Electronic Transactions on Numerical Analysis.
Volume 43, pp. 45-59, 2014.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

A MINIMAL RESIDUAL NORM METHOD FOR LARGE-SCALE SYLVESTER MATRIX EQUATIONS*

SAID AGOUJIL†, ABDESLEM H. BENTBIB‡, KHALIDE JBILOU§, AND EL MOSTAFA SADEK‡§

Abstract. In this paper, we present a new method for solving large-scale Sylvester matrix equations with a low-rank right-hand side. The proposed method is an iterative method based on a projection onto an extended block Krylov subspace by minimization of the norm of the residual. The obtained reduced-order problem is solved via different direct or iterative solvers that exploit the structure of the linear operator associated with the obtained matrix equation. In particular, we use the global LSQR algorithm as iterative method for the derived low-order problem. Then, when convergence is achieved, a low-rank approximate solution is computed given as a product of two low-rank matrices, and a stopping procedure based on an economical computation of the norm of the residual is proposed. Different numerical examples are presented, and the proposed minimal residual approach is compared with the corresponding Galerkin-type approach.

Key words. extended block Krylov subspaces, low-rank approximation, Sylvester equation, minimal residual methods

AMS subject classifications. 65F10, 65F30

1. Introduction.

In this paper, we consider the large-scale Sylvester matrix equation

$$AX + XB + EF^T = 0, \tag{1.1}$$

where A and B are square matrices of size $n \times n$ and $s \times s$, respectively, and E, F are matrices of size $n \times r$ and $s \times r$, respectively.

Sylvester equations play a fundamental role in many problems in control, communication theory, image processing, signal processing, filtering, model reduction problems, decoupling techniques for ordinary partial differential equations, and the implementation of implicit numerical methods for ordinary differential equations; see, e.g., [3, 6, 10, 13] and the references therein. Sylvester matrix equations have a unique solution if and only if $\alpha + \beta \neq 0$ for all $\alpha \in \sigma(A)$ and $\beta \in \sigma(B)$, where $\sigma(Z)$ denotes the spectrum of the matrix Z. For small- to medium-sized problems, direct methods such as the Bartels-Stewart [2] and the Hessenberg-Schur [8] algorithms can be used. These algorithms are based on reducing A and B into triangular or Hessenberg forms.

During the last years, several projection methods based on Krylov subspaces have been proposed; see, e.g., [5, 6, 10, 12, 13, 14, 18, 23] and the references therein. The main idea employed in these methods is to use a block (global or extended) subspace and then apply the block (global or extended block) Arnoldi process to construct orthonormal bases. Then, the original large Sylvester matrix equation is projected onto these Krylov subspaces. The alternating directional implicit (in short ADI) iterations [3, 4, 16] can also be utilized if spectral information about A and B is given. Note that, ADI iterations allow for faster convergence if optimal shifts to A and B can be effectively computed and linear systems with shifted coefficient matrices are solved effectively at low cost.

*Received October 30, 2013. Accepted May 6, 2014. Published online on September 5, 2014. Recommended by L. Reichel.

†Laboratory LAMAI, Département of Computer Science, Faculty of Sciencs and Technology, B.P 509 Errachidia 52000, Morocco (agoujil@gmail.com).

‡Faculté des Sciences et Techniques-Gueliz, Laboratoire de Mathématiques Appliquées et Informatique, Morocco (a.bentbib@uca.ma, sadek@lmpa.univ-littoral.fr).

§Laboratoire LMPA, 50 rue F. Buisson, ULCO, 62228, Calais, France (jbilou@lmpa.univ-littoral.fr).

ETNA
Kent State University
http://etna.math.kent.edu

The approximate solutions produced by Krylov subspace methods are given as $X_m = \mathbb{V}_m Y_m \mathbb{W}_m^T$, where $\mathbb{V}_m$ and $\mathbb{W}_m$ are orthonormal matrices whose columns form bases for the Krylov subspace $\mathcal{K}_m(A, E)$ and $\mathcal{K}_m(B^T, F)$, respectively. Then, the Sylvester matrix equation is projected via the following Galerkin condition

$$\mathbb{V}_m^T(A\mathbb{V}_m Y_m \mathbb{W}_m^T + \mathbb{V}_m Y_m \mathbb{W}_m^T B + EF^T)\mathbb{W}_m = 0.$$

The obtained low-order Sylvester matrix equation is solved by direct methods such as the Hessenberg-Schur method. Instead of using a Galerkin-type projection, in this paper we consider the minimization of the Frobenius norm of the residual associated with the Sylvester matrix equation (1.1), which leads to the following minimization problem

$$Y_m^{MR} = \underset{X_m = \mathbb{V}_m Y_m \mathbb{W}_m^T}{\operatorname{argmin}} \left\| AX_m + X_m B + EF^T \right\|_F .$$

The rest of the paper is organized as follows. In the next section, we recall the extended block Arnoldi algorithm with some of its properties. In Section 3, we define the minimal residual method for Sylvester matrix equations by using the extended Krylov subspaces $\mathcal{K}_m(A, E)$ and $\mathcal{K}_m(B^T, F)$. Then, in Section 4, we give some direct and iterative methods for solving the obtained low-order minimization problem. Finally, Section 5 is devoted to numerical experiments.

Throughout the paper, we use the following notations. The Frobenius inner product of the matrices X and Y is defined by $\langle X, Y\rangle_F = \operatorname{tr}(X^T Y)$, where $\operatorname{tr}(Z)$ denotes the trace of a square matrix Z. The associated norm is the Frobenius norm denoted by $\|.\|_F$. The Kronecker product of two matrices A and B is defined by $A \otimes B = [a_{i,j}B]$, where $A = [a_{i,j}]$. This product satisfies the properties $(A \otimes B)(C \otimes D) = (AC \otimes BD)$ and $(A \otimes B)^T = A^T \otimes B^T$. Finally, if X is a matrix, then $\operatorname{vec}(X)$ is the vector obtained by stacking all the columns of X.

2. The extended block Arnoldi process. In this section, we recall the extended Krylov subspace and extended block Arnoldi process. Let V be a matrix of dimension $n \times r$. Then the block Krylov subspace associated to (A, V) is defined as

$$\mathbb{K}_m(A, V) = \operatorname{Range}\left(\left\{V, AV, A^2V, \ldots, A^{m-1}V\right\}\right).$$

Assuming that the matrix A is nonsingular, the extended block Krylov subspace associated with (A, A^{-1}, V) is given by (see [5, 23])

$$\begin{aligned}\mathcal{K}_m^e(A, V) &= \operatorname{Range}\left(\left\{V, A^{-1}V, AV, A^{-2}V, A^2V, \ldots, A^{m-1}V, A^{-m+1}V\right\}\right)\\ &= \mathbb{K}_m(A, V) + \mathbb{K}_m(A^{-1}, A^{-1}V).\end{aligned}$$

The extended block Arnoldi process allows us to construct an orthonormal basis for the extended block Krylov subspace $\mathcal{K}_m^e(A, V)$; see [9, 23].

ETNA
Kent State University
http://etna.math.kent.edu

ALGORITHM 1. The extended block Arnoldi algorithm (EBA).

Input: A an $n \times n$ matrix, V an $n \times r$ matrix, and m an integer.

1. Compute the QR decomposition of $[V, A^{-1}V]$, i.e., $[V, A^{-1}V] = V_1\Lambda$.
2. Set $\mathcal{V}_0 = [\,]$.
3. For $j = 1, 2, ..., m$
   (a) Set $V_j^{(1)}$: first r columns of V_j, $V_j^{(2)}$: second r columns of V_j.
   (b) $\mathbb{V}_j = [\mathbb{V}_{j-1}, V_j]$, $\hat{V}_{j+1} = \left[A\,V_j^{(1)}, A^{-1}\,V_j^{(2)}\right]$
   (c) Orthogonalize $\hat{V}_{j+1}$ with respect to $\mathbb{V}_j$ to get V_{j+1}, i.e.,
      i. For $i = 1, 2, \ldots$
      ii. $H_{i,j} = V_i^T\,\hat{V}_{j+1}$
      iii. $\hat{V}_{j+1} = \hat{V}_{j+1} - V_i\,H_{i,j}$
      iv. End For
   (d) Compute the QR decomposition of $\hat{V}_{j+1}$, i.e., $\hat{V}_{j+1} = V_{j+1}\,H_{j+1,j}$.
4. End For

Since the extended block Arnoldi algorithm involves the Gram-Schmidt process, the obtained block vectors $\mathbb{V}_m = [V_1, V_2, \ldots, V_m]$ ($V_i \in \mathbb{R}^{n\times 2r}$) have their columns mutually orthogonal provided none of the upper triangular matrices $H_{j+1,j}$ are rank deficient. After m steps, Algorithm 1 has built an orthonormal basis $\mathbb{V}_m$ of the extended block Krylov subspace $\mathrm{Range}(\{V, A\,V, \ldots, A^{m-1}\,V, A^{-1}\,V, \ldots, (A^{-1})^m\,V\})$ and a block upper Hessenberg matrix $\mathbb{H}_m$ whose non zeros blocks are the entries $H_{i,j}$. Note that each submatrix $H_{i,j}$ ($1 \le i \le j \le m$) is of order $2r$.

Let $\mathbb{T}_m \in \mathbb{R}^{2mr\times 2mr}$ be the restriction of the matrix A to the extended Krylov subspace $\mathcal{K}_m^e(A, V)$, i.e., $\mathbb{T}_m = \mathbb{V}_m^T\,A\,\mathbb{V}_m$. It is shown in [23] that $\mathbb{T}_m$ is also block upper Hessenberg with $2r \times 2r$ blocks. Moreover, a recursion is derived to compute $\mathbb{T}_m$ from $\mathbb{H}_m$ without requiring matrix-vector products with A. For more details on how to compute $\mathbb{T}_m$ from $\mathbb{H}_m$, we refer to [23]. We note that for large problems, the inverse of the matrix A is not computed explicitly, and in this case we can use iterative solvers with preconditioners to solve linear systems involving A. We notice that Algorithm 1 could suffer from a possible breakdown if for some j the upper triangular matrix $H_{j+1,j}$ is rank deficient. To avoid this problem, one may define a sequential extended block Arnoldi with a deflation procedure in the same manner as it was already done for the classical block Arnoldi algorithm.

PROPOSITION 2.1 ([9]). *Let $\bar{\mathbb{T}}_m = \mathbb{V}_{m+1}^T A\mathbb{V}_m$. Then we have the following relations*

$$\begin{aligned} A\mathbb{V}_m &= \mathbb{V}_{m+1}\bar{\mathbb{T}}_m \\ &= \mathbb{V}_m\mathbb{T}_m + V_{m+1}T_{m+1,m}\mathbb{E}_m^T, \end{aligned}$$

where $\mathbb{E}_m = [0_{2(m-1)r\times 2r}, I_{2r}]^T$ is the matrix of the last $2r$ columns of the identity matrix I_{2mr}.

In the next section, we define the minimal residual approach for solving Sylvester matrix equations. For this we can use block, global or extended Krylov subspaces. We will focus only on extended Krylov subspace methods; the other approaches could be used in the same manner.

3. The minimal residual method for large-scale Sylvester matrix equations. In the last years, several projection methods based on Krylov subspaces have been proposed to produce approximations to exact solutions of Lyapunov or Sylvester equations; see, e.g., [6, 12, 13, 14, 22, 23]. Most of these projection methods use the Galerkin condition to extract the approximate solutions from the projection space.

ETNA
Kent State University
http://etna.math.kent.edu

In this section, we consider approximate solutions of the form

$$X^{MR} = \mathbb{V}_m Y_m^{MR} \mathbb{W}_m^T,$$

where Y_m^{MR} solves the following low-order minimization problem

$$Y^{MR} = \underset{X_m = \mathbb{V}_m Y_m \mathbb{W}_m^T}{\operatorname{argmin}} \left\| AX_m + X_m B + EF^T \right\|_F,$$

where $\mathbb{V}_m = [V_1, V_2, \cdots, V_m]$, $\mathbb{W}_m = [W_1, W_2, \cdots, W_m] \in \mathbb{R}^{n \times mr}$ are the orthonormal matrices constructed by applying simultaneously m steps of the extended block Arnoldi algorithm to the pairs (A, E) and (B^T, F), respectively. In this case, we have

$$A\mathbb{V}_m = \mathbb{V}_{m+1} \bar{\mathbb{T}}_m^A,$$
$$B^T \mathbb{W}_m = \mathbb{W}_{m+1} \bar{\mathbb{T}}_m^B.$$

We have the following result.

THEOREM 3.1. *Let $\mathbb{V}_m$ and $\mathbb{W}_m$ be the orthonormal matrices constructed by applying simultaneously m steps of the extended block Arnoldi algorithm to the pairs (A, E) and (B^T, F), respectively. Then the minimization problem*

$$Y_m^{MR} = \underset{X_m = \mathbb{V}_m Y_m \mathbb{W}_m^T}{\operatorname{argmin}} \left\| AX_m + X_m B + EF^T \right\|_F,$$

can be written as

$$Y_m^{MR} = \operatorname{argmin} \left\| \bar{\mathbb{T}}_m^A Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m (\bar{\mathbb{T}}_m^B)^T + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F, \tag{3.1}$$

where $E = V_1 R_E$ and $F = W_1 R_F$ are the QR-factorizations of E and F, respectively.

Proof. We have

$$\begin{aligned}
&\min_{X = \mathbb{V}_m Y_m \mathbb{W}_m^T} \left\| AX + XB + EF^T \right\|_F \\
&= \min_{Y_m} \| A\mathbb{V}_m Y_m \mathbb{W}_m^T + B\mathbb{V}_m Y_m \mathbb{W}_m^T + V_1 R_E R_F^T W_1^T \|_F \\
&= \min_{Y_m} \left\| \mathbb{V}_{m+1} \left(\bar{\mathbb{T}}_m^A Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m (\bar{\mathbb{T}}_m^B)^T + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right) \mathbb{W}_{m+1}^T \right\|_F \\
&= \min_{Y_m} \left\| \bar{\mathbb{T}}_m^A Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m (\bar{\mathbb{T}}_m^B)^T + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F. \qquad \square
\end{aligned}$$

Notice that the computation of the norm of the residual $R(X_m)$ requires only the knowledge of Y_m^{MR} but does not require the approximate solution X_m, which is computed only when convergence is achieved. This residual norm can be calculated by

$$r_m = \left\| \bar{\mathbb{T}}_m^A Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m (\bar{\mathbb{T}}_m^B)^T + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F. \tag{3.2}$$

Recall that the Galerkin approach produces approximations $X_m^{GA} = \mathbb{V}_m Y_m^{GA} \mathbb{W}_m^T$ where Y_m^{GA} is a solution of the low-order Sylvester matrix equation

$$\mathbb{T}_m^A Y_m^{GA} + Y_m^{GA} (\mathbb{T}_m^B)^T + \widetilde{E}_1 \widetilde{F}_1^T = 0, \tag{3.3}$$

where $\widetilde{E}_1 = \mathbb{V}_m^T E$ and $\widetilde{F}_1 = \mathbb{W}_m^T F$. The projected problem (3.3) has a unique solution if and only if the matrices $\mathbb{T}_m^A$ and $\mathbb{T}_m^{B^T}$ have no eigenvalue in common. Notice that the problem (3.1) does not have this restriction. Now, the main issue is how to solve the reduced-order minimization problem (3.1). In the next section, we describe different direct and iterative strategies for solving (3.1). These techniques were inspired from those in [18] for Lyapunov matrix equations.

ETNA
Kent State University
http://etna.math.kent.edu

4. Methods for solving the reduced minimization problem.

4.1. The least squares approach associated with the Kronecker form. Using the Kronecker product properties, problem (4.3) can be written as

$$\min_y \left\| \left(\begin{bmatrix} I \\ 0 \end{bmatrix} \otimes \bar{\mathbb{T}}_m^A + \bar{\mathbb{T}}_m^B \otimes \begin{bmatrix} I \\ 0 \end{bmatrix} \right) y + c \right\|_F, \tag{4.1}$$

where $c = \operatorname{vec}\left(\begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right)$ and $y = \operatorname{vec}(Y)$. This least squares problem is then solved by direct methods such as the classical QR algorithm. For small values of the iteration number m, this gives good results. However, when m increases, the method becomes very slow and cannot be used.

4.2. The Hu-Reichel method for solving the reduced problem. Here we discuss the Hu-Reichel method (see [11]) for solving the reduced-order matrix least squares problem (3.1). More precisely, we apply the approach given in [15] and the strategy used in [18].

Let $\mathbb{T}_m^A = U T_A U^T$ and $\mathbb{T}_m^B = V T_B V^T$ be the real Schur decompositions of the matrices $\mathbb{T}_A$ and $\mathbb{T}_B$, respectively. The minimization problem (3.1) is written as

$$\min_Y \left\| \begin{bmatrix} \mathbb{T}_m^A \\ h_A \end{bmatrix} Y \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y \begin{bmatrix} \mathbb{T}_m^B \\ h_B \end{bmatrix}^T + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F,$$

where h_A and h_B represent the $2r$ last rows of the matrices $\overline{\mathbb{T}}_m^A$ and $\overline{\mathbb{T}}_m^B$, respectively. Then we obtain the new minimization problem

$$\min_Y \left\| \begin{bmatrix} \mathbb{T}_m^A Y + Y {\mathbb{T}_m^B}^T + R_E R_F^T & 0 \\ h_A Y & Y h_B^T \end{bmatrix} \right\|_F,$$

which is equivalent to

$$\begin{aligned} &\min_Y \left(\left\| \mathbb{T}_m^A Y + Y {\mathbb{T}_m^B}^T + R_E R_F^T \right\|_F^2 + \|h_A Y\|_F^2 + \left\| Y h_B^T \right\|_F^2 \right) \\ =&\min_Y \left(\left\| U T_A U^T Y + Y (V T_B V^T)^T + R_E R_F^T \right\|_F^2 + \|h_A Y\|_F^2 + \left\| Y h_B^T \right\|_F^2 \right) \\ =&\min_{\widetilde{Y}} \left(\left\| T_A \widetilde{Y} + \widetilde{Y} T_B^T + U^T R_E R_F^T V \right\|_F^2 + \left\| h_A U \widetilde{Y} \right\|_F^2 + \left\| \widetilde{Y} V^T h_B^T \right\|_F^2 \right), \end{aligned}$$

where $\widetilde{Y} = U^T Y V$. Using the Kronecker product, the preceding problem is transformed into

$$\min_Y \left\| \begin{bmatrix} I \otimes T_A + T_B \otimes I \\ I \otimes h_A U \\ h_B V \otimes I \end{bmatrix} \operatorname{vec}(\widetilde{Y}) + \begin{bmatrix} \operatorname{vec}(U^T R_E R_F^T V) \\ 0 \\ 0 \end{bmatrix} \right\|_2^2. \tag{4.2}$$

Problem (4.2) can also be expressed as

$$\min_{\tilde{y}} \left\| \begin{bmatrix} R \\ S \end{bmatrix} \tilde{y} + \begin{bmatrix} d \\ 0 \end{bmatrix} \right\|_2^2,$$

where $R = I \otimes T_A + T_B \otimes I$ and $S = \begin{bmatrix} I \otimes h_A U \\ h_B V \otimes I \end{bmatrix}$.

ETNA
Kent State University
http://etna.math.kent.edu

The associated normal equation is $(R^TR + S^TS)\tilde{y} + R^Td = 0$. If the matrix R is nonsingular, then we have

$$(I + (SR^{-1})^TSR^{-1})z + d = 0, \qquad \text{where } z = R\tilde{y}.$$

Now, using the Sherman-Morrison-Woodbury formula, we get

$$z = -d + (SR^{-1})^T(I + SR^{-1}(SR^{-1})^T)^{-1}SR^{-1}d.$$

Therefore from z, we obtain $\widetilde{Y}$ and then Y, the solution of problem (3.1). We notice that an economical strategy for computing SR^{-1} is also possible and can be obtained in the same way as in [15, 18]. Furthermore, in its matrix form, problem (4.2) could also be solved by the global LSQR or by the global CG method with an appropriate preconditioner.

REMARK 4.1. Both the Kronecker least squares approach and the Hu-Reichel method require $O(m^3r^3)$ multiplications which is similar to the amount of computations required at each step of the Galerkin method when using the Bartels-Stewart algorithm for solving the low-order Sylvester equation (3.3). When m increases, the least squares approach and the Hu-Reichel method become expensive. In these cases, one should use iterative methods as will be defined in the next two subsections.

4.3. The global LSQR algorithm. In this section, we demonstrate how to adapt the LSQR algorithm of Paige and Sanders [19] for solving the low-order least squares problem (3.1). The classical LSQR algorithm is analytically equivalent to the conjugate gradient method applied to the associated normal equation. The LSQR method makes use of the Golub-Kahan bidiagonalization process [7].

At each step m of the process, we determine approximate solutions to the solution Y_m^{MR} of the problem (3.1). For simplicity, problem (3.1) is now written in the following way,

$$\min_Y \|\mathcal{L}_m(Y) - \mathcal{C}\|_F, \tag{4.3}$$

where

$$\mathcal{L}_m(Y) = \bar{\mathbb{T}}_m^A Y \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y \bar{\mathbb{T}}_m^{BT}, \tag{4.4}$$

and

$$\mathcal{C} = -\begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix}.$$

Notice that the adjoint of the linear operator $\mathcal{L}_m$ with respect to the Frobenius inner product is given by

$$\mathcal{L}_m^*(Z) = (\bar{\mathbb{T}}_m^A)^T Z \begin{bmatrix} I \\ 0 \end{bmatrix} + \begin{bmatrix} I & 0 \end{bmatrix} Z \bar{\mathbb{T}}_m^B. \tag{4.5}$$

The global Lanczos bidiagonalization process. This iterative procedure is initialized by

$$\beta_1 \widetilde{U}_1 = \mathcal{C}, \qquad \beta_1 = \|\mathcal{C}\|_F,$$
$$\alpha_1 \widetilde{V}_1 = \mathcal{L}_m{}^T(\widetilde{U}_1), \qquad \alpha_1 = \|\mathcal{L}_m{}^T(\widetilde{U}_1)\|_F,$$

ETNA
Kent State University
http://etna.math.kent.edu

and continued using the recurrence relations

$$\beta_{i+1}\widetilde{U}_{i+1} = \mathcal{L}_m(\widetilde{V}_i) - \alpha_i\widetilde{U}_i,$$
$$\alpha_{i+1}\widetilde{V}_{i+1} = \mathcal{L}_m^*\left(\widetilde{U}_i\right) - \beta_i\widetilde{V}_i.$$

The scalars $\alpha_i > 0$ and $\beta_i > 0$ are chosen such that $\|\widetilde{U}_i\|_F = \|\widetilde{V}_i\|_F = 1$, $i = 1, 2, \ldots$ We set $\widetilde{\mathbb{U}}_k := [\widetilde{U}_1, \widetilde{U}_2, ..., \widetilde{U}_k]$, $\widetilde{\mathbb{V}}_k := [\widetilde{V}_1, \widetilde{V}_2, ..., \widetilde{V}_k]$, and

$$\bar{T}_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}.$$

It is not difficult to show that the constructed blocks $\widetilde{V}_i$ and $\widetilde{V}_j$ are F-orthonormal (which means that they are orthonormal with respect to the Frobenius inner product). Using the global Lanczos process, we find approximate solutions Y^k of the exact solution Y_m^{MR} of problem (3.1). We can write the previous recurrence relations in matrix form as

$$[\mathcal{L}_m(\widetilde{V}_1), \mathcal{L}_m(\widetilde{V}_2), ..., \mathcal{L}_m(\widetilde{V}_k)] = \widetilde{\mathbb{U}}_{k+1}(\bar{T}_k \otimes I). \tag{4.6}$$

The method consists in searching for an approximate solution of the form

$$Y^k = \sum_{i=1}^{k} z^{(i)}\widetilde{V}_i.$$

Applying the linear operator $\mathcal{L}_m$ to Y^k, we get

$$\begin{aligned} \mathcal{L}_m(Y^k) &= \mathcal{L}_m\left(\sum_{i=1}^{k} z^{(i)}\widetilde{V}_i\right) = \sum_{i=1}^{k} z^{(i)}\mathcal{L}_m(\widetilde{V}_i) \\ &= [\mathcal{L}_m(\widetilde{V}_1), ..., \mathcal{L}_m(\widetilde{V}_k)]\,(z_k \otimes I_s) = \widetilde{\mathbb{U}}_{k+1}(\bar{T}_k \otimes I)(z_k \otimes I) \\ &= \widetilde{\mathbb{U}}_{k+1}(\bar{T}_k z_k \otimes I), \end{aligned}$$

where $z_k = (z^{(1)}, z^{(2)}, \cdots, z^{(k)})^T$. Then,

$$\begin{aligned} \left\|\mathcal{C} - \mathcal{L}_m(Y^k)\right\|_F &= \left\|\beta_1\widetilde{U}_1 - \widetilde{\mathbb{U}}_{k+1}(\bar{T}_k z_k \otimes I_s)\right\|_F \\ &= \left\|\widetilde{\mathbb{U}}_{k+1}(\beta_1 e_1 \otimes I_s) - \widetilde{\mathbb{U}}_{k+1}(\bar{T}_k z_k \otimes I_s)\right\|_F \\ &= \left\|\widetilde{\mathbb{U}}_{k+1}\left[(\beta_1 e_1 \otimes I_s) - (\bar{T}_k z_k \otimes I_s)\right]\right\|_F \\ &= \left\|\beta_1 e_1 - \bar{T}_k z_k\right\|_2. \end{aligned}$$

Therefore, problem (4.3) is equivalent to

$$\min \left\|\beta_1 e_1 - \bar{T}_k z_k\right\|_2.$$

The global LSQR algorithm for solving problem (3.1) is summarized in Algorithm 2.

ETNA
Kent State University
http://etna.math.kent.edu

ALGORITHM 2. The global LSQR algorithm (Gl-LSQR).

1. Set $Y^0 = 0$.
   Compute $\beta_1 = \|\mathcal{C}\|_F$, $\widetilde{U}_1 = \mathcal{C}/\beta_1$, $\alpha_1 = \|(\mathcal{L}_m)^T(\widetilde{U}_1)\|_F$, $\widetilde{V}_1 = (\mathcal{L}_m)^T(\widetilde{U}_1)/\alpha_1$, and set $\widetilde{W}_1 = \widetilde{V}_1$, $\overline{\Phi}_1 = \beta_1$, $\overline{\rho}_1 = \alpha_1$.
2. For $i = 1, 2, ..., k_{\max}$
   (a) $\widetilde{W}_i = \mathcal{L}_m(\widetilde{V}_i) - \alpha_i \widetilde{U}_i$, $\beta_{i+1} = \|\widetilde{W}_i\|$
   (b) $\widetilde{U}_{i+1} = \widetilde{W}_i/\beta_{i+1}$
   (c) $L_i = (\mathcal{L}_m)^T(\widetilde{U}_{i+1}) - \beta_{i+1}\widetilde{V}_i$
   (d) $\alpha_{i+1} = \|L_i\|_F$
   (e) $\widetilde{V}_{i+1} = L_i/\alpha_{i+1}$, $\rho_i = \sqrt{\overline{\rho}_1^2 + \beta_{i+1}^2}$
   (f) $c_i = \overline{\rho}_1/\rho_1$, $s_i = \beta_{i+1}/\rho_1$
   (g) $\theta_{i+1} = s_i\alpha_{i+1}$, $\overline{\rho}_{i+1} = c_i\alpha_{i+1}$
   (h) $\Phi_i = c_i\overline{\Phi}_i$, $\overline{\Phi}_{i+1} = s_i\Phi_i$
   (i) $Y^i = Y^{i-1} + (\Phi_i/\rho_i)\widetilde{W}_i$
   (j) $W_{i+1} = V_{i+1} - (\theta_{i+1}/\rho_i)\widetilde{W}_i$. If $\left|\overline{\Phi}_{i+1}\right|$ is small enough then stop.
3. End For

The preceding algorithm allows us to compute an approximate solution Y^k, with $1 \leq k \leq k_{\max}$, of the minimization problem (4.6). Next, we give an expression of the associated residual norm.

THEOREM 4.2. *Let $X_m = \mathbb{V}_m Y_m \mathbb{W}_m^T$ be the approximate solution to the Sylvester equation obtained after m iterations of the (EBA) algorithm, where*

$$Y_m = \underset{Y}{\operatorname{argmin}} \left\| \bar{\mathbb{T}}_m^A Y \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y \bar{\mathbb{T}}_m^{BT} + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F,$$

is obtained by the global LSQR algorithm. Then

$$\|R_m(X_m)\|_F = \Phi,$$

where $\Phi = |\overline{\Phi}_{k_{\max}+1}|$ is given in Algorithm 2.

Proof. We have

$$\begin{aligned} \|R(X_m)\|_F &= \|AX_m + X_m B + EF^T\|_F \\ &= \left\| \bar{\mathbb{T}}_m^A Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m \bar{\mathbb{T}}_m^{BT} + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F \\ &= \min_Y \| \mathcal{L}_m(Y) - \mathcal{C}\|_F \\ &= \Phi. \qquad \square \end{aligned}$$

Theorem 4.2 plays a very important role in practice. It allows us to compute the norm of the residual without computing the approximate solution, which is available only when convergence is achieved.

If the matrices A and B are stable ($\mathrm{Re}(\lambda_i(A)) < 0$ and $\mathrm{Re}(\lambda_i(B)) < 0$), then the Sylvester matrix equation (1.1) has a unique solution given by the integral representation (see [17])

$$X = \int_0^\infty e^{tA}\, EF^T\, e^{tB}\, dt.$$

ETNA
Kent State University
http://etna.math.kent.edu

The logarithmic "2-norm" of the stable matrix A is defined by

$$\mu_2(A) = \frac{1}{2}\,\lambda_{max}(A + A^T) < 0.$$

The logarithmic norm provides a useful bound for the matrix exponential. It is known (see [21]) that

$$\| e^{tA} \|_2 \leq e^{\mu_2(A)t}, \qquad t \geq 0.$$

In the following proposition, we give an upper bound for the norm of the error $X - X_m$.

THEOREM 4.3. *Assume that A and B are stable matrices, and let $X_m = \mathbb{V}_m Y_m \mathbb{W}_m^T$ be the approximate solution to the Sylvester equation obtained after m iterations of the extended block Arnoldi algorithm. Then we obtain the following upper bound for the error $X - X_m$:*

$$\| X - X_m \|_2 \leq \frac{-r_m}{\mu_2(A) + \mu_2(B)},$$

where r_m is the residual norm given by (3.2).

Proof. Subtracting $R(X_m) = AX_m + X_m B + EF^T$ from the initial Sylvester matrix equation (1.1), we get

$$A(X_m - X) + B(X_m - X) = R(X_m).$$

As A and B are assumed to be stable, we find

$$X - X_m = \int_0^\infty e^{tA}\, R(X_m)\, e^{tB}\, dt.$$

Hence, by using the logarithmic norm, we obtain

$$\| X - X_m \| \leq \| R(X_m) \|_2 \int_0^\infty e^{t(\mu_2(A)+\mu_2(B))}\, dt.$$

Therefore, as $\| R(X_m) \|_2 \leq \| R(X_m) \|_F$ and $\mu_2(A) + \mu_2(B) < 0$, the result follows. ∎

The convergence of the global LSQR algorithm may be slow, and then we have to use this method with a preconditioner. Different strategies are possible, and one can adapt the techniques used in [1].

4.4. The preconditioned global conjugate gradient method. In this section, we consider the preconditioned global conjugate gradient method (PGCG) for solving the least squares reduced problem (4.3). This is a matrix version of the well known PCGLS, the classical preconditioned CG method applied to the normal equation. The normal equation associated with (4.3) is given by

$$\mathcal{L}_m^*(\mathcal{L}_m Y)) = \mathcal{L}_m^*(\mathcal{C}), \tag{4.7}$$

where the operators $\mathcal{L}_m$ and $\mathcal{L}_m^*$ are defined in (4.4) and (4.5), respectively.

Let the matrices $\overline{\mathbb{T}}_m^A$ and $\overline{\mathbb{T}}_m^B$ be expressed as

$$\overline{\mathbb{T}}_m^A = \begin{bmatrix} \mathbb{T}_m^A \\ h_m^A \end{bmatrix} \quad \text{and} \quad \overline{\mathbb{T}}_B = \begin{bmatrix} \mathbb{T}_m^B \\ h_m^B \end{bmatrix},$$

ETNA
Kent State University
http://etna.math.kent.edu

where h_A and h_B represent the $2r$ last rows of the matrices $\overline{\mathbb{T}}_m^A$ and $\overline{\mathbb{T}}_m^B$, respectively. Then the normal equations (4.7) can be written as

$$\overline{\mathbb{T}}_m^{A\,T}\overline{\mathbb{T}}_m^A Y + Y\overline{\mathbb{T}}_m^{B\,T}\overline{\mathbb{T}}_m^B + \mathbb{T}_m^{A\,T} Y \mathbb{T}_m^{B\,T} + \mathbb{T}_m^A Y \mathbb{T}_m^B - \mathcal{C}_1 = 0, \tag{4.8}$$

where $\mathcal{C}_1 = \mathcal{L}_m^T(\mathcal{C})$. Considering the singular value decomposition (SVD) of the matrices $\overline{\mathbb{T}}_m^A$ and $\overline{\mathbb{T}}_m^B$,

$$\overline{\mathbb{T}}_m^A = \overline{U}_A \overline{\Sigma}_A \overline{V}_A^T, \qquad \overline{\mathbb{T}}_m^B = \overline{U}_B \overline{\Sigma}_B \overline{V}_B^T,$$

we get the following eigendecompositions

$$\overline{\mathbb{T}}_m^{A\,T}\overline{\mathbb{T}}_m^A = Q_A D_A Q_A^T, \qquad \overline{\mathbb{T}}_m^{B\,T}\overline{\mathbb{T}}_m^B = Q_B D_B Q_B^T,$$

where $Q_A = \overline{V}_A$, $Q_B = \overline{V}_B$, and $D_A = \overline{\Sigma}_A^T \overline{\Sigma}_A$. Setting $\widetilde{Y} = Q_A^T Y Q_B$ and $\widetilde{\mathcal{C}} = Q_A^T \mathcal{C}_1 Q_B$, the normal equations (4.8) are now expressed as

$$D_A \widetilde{Y} + \widetilde{Y} D_B + \widetilde{\mathbb{T}}_m^A \widetilde{Y} \widetilde{\mathbb{T}}_m^B + (\widetilde{\mathbb{T}}_m^A)^T \widetilde{Y} \widetilde{\mathbb{T}}_m^B)^T - \widetilde{\mathcal{C}} = 0, \tag{4.9}$$

where $\widetilde{\mathbb{T}}_m^A = Q_A^T \mathbb{T}_m^A Q_A$, $\widetilde{\mathbb{T}}_m^B = Q_B^T \mathbb{T}_m^B Q_B$, and $\widetilde{Y} = Q_A^T Y Q_B$. This expression suggests that one can use the first part as a preconditioner, that is, the matrix operator

$$\mathcal{P}(\widetilde{Y}) = D_A \widetilde{Y} + \widetilde{Y} D_B. \tag{4.10}$$

It can be seen that expression (4.9) corresponds to the normal equations of the matrix operator

$$\widetilde{\mathcal{L}}_m(\widetilde{Y}) = \widetilde{\mathcal{T}}_m^A \widetilde{Y} \begin{bmatrix} Q_B^T & 0 \end{bmatrix} + \begin{bmatrix} Q_A \\ 0 \end{bmatrix} \widetilde{Y} (\widetilde{\mathcal{T}}_m^B)^T, \tag{4.11}$$

where $\widetilde{\mathcal{T}}_m^A = \overline{\mathbb{T}}_m^A Q_A$ and $\widetilde{\mathcal{T}}_m^B = \overline{\mathbb{T}}_m^B Q_B$. Therefore, the preconditioned global conjugate gradient algorithm is obtained by applying the preconditioner (4.10) to the normal equations associated with the matrix linear operator defined by (4.11). This is summarized in Algorithm 3.

ALGORITHM 3. The preconditioned global CG algorithm (PGCG).

1. Choose a tolerance $\mathrm{tol} > 0$, a maximum number of $j_{\max}$ iterations.
   Choose $\widetilde{Y}_0$, and set $\widetilde{R}_0 = \mathcal{C} - \widetilde{\mathcal{L}}_m(\widetilde{Y}_0)$, $S_0 = \widetilde{\mathcal{L}}_m^*(\widetilde{R}_0)$, $Z_0 = \mathcal{P}^{-1}(S_0)$, $P_0 = S_0$.
2. For $j = 0, 1, 2, ..., j_{\max}$
   (a) $W_j = \widetilde{\mathcal{L}}_m(P_j)$
   (b) $\alpha_j = \langle S_j, Z_j \rangle_F / |W_j|_F^2$
   (c) $\widetilde{Y}_{j+1} = \widetilde{Y}_j + \alpha_j P_j$
   (d) $\widetilde{R}_{j+1} = \widetilde{R}_j - \alpha_j W_j$
   (e) If $\|\widetilde{R}_{j+1}\|_F < \mathrm{tol}$, stop
   else
   (f) $S_{j+1} = \widetilde{\mathcal{L}}_m^*(\widetilde{R}_{j+1})$
   (g) $Z_{j+1} = \mathcal{P}^{-1}(S_{j+1})$
   (h) $\beta_j = \langle S_{j+1}, Z_{j+1} \rangle_F / \langle S_j, Z_j \rangle_F$
   (i) $P_{j+1} = Z_{j+1} + \beta_j P_j$.
3. End For

Notice that the use of the preconditioner $\mathcal{P}$ requires solving a Sylvester equation at each iteration. But since the matrices D_A and D_B of these Sylvester matrix equations are diagonal matrices, the cost is reduced.

ETNA
Kent State University
http://etna.math.kent.edu

4.5. Low-rank form of the approximate solutions. The solution X_m can be given as a product of two matrices of low-rank. It is possible to decompose it as $X_m = Z_1 Z_2^T$, where the matrices Z_1 and Z_2 are of low-rank (lower than $2m$). Consider the singular value decomposition of the $2mr \times 2mr$ matrix

$$Y_m^{MR} = \widetilde{Y}_1 \Sigma \widetilde{Y}_2^T,$$

where Σ is the diagonal matrix of the singular values of Y_m^{MR} sorted in decreasing order. Let $Y_{1,l}$ and $Y_{2,l}$ be the $2mr \times l$ matrices consisting of the first l columns of $\widetilde{Y}_1$ and $\widetilde{Y}_2$, respectively, corresponding to the l singular values of magnitude larger than some tolerance. We obtain the truncated singular value decomposition

$$Y_m^{MR} \approx U_{1,l}\, \Sigma_l\, {U_{2,l}}^T,$$

where $\Sigma_l = \mathrm{diag}[\sigma_1, \ldots, \sigma_l]$. Setting $Z_{1,m} = \mathbb{V}_m\, U_{1,l}\, \Sigma_l^{1/2}$, and $Z_{2,m} = \mathbb{W}_m\, U_{2,l}\, \Sigma_l^{1/2}$, it follows that

$$X_m \approx Z_{1,m}\, Z_{2,m}^T. \tag{4.12}$$

This is very important for large problems, because one does not have to compute and store the approximation X_m at each iteration. We notice that there are cheaper ways to compute low-rank representations of the approximate solution; see [3].

The minimal residual algorithm for solving Sylvester matrix equations (MRS) is summarized in Algorithm 4.

ALGORITHM 4. The minimal residual method for large Sylvester matrix equations (MR).

1. Choose a tolerance $\mathrm{tol} > 0$, a maximum number of $\mathrm{itermax}$ iterations.
2. For $m = 1, 2, 3, ..., \mathrm{itermax}$
3. Update $\mathbb{V}_m, \overline{\mathbb{T}}_m^A$, by EBA (Algorithm 1) applied to (A, E).
4. Update $\mathbb{W}_m, \overline{\mathbb{T}}_m^B$, by EBA applied to (B^T, F).
5. Solve the low-order problem (3.1);
6. if $\|R(X_m)\|_F \leq \mathrm{tol}$, stop.
7. End For
8. Using (4.12), the approximate solution X_m is given by $X_m \approx Z_{1,m}\, Z_{2,m}^T$.

5. Numerical experiments. In this section, we present some numerical examples of continuous-time Sylvester matrix equations. We give comparisons between the Galerkin projection approach (GA) and the minimal residual (MR) approach for large-scale problems using the global LSQR (Gl-LSQR), the preconditioned global conjugate gradient method (PGCG), the direct Kronecker product when solving the reduced least squares minimization problem, and the Hu-Reichel method (HR). In the last experiment, we also give a comparison with the low-rank factored ADI (Lr_ADI) method described in [3, 4]. The algorithms are coded in Matlab 8.0. For the Galerkin approach (GA) and at each iteration m, the projected problem of size $2mr \times 2mr$ is solved by using the Bartels-Stewart algorithm [2]. When solving the reduced minimization problem, the global LSQR and the preconditioned global CG (PGCG) are stopped when the relative norm of the residual is less than $\mathrm{tol}_l = 10^{-12}$ or when a maximum of $k_{\max} = 1000$ iterations is achieved. The minimal residual (Algorithm 4) and the Galerkin approach are stopped when the norm of the residual is less than $\mathrm{tol} = 10^{-7}$, and a maximum of $\mathrm{itermax} = 50$ outer iterations is allowed.

ETNA
Kent State University
http://etna.math.kent.edu

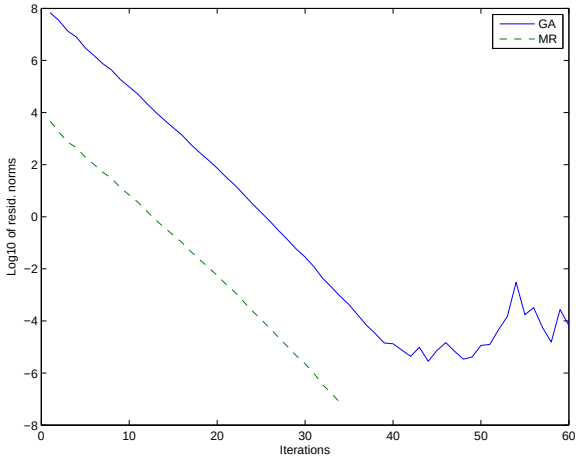

FIG. 5.1. *GA: solid line, MR: dashed line.*

The first set of matrices A and B is obtained from the discretisation of the operator

$$L_u = \Delta u - f_1(x,y)\frac{\partial u}{\partial x} + f_2(x,y)\frac{\partial u}{\partial y} + g(x,y) \tag{5.1}$$

on the unit square $[0,1] \times [0,1]$ with homogeneous Dirichlet boundary conditions. The number of inner grid points in each direction is n_0 for the operator L_u. The dimensions of the matrices A and B are $n = n_0^2$ and $s = s_0^2$, respectively. The discretization of the operator L_u yields matrices extracted from the `Lyapack` package [20] using the command `fdm_2d_matrix` and denoted as $A = fdm(f_1(x,y), f_2(x,y), g(x,y))$. For the second set of matrix tests, we use the matrices $add32$, $pde2961$, and $thermal$ from the Harwell Boeing Collection[1] and also the matrix 'flow_meter' from the Oberwolfach Collection[2]. The coefficients of the matrices E and F are random values uniformly distributed on $[0,1]$, and we set $r = 2$.

EXAMPLE 5.1. In Figure 5.1 and Figure 5.2 we plot the residual norms versus the number of iterations for the minimal residual and the Galerkin approaches. For this first experiment, we use the global LSQR algorithm to solve the reduced minimization problem (3.1). In Figure 5.1, the matrices A and B are obtained from the discretisation of the operator L_u with dimensions $n = 4900$ and $s = 3600$, respectively. In Figure 5.2, we use the matrices $A = pde2961$ and $B = thermal$ from the Harwell Boeing collection with dimensions $n = 2961$ and $s = 3456$, respectively. For this experiment, the reduced-order problem is solved by the direct least squares method associated to the Kronecker form (4.1). As can be seen from these two figures, the MR algorithm converges successfully, while for the GA approach, we obtain residual norms around 10^{-5}.

EXAMPLE 5.2. For the second set of experiments, we compare the performances of the MR method associated to the different techniques for solving the reduced-order minimization problem and the Galerkin approach (GA). The reduced minimization problem is solved by the direct least squares method, by the global LSQR, by the preconditioned conjugate gradient

[1] `http://math.nist.gov/MatrixMarket`
[2] `https://portal.uni-freiburg.de/imteksimulation/benchmark`

ETNA
Kent State University
http://etna.math.kent.edu

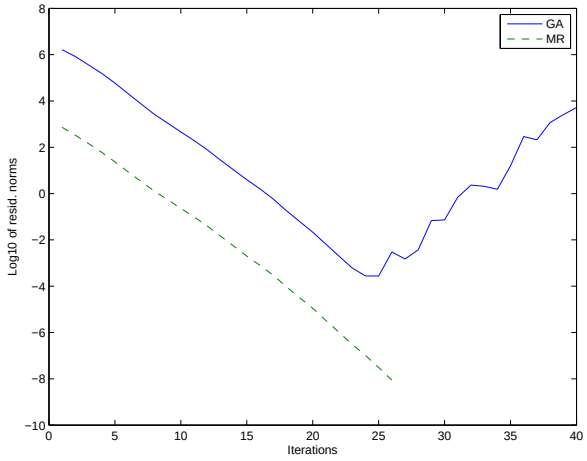

FIG. 5.2. *GA: solid line, MR: dashed line.*

method for the normal equation (GPCG), and by the Hu-Reichel method (HR). In Table 5.1, we list the CPU times (in seconds), the total number of outer iterations, and the norms of the residuals obtained by these different approaches. The reported times include the time required for LU computations of the matrices A and B used in the extended block Arnoldi process. As can be seen from this table, the results given by the Galerkin approach are not good (except for the first experiment). In some cases, the approximations produced by the GA method deteriorate after a number of iterations. We also notice that when using the direct method for solving the reduced minimization problem, the CPU time is higher when many outer iterations are needed to achieve convergence; this is the case for the last two experiments.

EXAMPLE 5.3. For the last experiment, we compare the performances of the GA approach, the MR method with GPCG as a solver for the low-order minimization problem, and the low-rank factored ADI (Lr_ADI) method described in [3, 4]. For this experiment, the matrices A and B are the same as those given in [4, Example 1]. They are obtained from the 5-point discretization of the operator L_u in (5.1) with dimensions $n = 6400$ for A and $s = 3600$ for B. For this experiment, E and F are random matrices with $r = 4$ columns. In Table 5.2, we list the residual norms and the corresponding CPU time for each method. For this experiment, the algorithms are stopped when the relative residual norms are smaller than 10^{-11}.

6. Conclusion. In this paper we presented an iterative method for solving large-scale Sylvester matrix equations with low-rank right-hand sides. The proposed method is based on a projection onto extended block Krylov subspaces with a minimization property. The obtained low-order minimization problem is solved via iterative or direct methods. To speed up the convergence when solving the low-order minimization problem, we use a preconditioned global conjugate gradient method associated to the normal matrix equation. The approximate solutions are given as a product of two low-rank matrices, which allows to save memory for large problems. The advantage of the minimal residual method as compared to the Galerkin-type counterpart is its stability. This is demonstrated by the performed numerical tests.

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 5.1
Results for experiments 2.

Matrices	Method	CPU time	Its.	Res. norm
$A = thermal$,	MR(PGCG)	$26s$	10	1.1×10^{-8}
$B = add32$	MR (direct)	$26s$	9	2.1×10^{-8}
$n = 3456$, $s = 4960$	MR (Gl-LSQR)	$28s$	10	1.3×10^{-8}
	MR(HR)	$27s$	9	3.5×10^{-8}
	GA	$25s$	9	1.4×10^{-8}
$A = fdm(cos(xy), e^{y^2 x}, 100)$	MR(PGCG)	$45s$	15	1.9×10^{-8}
$B = add32$	MR (direct)	$56s$	15	2.3×10^{-8}
$n = 90000$, $s = 4960$	MR (Gl-LSQR)	$49s$	17	1.1×10^{-8}
	MR(HR)	$88s$	15	3.1×10^{-8}
	GA	$83s$	21	8.6×10^{-5}
$A = fdm(sin(xy), e^{xy}, 10)$	MR(PGCG)	$56s$	18	2.5×10^{-8}
$B = thermal$	MR(direct)	$58s$	16	3.1×10^{-8}
$n = 122500$, $s = 3456$	MR(Gl-LSQR)	$--$	50	$----$
	MR(HR)	$110s$	16	2.0×10^{-8}
	GA	$82s$	25	8.4×10^{-5}
$A = flow$	MR (PGCG)	$49s$	16	2.0×10^{-8}
$B = fdm(sin(xy), xy, 1000)$	MR (direct)	$700s$	35	3.1×10^{-8}
$n = 9669$, $s = 40000$	MR(Gl-LSQR)	$--$	50	$----$
	MR(HR)	$150s$	35	2.4×10^{-8}
	GA	$95s$	50	2.5×10^{-5}
$A = fdm(xy, y^2, 1)$	MR(PGCG)	$706s$	18	2.1×10^{-8}
$B = fdm(xy, cos(xy), 10)$	MR(direct)	$1500s$	42	1.5×10^{-8}
$n = 122500$, $s = 48400$	MR(Gl-LSQR)	$--$	$--$	$----$
	MR(HR)	$--$	$--$	$----$
	GA	$805s$	50	4.2×10^{-4}

TABLE 5.2
Results for experiments 3.

Method	Res. norm.	CPU time
MR(PGCG)	1.2×10^{-12}	$2.8s$
GA	6.4×10^{-12}	$3.2s$
Lr_ADI	2.5×10^{-12}	$5.2s$

Acknowledgments. We would like to thank the two referees for their valuable remarks and helpful comments and suggestions. We also thank Dr. Patrick Kürschner for providing us with the Matlab program for the Lr_ADI method.

REFERENCES

[1] J. BAGLAMA, L. REICHEL, AND D. RICHMOND, *An augmented LSQR method*, Numer. Algorithms, 64 (2013), pp. 263–293.
[2] R. H. BARTELS AND G. W. STEWART, *Algorithm* 432*: Solution of the matrix equation* $AX + XB = C$, Comm. ACM, 15 (1972), pp. 820–826.

ETNA
Kent State University
http://etna.math.kent.edu

[3] P. BENNER, R. C. LI, AND N. TRUHAR, *On the ADI method for Sylvester equations*, J. Comput. Appl. Math., 233 (2009), pp. 1035–1045.
[4] P. BENNER AND P. KÜRSCHNER, *Computing real low-rank solutions of Sylvester equations by the factored ADI method*, Comput. Math. Appl., 67 (2014), pp. 1656–1672.
[5] V. DRUSKIN AND L. KNIZHNERMAN, *Extended Krylov subspaces: approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 755–771.
[6] A. EL GUENNOUNI, K. JBILOU, AND A. J. RIQUET, *Block Krylov subspace methods for solving large Sylvester equations*, Numer. Algorithms, 29 (2002), pp. 75–96.
[7] G. H. GOLUB AND W. KAHAN, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224.
[8] G. H. GOLUB, S. NASH, AND C. VAN LOAN, *A Hessenberg Schur method for the problem* $AX + XB = C$, IEEE Trans. Automat. Control, 24 (1979), pp. 909–913.
[9] M. HEYOUNI AND K. JBILOU, *An extended Block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation*, Electron. Trans. Numer. Anal., 33 (2009), pp. 53–62. http://etna.mcs.kent.edu/vol.33.2008-2009/pp53-62.dir
[10] M. HEYOUNI, *Extended Arnoldi methods for large low-rank Sylvester matrix equations*, Appl. Numer. Math., 60 (2010), pp. 1171–1182.
[11] D. HU AND L. REICHEL, *Krylov-subspace methods for the Sylvester equation*, Linear Algebra Appl., 172 (1992), pp. 283–313.
[12] I. M. JAIMOUKHA AND E. M. KASENALLY, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., 31 (1994), pp. 227–251.
[13] K. JBILOU, *Low-rank approximate solution to large Sylvester matrix equations*, App. Math. Comput., 177 (2006), pp. 365–376.
[14] K. JBILOU AND A. J. RIQUET, *Projection methods for large Lyapunov matrix equations*, Linear Algebra Appl., 415 (2006), pp. 344–358.
[15] Z. JIA AND Y. SUN, *A QR decomposition based solver for the least squares problems from the minimal residual method for the Sylvester equation*, J. Comput. Math., 25 (2007), pp. 531–542.
[16] D. KRESSNER AND C. TOBLER, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1288–1316.
[17] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, Academic Press, Orlando, 1985.
[18] Y. LIN AND V. SIMONCINI, *Minimal residual methods for large scale Lyapunov equations*, App. Numer. Math., 72 (2013), pp. 52–71.
[19] C. C. PAIGE AND A. SAUNDERS, *LSQR: an algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[20] T. PENZL, *LYAPACK: A MATLAB toolbox for large Lyapunov and Riccati equations, model reduction problems, and linear-quadratic optimal control problems*, software available at https://www.tu-chemnitz.de/sfb393/lyapack/.
[21] C. MOLER AND C. VAN LOAN, *Ninteen dubious ways to compute the exponential of a matrix*, SIAM Rev., 20 (1978), pp. 801–836.
[22] Y. SAAD, *Numerical solution of large Lyapunov equations*, in Signal Processing, Scattering, Operator Theory and Numerical Methods, M. A. Kaashoek, J. H. Van Shuppen, and A. C. Ran, eds., Progress in Systems and Control Theory 5, Birkhäuser, Boston, 1990, pp. 503–511.
[23] V. SIMONCINI, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.