL-CG method.

Example 4. Consider the following Lyapunov equation

$$AX + XA^T = E,$$

where A is 1000×1000 tridiagonal matrix with diagonal entries 0.01, 0.1, 1, 2, 3, 4, …, 998 and -1's on the lower and upper diagonal.

We have applied the GL-LSQR [12] and GL-CG algorithms to obtain the approximate solution of Example 4. The convergence information is shown in Figure 1, however, GL-LSQR converges more smoothly.

Example 5. (See [1]) Consider the following matrix equation

$$AXB + X^T = E. \tag{14}$$

We have applied the Gl-CG to solve this matrix equation for the following coefficient matrices: For the first case, let

$$A = \begin{pmatrix} 1 & 6 & -2 & -9 & 2 \\ 3 & -14 & -6 & 21 & 6 \\ 0 & 12 & 0 & -18 & 0 \\ -5 & 10 & 10 & -15 & -10 \\ 9 & 8 & -18 & -12 & 18 \\ 3 & -16 & -6 & 24 & 6 \end{pmatrix}, \quad B = \begin{pmatrix} -12 & -1 & 5 & 11 & -3 \\ 3 & -14 & -6 & 2 & 15 \\ 0 & 3 & 1 & -1 & -3 \\ -27 & -18 & 6 & 30 & 9 \\ 24 & -13 & -15 & -17 & 21 \\ -15 & -14 & 2 & 18 & 9 \end{pmatrix},$$

and in the second case, we consider the m-by-m matrices

$$A = \begin{pmatrix} 3 & -1 & & & -1 \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 3 & -1 \\ -1 & & & -1 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 5 & -2 & & & -2 \\ -2 & 5 & -2 & & \\ & \ddots & \ddots & \ddots & \\ & & -2 & 5 & -2 \\ -2 & & & -2 & 5 \end{pmatrix}.$$

We have compared the GL-CG algorithm with Algorithm 1 in [1] for the first case of Example 5. Numerical results of both algorithms are given in Table 4. As seen the results of both of the algorithms are similar. In these tables, k, r_k and e_k are iteration, residual norm and relative error respectively. Also we have compared the CPU time(s) of both algorithms for the second case. The results are given in Table 5. In this table, m is the order of matrices A and B. As we observe, the GL-CG algorithm converges faster than Algorithm 1.

Table 4: Numerical results of the GL-CG algorithm and Algorithm 1 for Example 4.

	GL-CG		Algorithm 1	
k	r_k	e_k	r_k	e_k
5	6.53×10^{-1}	3.16×10^{-1}	5.437	6.542
10	2.32×10^{-3}	1.97×10^{-1}	2.08×10^{-1}	4.30×10^{-2}
15	1.02×10^{-5}	9.96×10^{-5}	1.10×10^{-3}	6.96×10^{-5}
20	2.52×10^{-7}	4.60×10^{-8}	2.56×10^{-8}	1.40×10^{-10}
25	3.19×10^{-10}	4.42×10^{-10}	3.19×10^{-14}	6.53×10^{-14}
30	5.54×10^{-12}	4.38×10^{-10}	3.77×10^{-16}	6.53×10^{-14}

Table 5: The CPU time(s) of the GL-CG algorithm and Algorithm 1 for Example 4.

m	20	30	40	50	60
GL-CG	6.6×10^{-3}	1.1×10^{-2}	4.8×10^{-2}	5.2×10^{-2}	6.0×10^{-2}
Algorithm 1	0.210	1.31	4.17	10.6	27.1

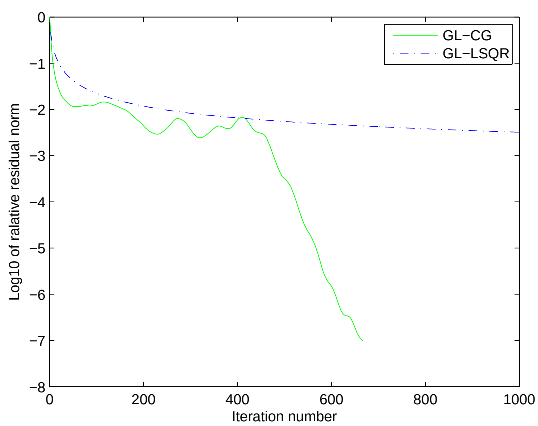

Figure 1: Convergence information of the GL-LSQR and GL-CG algorithms for Example 4.

5 Conclusion

We have proposed GL-CG method for solving multiple linear systems $Ax_i = b_i$, for $1 \leq i \leq s$, where the coefficient matrix A is an SPD matrix. We have demonstrated how to apply the GL-CG algorithm to solve the general Sylvester matrix equation. The numerical results have shown the efficiency of the new method.

Acknowledgements

The author would like to thank the referee very much for helpful comments and suggestions for revising this manuscript.

References

[1] F.P.A. Beik and D.K. Salkuyeh, *On the global Krylov subspace methods for solving general coupled matrix equation*, Comput. Math. Appl. **62** (2011) 4605–4613.

[2] A. Bouhamidi and K. Jbilou, *A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications*, Appl. Math. Comput. **206** (2008) 687–694.

[3] F. Ding and T. Chen, *Iterative least-squares solutions of coupled Sylvester matrix equations*, Syst. Control Lett. **54** (2005) 95-107.

[4] F. Ding and T. Chen, *On iterative solutions of general coupled matrix equations*, SIAM J. Control Optim. 44 (2006) 2269-2284.

[5] F. Ding and T. Chen, *Gradient based iterative algorithms for solving a class of matrix equations*, IEEE T. Automat. Control **50** (2005) 1216-1221.

[6] J. Ding, Y.J. Liu and F. Ding, *Iterative solutions to matrix equations of form $A_iXB_i = F_i$*, Comput. Math. Appl. **59** (2010) 3500-3507.

[7] F. Ding, P.X. Liu and J. Ding, *Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle*, Appl. Math. Comput. 197 (2008) 41-50.

[8] G. Golub and C. Loan, *Matrix Computations*, 2nd ed., Johns Hopkins Press, Bultimore, MD., 1989.

[9] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1991.

[10] K. Jbilou, A. Messaudi and H. Sadok, *Global FOM and GMRES algorithms for matrix equations*, Appl. Numer. Math. **31** (1999) 49–63.

[11] K. Jbilou and A.J. Riquet, *Projection methods for large Lyapunov matrix equations*, Linear Algebra Appl. **415** (2006) 344–358.

[12] F. Toutounian and S. Karimi, *Global Least squares method (GL-LSQR) for solving general linear systeams with several right-hand sides*, Appl. Math. Comput. **178** (2006) 452-460.

[13] J.J. Zhang, *A note on the iterative solutions of general coupled matrix equation*, Appl. Math. Comput. **217** (2011) 8386–9380.

[14] B. Zhou and G.R. Duan, *On the generalized Sylvester mapping and matrix equation*, Systems Control Lett. **57** (2008) 200–208.