The BGMRES Method for Generalized Sylvester Matrix Equation $AXB - X = C$ and Preconditioning

Azita Tajaddini, Ramleh Shamsi

Abstract—In this paper, we present the block generalized minimal residual (BGMRES) method in order to solve the generalized Sylvester matrix equation. However, this method may not be converged in some problems. We construct a polynomial preconditioner based on BGMRES which shows why polynomial preconditioner is superior to some block solvers. Finally, numerical experiments report the effectiveness of this method.

Keywords—Linear matrix equation, Block GMRES, matrix Krylov subspace, polynomial preconditioner.

I. INTRODUCTION

Consider the generalized linear sylvester equation

$$AXB - X = C,$$  

(1)

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times s}$, $C \in \mathbb{R}^{n \times s}$, and $X$ is the unknown matrix in $\mathbb{R}^{n \times s}$. The matrix equation (1) plays an important role in control and communication theory; see [3], [4]. Also, the discrete-time time-invariant linear systems

$$x_{k+1} = Ax_k + Bu_k,$$

$$y_k = cx_k,$$

$k = 0, 1, 2, ...$ and $x_0$ is given. These systems are associated, for instance, with the discrete-time Lyapunov equation

$$AXA^T - X = -BB^T.$$  

It arises naturally in a wide variety of control applications such as stability analysis [17] model order reduction [12], [15], [18] and Newton’s method for discrete algebraic Riccati equations [2]. The discrete-time Lyapunov equation is a special case of matrix equation (1).

The analytical solution of the matrix equation (1) has been considered by many authors; see [6]. They have proposed the Bartels-Stewart algorithm for solving matrix equation (1). The direct methods for solving the matrix equation (1) such as stability analysis [17] model order reduction [12], [15], [18] and Newton’s method for discrete algebraic Riccati equations [2]. The discrete-time Lyapunov equation is a special case of matrix equation (1).

The analytical solution of the matrix equation (1) has been considered by many authors; see [6]. They have proposed the Bartels-Stewart algorithm for solving matrix equation (1). The direct methods for solving the matrix equation (1) such as stability analysis [17] model order reduction [12], [15], [18] and Newton’s method for discrete algebraic Riccati equations [2]. The discrete-time Lyapunov equation is a special case of matrix equation (1).

The direct methods for solving the matrix equation (1) such as stability analysis [17] model order reduction [12], [15], [18] and Newton’s method for discrete algebraic Riccati equations [2]. The discrete-time Lyapunov equation is a special case of matrix equation (1).

Throughout this paper, we use the following notations:

Let $X, Y \in \mathbb{R}^{n \times p}$, the Frobenius inner product is defined $<X, Y>_{F} = tr(X^TY)$, where $tr(\cdot)$ denotes the trace and $X^T$ the transpose of the matrix $X$. The associated norm is the well-known Frobenius norm denoted by $\|\cdot\|_F$. A system of matrices of $\mathbb{R}^{n \times p}$ is said to be F-orthogonal, if it is orthogonal with respect to the scalar product $<\cdot, \cdot>_F$, that means $tr(X^TY) = 0$. The Frobenius product of $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ is defined by $A \otimes B := [a_{ij}] \in \mathbb{R}^{mp \times nq}$. We Krylov subspace and simultaneously reduce the order of the generalized Sylvester equation (1). By the same token, in [3], Bouhamidi et al. proposed the global generalized minimum residual (GL-GMRES) to solve the following general linear matrix equations:

$$\sum_{j=1}^{p} A_jX B_j = C,$$

and then they applied ILU (incomplete LU factorization) and SSOR preconditioning in order to solve the Sylvester equation $AX+XB=C$. Here, we use ILU and SSOR preconditioning the same as the methods that [3] were used.

In this paper, we generalize the GMRES method to obtain a matrix iterative method to solve the matrix equation (1). Also, we present a polynomial preconditioner. Then, we compare these methods with BGMRES (ILU), BGMRES (SSOR) and NSCG [19], squared Smith (SM) and restarted Krylov squared Smith (RKSS) [10], [16] methods. Finally, we show that PBGMRES is more effective than the other methods.

Let $X = [x_1, x_2, ..., x_s]$, where $x_i$, $i = 1, 2, ..., s$ is ith column of $X$. We define a linear operator

$$\text{vec} : \mathbb{R}^{n \times s} \rightarrow \mathbb{R}^{ns}$$

$$X \mapsto [x_1^T, x_2^T, ..., x_s^T]^T$$

(2)

Hence, the linear matrix equation (1) can be written as the following $ns \times ns$ linear system

$$Ax = c,$$  

(3)

where $A = (B^T \otimes A - I_{ns}), x = \text{vec}(X), c = \text{vec}(C)$ and $\otimes$ denotes the Kronecker product, see [4]. This product satisfies the properties

$$(A \otimes B)(C \otimes D) = (AC \otimes BD), \quad (A \otimes B)^T = A^T \otimes B^T.$$  

Equation (3) has a unique solution if and only if the matrix $A$ is nonsingular.

A. Tajaddini is with the Department of Applied Mathematics, Faculty of Mathematics & Computer Sciences, Shahid Bahonar University of Kerman, Kerman, Iran (e-mail: atajadini@uk.ac.ir).

R. Shamsi is with Anonymous University.
use the notation * for the following product [8], [9], let \( V_m \) denote the \( n \times m \) block matrix \( V_m = [V_1, \ldots, V_m] \), where \( V_i \in \mathbb{R}^{n \times s} \) for \( i = 1, \ldots, m \). If \( D_m = [d_1, \ldots, d_m] \in \mathbb{R}^{n \times m} \) and \( \alpha = [\alpha_1, \ldots, \alpha_m]^T \in \mathbb{R}^m \), then we have

\[
V_m * \alpha = \sum_{i=1}^{m} \alpha_i V_i,
\]

\[
V_m * D_m = [V_m * d_1, \ldots, V_m * d_m].
\]

If \( H_m \in \mathbb{R}^{m \times m} \) and \( \alpha, \beta \in \mathbb{R}^m \), then the matrix product * satisfies the following properties

\[
V_m * (\alpha + \beta) = (V_m * \alpha) + (V_m * \beta),
\]

\[
(V_m * H_m) * \alpha = V_m * (H_m \alpha),
\]

\[
(V_m * \alpha)^T = V_m^T * \alpha.
\]

The remainder of this paper is organized as follows: In Section II, a description of the block generalized minimal residual (BGMRES) is given. In Section III, we show how to apply polynomial preconditioner in order to solve the generalized Sylveter matrix equation \( AXB - X = C \). Section IV is devoted to some numerical experiments. Finally, conclusion is given in Section V.

II. THE BLOCK GMRES METHOD

In this section, we define the \( m \)th generalized matrix Krylov subspace and recall the modified global Arnoldi process; for more details, see [11], [13].

Definition 1. Let \( V \) be any \( n \times s \) matrix. Then, the generalized matrix Krylov subspace is associated to \((A, V, B)\) and an integer \( m \) is defined as

\[
\mathcal{K}_m(A, V, B) = \text{span}\{V, AVB, \ldots, A^{m-1}V B^{m-1}\}.
\]

The modified global Arnoldi process allows us to construct an F-orthonormal basis for the generalized matrix Krylov \( \mathcal{K}_m(A, V, B) \), see [11], [13].

Algorithm 1 The Modified Global Arnoldi Algorithm

Require: \( A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{s \times s}, m \in \mathbb{N} \) and the nonzeros matrix \( V \in \mathbb{R}^{n \times s} \).

Ensure: The block vectors \( V_1, V_2, \ldots, V_{m+1} \) and the semi upper hessenberg matrix \( H_m = (h_{ij}) \).

1: set \( V_1 = V/\|V\|_F \);  
2: for \( j = 1, \ldots, m \) do  
3: \( W = AV_j \);  
4: \( W = WB \);  
5: for \( i = 1, \ldots, j \) do  
6: \( h_{ij} = <V_i, W>_F = tr(W^T V_i) \);  
7: \( W = W - h_{ij} V_i \);  
8: end for  
9: Compute \( h_{j+1,j} = \|W\|_F \); if \( h_{j+1,j} = 0 \), stop  
10: Compute \( V_{j+1} = W/h_{j+1,j} \);  
11: end for

Since the modified Arnoldi algorithm involves the Gram-Schmidt process, algorithm 1 builds an F-orthonormal basis \( V_m = [V_1, V_2, \ldots, V_m] \), \( V_i \in \mathbb{R}^{n \times s} \) for the generalized Krylov subspace \( \mathcal{G}_m(A, V, B) \) and a semi upper Hessenberg matrix \( H_m \in \mathbb{R}^{m+1 \times m} \). The following theorem can be easily proved.

Theorem 1. Let \( V_m, H_m \) and \( H_m \) be as defined above. The global Arnoldi process satisfies the following 
\begin{enumerate}
    \item \( AV_m (I_m \otimes B) = V_m * H_m + E_{m+1} \), where \( E_{m+1} = h_{m+1,m}[O_{n \times s}, \ldots, O_{n \times s}, V_{m+1}] \).
    \item \( AV_m (I_m \otimes B) = V_m * H_m \).
    \item For any \((m+1) \times s\) matrix \( G \), we have
    \[ \|V_{m+1} * G\|_F = \|G\|_2 \].
\end{enumerate}

Theorem 2. Let \( A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{s \times s}, C \in \mathbb{R}^{s \times s} \). Let \( X_0 \in \mathbb{R}^{n \times s} \) be an initial guess and \( R_0 \) is its corresponding residual. Then

\[
K_m(B^T \otimes A, r_0) = K_m(A, r_0),
\]

where \( A = (B^T \otimes A - I_n) \), \( r_0 = vec(R_0) \) and \( K_m(A, r_0) \) are the classic Krylov subspaces.

Remark 1. Suppose that
\[ K_m(B^T \otimes A, r_0) = \text{span}\{r_0, (B^T \otimes A)r_0, \ldots, (B^T \otimes A)^{m-1}r_0 \}, \]

where \( r_0 = vec(R_0) \). The map
\[ T: \mathcal{G}_m(A, R_0, B) \rightarrow K_m(B^T \otimes A, r_0) \]

given by \( Z \mapsto vec(Z) \) is an isomorphism. Hence, by theorem 2 and above discussion, the two subspace \( \mathcal{G}_m(A, R_0, B) \) and \( K_m(B^T \otimes A, r_0) \) are isomorphic i.e.

\[
\mathcal{G}_m(A, R_0, B) \simeq K_m(A, r_0).
\]

Therefore, we can conclude that

\[
AGK_m(A, R_0, B) \simeq AK_m(A, r_0),
\]

where

\[
AGK_m(A, R_0, B) = \text{span}\{AR_0B, A^2R_0B^2, \ldots, A^mR_0B^m\},
\]

and

\[
AK_m(A, r_0) = \text{span}\{Ar_0, A^2r_0, \ldots, A^mr_0\}.
\]

Let \( X_0 \in \mathbb{R}^{n \times s} \) be an initial guess and the corresponding residual is \( R_0 = C - AX_0B + X_0 \). The block GMRES algorithm, at the \( m \)th step, constructs the approximation solution \( X_m \) to the solution of (1) such that

\[
X_m = X_0 + Z_m \quad \text{s.t.} \quad Z_m \in \mathcal{G}_m(A, R_0, B).
\]

with F-orthogonality relation

\[
R_m = C - AX_mB + X_m \perp_{F} AGK_m(A, R_0, B). \quad (6)
\]

In theorem 3, we show that F-orthogonality (6) is equivalent to minimization problem (7).

Theorem 3. Let \( A \) and \( B \) be two arbitrary matrices. Let \( X_0 \) be an initial guess and \( R_0 \) is its corresponding residual. Then a matrix \( X_m \) is the result of an oblique projection method onto \( \mathcal{G}_m(A, R_0, B) \) and F-orthogonal to \( AGK_m(A, R_0, B) \) if and only if it minimizes the F-norm of the residual matrix.
Theorem 4. At step $m$, the approximation solution $X_m$ produced by the block GMRES method is given by $X_m = X_0 + V_m y_m$, where $y_m$ is the solution of the following small least square problem:

$$y_m = \arg \min_{y \in \mathbb{R}^m} || R_0 ||_F e_1 - \left( H_m - I_m \begin{bmatrix} \bar{H} & m & e_m^T \end{bmatrix} \right) y_2,$$

(8)

where $e_1$ is the first unit vector of $\mathbb{R}^{m+1}$.

The minimization (8) can be solved by the QR factorization of $\left( H_m - I_m \begin{bmatrix} \bar{H} & m & e_m^T \end{bmatrix} \right)$ with Givens rotations. By algorithm 1, the $\left( H_m - I_m \begin{bmatrix} \bar{H} & m & e_m^T \end{bmatrix} \right)$ is an unreduced semi upper Hessenberg matrix and $\text{rank}(\left( H_m - I_m \begin{bmatrix} \bar{H} & m & e_m^T \end{bmatrix} \right)) = m$. Then, there is an orthogonal matrix $Q_m \in \mathbb{R}^{(m+1) \times (m+1)}$ and an invertible upper triangular matrix $\bar{R}_m \in \mathbb{R}^{(m)\times (m)}$ such that

$$\left( H_m - I_m \begin{bmatrix} \bar{H} & m & e_m^T \end{bmatrix} \right) = Q_m \begin{bmatrix} \bar{R}_m & 0 \end{bmatrix},$$

(9)

With substitution (9) into (8), we obtain

$$|| R_m ||_F = \min_{y \in \mathbb{R}^m} || R_0 ||_F (Q_m^T e_1) = \begin{bmatrix} \bar{R}_m & 0 \end{bmatrix} y_2$$

for obtaining $y_m$, we solve the following upper triangular system:

$$\bar{R}_m y = || R_0 ||_F \begin{bmatrix} I_m & 0 \end{bmatrix} (Q_m^T e_1).$$

Then, we get the BGMRES iterative solution to (1),

$$X_m = X_0 + V_m y_m.$$

It can be easily shown that the residual matrix form is as:

Theorem 5. The residual matrix at step $m$, $R_m = C - A X_m B + X_m$ produced by the block GMRES for the linear matrix equation satisfies the following properties

$$R_m = \gamma_{m+1} V_{m+1}^* (Q_m e_{m+1}),$$

and

$$|| R_m ||_F = | \gamma_{m+1} |,$$

where $\gamma_{m+1}$ is the last component of the vector $g_m = || R_0 ||_F Q_m e_1$ and $e_{m+1} = (0, \ldots, 0, 1)^T \in \mathbb{R}^{m+1}$.

Finally, the previous results can summarized in the following algorithm.

Algorithm 2 The block GMRES algorithm (BGMRES)

Require: $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{s \times s}$, an initial guess matrix $X_0 \in \mathbb{R}^{n \times s}$, $m \in \mathbb{N}$ and $\epsilon$.

Ensure: The solution $X_m$.

1: Compute $R_0 = C - A X_0 B + X_0$, $\beta = || R_0 ||_F$, $V_1 = \begin{bmatrix} R_0 \end{bmatrix} / \beta$.
2: Construct the F-orthonormal basis $V_1, V_2, ..., V_{m+1}$ and the semi upper Hessenberg matrix $H_m$ by Algorithm 1;
3: Solve the least squares problem

$$y_m = \arg \min_{y \in \mathbb{R}^m} || R_0 ||_F (Q_m^T e_1) = \begin{bmatrix} \bar{R}_m & 0 \end{bmatrix} y_2$$

4: Compute: $X_m = X_0 + V_m y_m$;
5: Compute the residual $R_m = C - A X_m B + X_m$ and $|| R_m ||_F$ by using Theorem 1;
6: if $|| R_m ||_F < \epsilon$ then
7: Stop;
9: end if
10: set $X_0 = X_m$. Go to 1;
Assume that Algorithm 1 does not stop before the mth step. Then by relation 1 from the theorem 1, we get
\[ V_{m+1} = h^{-1}_{m+1}(AV_mB - V_mH_m), \]  
(11) where
\[ H_m = (h_{1m}, \ldots, h_{mm})^T. \]
Since the space generated by the matrix \( \mathbb{K}_m \) is the same as the space spanned by the matrix \( V_m \). Therefore, we have
\[ V_m = \mathbb{K}_m * S_m, \]  
(12) where \( S_m \) is an upper triangular matrix
\[ S_m = \begin{bmatrix} s_{11} & \cdots & s_{1m} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_{mm} \end{bmatrix}. \]
By Kronecker and * product properties, we have
\[ V_m * H_m = (\mathbb{K}_m * S_m) * H_m = \mathbb{K}_m * (S_mH_m) \]
\[ = \begin{bmatrix} \mathbb{K}_m, A^mR_0B^m \end{bmatrix} * (S_mH_m) = \mathbb{K}_{m+1} * (S_mH_m). \]  
(13) Since \( AV_m(I_m \otimes B) = (A\mathbb{K}_m(I_m \otimes B)) * S_m \), then
\[ AV_mB = (A\mathbb{K}_m(I_m \otimes B)) * S_m \]
\[ = (AR_0B, A^2R_0B^2, \ldots, A^mR_0B^m) * S_m \]
\[ = \mathbb{K}_{m+1} * (S_mH_m). \]  
(14)
By substitution (13) and (14) into (11), we obtain
\[ V_{m+1} = \mathbb{K}_{m+1} * (h^{-1}_{m+1} \begin{bmatrix} 0 \\ S_mH_m \end{bmatrix}) \]  
(15) Since \( V_{m+1} = \mathbb{K}_{m+1} * S_{m+1} \), then
\[ V_{m+1} = \mathbb{K}_{m+1} * S_{m+1}. \]  
(16) From relations (15), (16) and linearly independent columns of the \( \mathbb{K}_{m+1} \), we conclude
\[ S_{m+1} = h^{-1}_{m+1} \begin{bmatrix} 0 \\ S_mH_m \end{bmatrix} = h^{-1}_{m+1}S_mH_m. \]  
(17) In the sequel, consider the approximation solution \( X_m \) that by using BGMRES method is obtained
\[ X_m = X_0 + V_m * y_m. \]
Since \( V_m * y_m = K_m * (S_m y_m) \) and
\[ R_m = R_0 - AX_mB + X_m \]
\[ = R_0 - (A(K_m(I_m \otimes B)) * S_m y_m + \mathbb{K}_m * S_m y_m = R_0 - (A\mathbb{K}_m(I_m \otimes B) - \mathbb{K}_m) * S_m y_m. \]  
(18) Then by applying \( \text{vec}(.) \) operator on equation (18), we get
\[ \text{vec}(R_m) = \]
\[ (I_n - (B^T \otimes A - I_{ns}) \sum_{j=0}^{m-1} a_j((B^T)^j \otimes A^j)) \text{vec}(R_0) \]
\[ = P(B^T; A) \text{vec}(R_0) \]  
(20) where
\[ (a_0, \ldots, a_{m-1})^T = S_my_m, \]  
(21) and the residual polynomial is as:
\[ P_m(x, y) = 1 - (xy - 1)Q_{m-1}(x, y), \]  
(22) where
\[ Q_{m-1}(x, y) = \sum_{j=0}^{m-1} a_j x^j y^j. \]  
(23) From (22), we have \[ Q_{m-1}(x, y) = \frac{1 - P_m(x, y)}{(xy - 1)} \] and \( (B^T \otimes A - I_{ns})^{-1} \approx Q_{m-1}(B^T; A) \). Therefore, we can apply \( Q_{m-1}(x, y) \) as the preconditioner polynomial. We must solve the linear equation system:
\[ Q_{m-1}(B^T; A)(B^T \otimes A - I_{ns}) \text{vec}(X) = Q_{m-1}(B^T; A) \text{vec}(C), \]  
(24) by using BGMRES. The algorithm BGMRES, at the kth step, constructs the approximate solution \( X_k \) to the solution (24) such that
\[ X_k = X_0 + Z_k \text{ s.t. } Z_k \in \mathbb{K}_k(A, R_0, B) \]  
(25) and with F-orthogonality relation
\[ R_k = \sum_{j=0}^{m-1} a_j A^jCB^j - \sum_{j=0}^{m-1} a_j (A^{j+1}X_kB^{j+1} - A^jX_kB^j) \]
\[ \perp F \mathbb{K}_k(A, R_0, B)B. \]
The above F-orthogonal relation is equivalent to the following relation:
\[ \| R_k \|_F = \min_{Z \in \mathbb{K}_k(A,R_0,B)} \| R \|_F, \]  
(26) where
\[ R = \sum_{j=0}^{m-1} a_j (A^jCB^j - (A^{j+1}(X_0 + Z)B^{j+1} - A^j(X_0 + Z)B^j)). \]  
(27) Consider F-orthonormal basis \( \mathbb{V}_k \), which is constructed by using algorithm 1. After kth step of algorithm 1 to the matrices \( A \) and \( B \) with the nonzero residual matrix \( R_0 \), we can rewrite...
the relation (26) as follows:
\[
\| \sum_{j=0}^{m-1} a_j A^j C B^j - A^{j+1}(X_0 + Z)B^{j+1} - A^j(X_0 + Z)B^j \|_F
\]
\[
= \| \sum_{j=0}^{m-1} a_j A^j C B^j - A^{j+1}(X_0 + V_k * y_k)B^{j+1} - A^j(V_k * y_k)B^j \|_F
\]
\[
= \| R_0 - \sum_{j=0}^{m-1} a_j (A^{j+1}(V_k * y_k)B^{j+1} - A^j(V_k * y_k)B^j) \|_F
\]
\[
= \| R_0 - \sum_{j=0}^{m-1} a_j A^j V_k (I_k \otimes B^{j+1}) - A^j V_k (I_k \otimes B^j) \|_F * y_k.
\]

By * product properties and theorem 1, we get
\[
\| R_0 - (a_0 V_k + (a_0 - a_1) A V_k(I_k \otimes B) + (a_1 - a_2) A^2 V_k(I_k \otimes B^2) + \ldots + (a_{m-2} - a_{m-1}) A^{m-1} V_k(I_k \otimes B^{m-1}) + a_{m-1} A^m V_k(I_k \otimes B^m)) \|_F
\]
\[
= \| R_0 - (a_0 V_k + (a_0 - a_1) V_k(I_k \otimes I_s) + (a_1 - a_2) V_{k+2}(I_k \otimes I_s) + \ldots + (a_{m-2} - a_{m-1}) V_{k+m-1}(I_k \otimes I_s) + a_{m-1} V_{k+m}(I_k \otimes I_s)) \|_F * y_k.
\]

Therefore, we obtain
\[
= \| V_{m+k}(R_0) \|_F (e_1 \otimes I_s) - (a_0(I_k \otimes I_s) + (a_0 - a_1)(I_k \otimes I_s) + (a_1 - a_2)(I_k \otimes I_s) + \ldots + (a_{m-2} - a_{m-1})(I_k \otimes I_s) + a_{m-1}(I_k \otimes I_s)) * y_k.
\]

The minimization problem (28) can be solved by the QR factorization the matrix
\[
= -a_0(I_k \otimes I_s) + \sum_{j=0}^{m-2} (a_j - a_{j+1})(I_k \otimes I_s) + a_{m-1} I_k \otimes I_s.
\]

This matrix is transformed to an upper triangular matrix and then by solving the upper triangular system, we can obtain the vector y_k.

Using the above results, the polynomial preconditioner, i.e. PBGMRES algorithm based on the BGMRES algorithm is summarized in algorithm 3:

**IV. NUMERICAL EXAMPLES**

In this section, we present some numerical examples to illustrate the potential of the new algorithm with polynomial preconditioner for the solution of the generalized linear Sylvester equation (1). In the following examples, we mainly evaluate and compare the performance of the new method against block GMRES with ILU and SSOR preconditioner [3], NSCG [19], squared Smith and restarted Krylov squared Smith [10], [16]. We use Matlab 2014a on a PC- Pentium(R), CPU 2.66GHz, 4.00 GB of RAM. We use the zero initial vector and stopping criterion \( \| R_0 \|_F < 1e^{-9} \) for all the methods.

In examples 4, 5, \( C = \begin{pmatrix} 1 & \cdots & 1 \\ 1 & \cdots & 1 \end{pmatrix} \) for all the solvers. In Tables I and II, we note that Iter, Res.norm and cputime denote iteration number, residual norm and cputime, respectively.

**Example 1.** In this example, we have tested the BGMRES, BGMRm with polynomial preconditioning, squared Smith (SM) and restarted Krylov squared Smith (RKSS) methods with \( m = 5, k = 10 \) on selected numerical example from [10], [16]. The convergence behavior of these methods are shown in Table 1.

From Table I, we can see that the PBGMRES is faster than the other methods.

**Example 2.** ([16]). We consider the continuous-time Lyapunov equation \( TX + XT^T = -EE^T \), where \( T \) is the matrix \( TU'B1000 \) of order \( n = 1000 \) representing the Jacobian of a tabular reductor model, and \( E \) is a one column vector such...
**Algorithm 3 The PBGMRES algorithm**

Require: $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times s}$, $X_0 \in \mathbb{R}^{n \times s}$, $k$, $\varepsilon$ and degree $m$ of $Q_{m-1}(x,y)$.

Ensure: $X_k$

1: Compute polynomial precondioner $Q_{m-1}(x,y)$.
2: for $l = 1, \ldots$, until convergence do
3: Compute $R_0 = C - AX_0B + X_0, \beta = \|R_0\|_F$, $V_1 = R_0/\beta, S_1 = 1/\beta$.
4: for $j = 1, \ldots, m$ do
5: $W = AV_j$;
6: $W = WB$;
7: for $i = 1, \ldots, j$ do
8: $h_{ij} = <W, Vi>_F$;
9: $W = W - h_{ij}Vi$;
10: end for
11: $h_{j+1,j} = \|W\|_F$; if $h_{j+1,j} = 0$ stop;
12: $V_{j+1} = V_{j+1}/h_{j+1,j}$;
13: set $H_{j,j} = [s_{11} \ldots s_{1j}]; S_j = [\ldots s_{j,j}]$;
14: Compute $H_{j,j+1} = [s_{1,j+1} \ldots s_{j,j+1}]$
15: $h_{j+1,j}^{-1} = [0, \ldots, 0] - h_{j+1,j}^{-1} [S_j H_{j,j}]$;
16: end for
17: Solve the least squares problem
18: $y_m = \arg \min_{y \in \mathbb{R}^m} \|R_0\|_F ||y||_2$.
19: Compute $X_m = X_0 + V_m * y_m$;
20: Compute $X_m = C - AX_mB + X_m$; if $\|R_m\|_F < \varepsilon$, stop;
21: Compute polynomial precondioner $Q_{m-1}(x,y)$:
22: $S_m y_m = (a_0, \ldots, a_{m-1})^T$,
23: $Q_{m-1}(x,y) = \sum_{j=0}^{m-1} a_j x^j y^j$.
24: Compute the solution of $Q_{m-1}(B^T; A)(B^T \otimes A - I_{ns}) vec(X) = Q_{m-1}(B^T; A) vec(C)$; by solving the least square problem (28) and $X_k = X_m + V_k * y_k$;
25: Set $X_0 = X_k$
26: end for

**Example 3.** The purpose of this example is to illustrate the numerical behavior of BGMRES and BGMRES with ILU, SSOR and polynomial preconditioning and $m = 5$. The matrix $A \in \mathbb{R}^{2 \times 2}$ is a bidiagonal matrix with entries $a_1 = 2, a_2 = 4, a_3 = 6, 2, 3, 4, \ldots, 64$ on the main diagonal, and super diagonal entries $1$. The matrix $B$ is the same as $A$. Also, the right hand side of the generalized linear Sylvester equation $AXB - X = C$ is such that $X = 1$ is the exact solution. The numerical computations are carried out with $m = 6, k = 10$. The convergence curves plotted in Fig. 1. From Fig. 1, we can see that BGMRES with polynomial precondioning is faster than the other methods.

**Example 4.** In this example, we use the matrices $A = \text{tridiag}(1+d, 1-d)$, $B = A$ with $d = 5$ and $n = s = 64$ and $k = m = 25$. We evaluate the performance of the four block...
solverson Fig. 2, we show that BGMRES with polynomial precon-
ingtion is faster convergence than the other methods.

Figure 2 The convergence result of BGMRES with SSOR, ILU, polynomial precon-
ingtion and BGMRES methods

Example 5. The matrices are the same as in Example 4, but d=8, m=k=10. We compare the convergence behavior of BGMRES, PBGMRES, PBGMRES (ILU, SSOR). The results are summarized in Table III.

<table>
<thead>
<tr>
<th>Method</th>
<th>Iter.</th>
<th>Res.norm</th>
<th>Cpute(time/seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGMRES(polynomial)</td>
<td>5000</td>
<td>0.8503</td>
<td>1.438e+03</td>
</tr>
<tr>
<td>PBGMRES(ILU)</td>
<td>7</td>
<td>3.8132e-10</td>
<td>14.235</td>
</tr>
<tr>
<td>PBGMRES(SSOR)</td>
<td>19</td>
<td>5.6427e-10</td>
<td>2.5800</td>
</tr>
<tr>
<td>NSCG</td>
<td>88</td>
<td>1.1340e-10</td>
<td>1.798e+03</td>
</tr>
<tr>
<td></td>
<td>218</td>
<td>NaN</td>
<td>-</td>
</tr>
</tbody>
</table>

V. CONCLUSION

We have derived a polynomial preconditioning block GMRES method for the generalized Sylvester matrix equation. It is observed by examples that PBGMRES is faster than some other block solvers.

APPENDIX A

PROOF OF THEOREM 2

Since

\[ A r_0 = (B^T \otimes A) r_0 - r_0. \]

Thus, \( A r_0 \) is a combination of \((B^T \otimes A)r_0\) and \( r_0 \) and \( \text{span}\{r_0, Ar_0\} = \text{span}\{r_0, (B^T \otimes A)r_0\} \).

Next, we consider \( A^2 r_0 \),

\[ A^2 r_0 = (B^T \otimes A)^2 r_0 - 2(B^T \otimes A)r_0 + r_0. \]

Therefore,

\[ \text{span}\{r_0, Ar_0, A^2 r_0\} = \text{span}\{r_0, (B^T \otimes A)r_0, (B^T \otimes A)^2 r_0\}. \]

Continuing this, the two subspaces are the same. This establishes the claim.

APPENDIX B

PROOF OF THEOREM 3

The proof of theorem 3 proceeds as:

\[
\min_{X \in X_0 + \mathcal{G}K_m(A, R_0, B)} \|R\|_F = \min_{X \in X_0 + \mathcal{G}K_m(A, R_0, B)} \|C - AXB + X\|_F.
\]

Let \( X^* \) be the exact solution of the matrix equation (1), therefore

\[
\min_{X \in X_0 + \mathcal{G}K_m(A, R_0, B)} \|A(X^* - X)B - (X^* - X)\|_F
\]

since \( X \in X_0 + \mathcal{G}K_m(A, R_0, B) \), there exists a \( Z \in \mathcal{G}K_m(A, R_0, B) \) such that \( X = X_0 + Z \) hence

\[
\min_{Z \in \mathcal{G}K_m(A, R_0, B)} \|A(Z^* - Z)B - (Z^* - Z)\|_F
\]

by (4), we have

\[
\min_{Y \in AGK_m(A, R_0, B)} \|Y^* - Y\|_F = \|Y^* - Y_m\|_F
\]

where \( Y = AZB - Z \). By corollary 1.39, [14],

\[
\min_{Y \in AGK_m(A, R_0, B)} \|Y^* - Y\|_F = \|Y^* - Y_m\|_F
\]

if and only if

\[
\begin{cases}
Y_m \in AGK_m(A, R_0, B)B, \\
Y^* - Y_m \perp AGK_m(A, R_0, B)B.
\end{cases}
\]

Since \( Y^* - Y_m = R_m \), then \( R_m = C - AX_mB + X_m \perp F \) \( AGK_m(A, R_0, B)B \).

REFERENCES


