On the Numerical Solution of Large Scale Sylvester Matrix Equations

Amer Kaabi

Department of Mathematics
Faculty of Basic Sciences
Persian Gulf University, Bushehr, Iran
kaabi@pgu.ac.ir, kaabi_amer@yahoo.com

Abstract

This paper presents equivalent forms of the Sylvester matrix equations. These equivalent forms allow us to use block linear methods for solving large Sylvester matrix equations. In each step of these iterative methods we use global FOM or global GMRES algorithm for solving an auxiliary block matrix equations. Also, some numerical experiments for obtaining the numerical approximated solution of Sylvester matrix equations are given.

Mathematics Subject Classification: 65F10

Keywords: Sylvester matrix equations, Global FOM method, Global GMRES method

1 Introduction

The Sylvester matrix equations is very important in many control applications. some example of stability analysis of nonlinear systems [12], optimal $\mathcal{H}_\infty$ control design of linear systems [3, 4], iterative solution of the Riccati equation [17], balancing of linear systems [14], and model reduction methods based on balancing [4, 14]. The Sylvester matrix equations

$$AX - XB = C,$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{p \times p}$, $C \in \mathbb{R}^{n \times p}$. The necessary and sufficient condition for (1) to have a unique solution is that

$$\lambda(A) \cap \lambda(B) = \emptyset,$$
where $\lambda(S)$ is the spectrum of matrix $S$ [2, 5, 11, 13]. Under this assumption, a unique solution $X \in \mathbb{IR}^{n \times p}$, exists. There are a number of direct methods for solving the Sylvester matrix equations (1) numerically, the most important of which are the Bartels-Stewart method [1] and the Hammarling method [8]. These algorithms are based on the Schur factorization and the successive solution of triangular systems. Another algorithm for this case is the one proposed by Golub, Nash and Van Loan [6] which is a modification of the algorithm of Bartels-Stewart, but using the Hessenberg factorization instead. Unfortunately, these methods ignore any sparsity in the equation and are not very attractive for parallelization. A rough estimation of the complexity of the Bartels-Stewart method and the Hammarling method gives about $25n^3$ flops and $3n^2$ words of memory. When $n$ is large, we can use the Krylov-subspaces Galerkin and minimal residual algorithms which have been presented by D.Y. Hu and L. Reichel [9]. In [18], Simoncini extended the work of Hu and Reichel to block form using the idea developed in [9]. El Guennouni et al. introduced new Krylov methods to solve (1) based on block-Arnoldi and nonsymmetric block-Lanczos algorithms [7]. In [16], M. Robbe and M. Sadkane discussed convergence properties of the block GMRES and FOM methods for solving large Sylvester equations of the form (1). Marlliny Monsalve [15], proposed block linear methods for large scale Sylvester matrix equations. He proved that the proposed methods are convergent if one of the conditions $\| A^{-1} \| \| B \| < 1$ or $\| B^{-1} \| \| A \| < 1$ are satisfied. But, for some important applications these conditions are not satisfied, these proposed schemes cannot be used. Also the these schemes cannot be applied for solving the well-known Lyapunov matrix equations. To remedy these difficulties, we develop these methods by proposing equivalent forms of Sylvester matrix equations.

Throughout this paper, we use the following notations. $e^{(k)}_1$ denotes the first axis vector of dimension $k$, $\mathbb{IE} = \mathbb{IE}_{n \times p}$ denotes the vector space, on the field $\mathbb{IR}$, of rectangular matrices of dimension $n \times p$. For $X$ and $Y$ in $\mathbb{IE}$, we define the inner product $\langle X, Y \rangle_F = \text{tr}(X^TY)$, where $\text{tr}(Z)$ denotes the trace of the square matrix $Z$ and $X^T$ denotes the transpose of the matrix $X$. The associated norm is the well-known Frobenius norm denoted by $\| . \|_F$. For a matrix $V \in \mathbb{IE}$, the block Krylov subspace $\mathcal{K}_m(A, V)$ is the subspace generated by the columns of the matrices $V, Av, ..., A^{m-1}V$. A set of members of $\mathbb{IE}$ is said to be F-Orthonormal if it is orthonormal with respect to scaler product $\langle ., . \rangle_F$.

This paper is organized as follows. In Section 2, a brief description of the global FOM and GMRES methods for solving matrix equations are given, and the iterative methods for solving Sylvester matrix equations are summarized in Section 3. In Section 4 the equivalence forms of Sylvester matrix equations are presented. In Section 5 some numerical examples are tested. Finally, Section 6 summarizes the main conclusion of this paper.
2 Global FOM and GMRES Algorithms For Matrix Equations

Global Arnoldi process construct an F-Orthonormal basis $V_1, V_2, ..., V_m$, of Krylov subspace $\mathcal{K}_m(A,V)$. By using modified global Arnoldi algorithm, we can obtain an F-Orthonormal basis of $\mathcal{K}_m(A,V)$. This algorithm is as follows.

**Algorithm 1. Modified global Arnoldi algorithm**

1. Choose an $n \times p$ matrix $V_1$ such that $\| V \|_F = 1$.
2. For $j = 1, ..., m$ Do:
3. $V_j = AV_j$,
4. For $i = 1, ..., j$ Do:
5. $h_{ij} = \text{tr}(V_i^T V_j)$,
6. $\tilde{V}_j = V_j - h_{ij} V_i$,
7. $h_{j+1,j} = \| \tilde{V}_j \|_F$
8. EndDo.
9. If $h_{j+1,j} = 0$ then stop.
10. Set $V_{j+1} = \tilde{V}_j / h_{j+1,j}$.
11. EndDo

We denote by $H_m$ the upper $m \times m$ Hessenberg matrix whose nonzero entries are the scalars $h_{ij}$ and the $(m + 1) \times m$ matrix $\tilde{H}_m$ is the same as $H_m$ except for an additional row whose only nonzero element is $h_{m+1,m}$ in the $(m+1,m)$ position.

- **The Global Full Orthogonalization Method**
Let $X_0 \in \mathbb{R}^{n \times p}$ be an initial approximate solution of the matrix equations

$$SX = G,$$  \hspace{1cm} (2)

then the residual matrix vector is given by

$$R_0 = G - SX_0.$$ 

The global full Orthogonalization method (GL-FOM) seeks an approximate solution $X_m$ from the affine subspace $X_0 + \mathcal{K}_m(S, R_0)$ of dimension $m$ by imposing the Galerkin condition

$$R_m = G - S(X_m) \perp_F \mathcal{K}_m(S, R_0).$$

This method uses the global Arnoldi process with $V_1 = R_0 / \| R_0 \|_F$, to compute an F-Orthonormal basis of Krylov-subspace $\mathcal{K}_m(S, R_0)$. We can summarize the GL-FOM algorithm as follows. (See [10])
2898

Algorithm 2. Restarted global FOM algorithm (GL-FOM).
1. Choose an initial approximate solution $X_0$ and a tolerance $\epsilon$.
2. Compute $R_0 = G - SX_0$ and set $V_1 = R_0/\| R_0 \|_F$.
3. If $\| R_0 \|_F \leq \epsilon$ then stop.
4. Generate F-orthonormal basis $V_1, V_2, ..., V_m$ of $K_m(S, R_0)$ and
   the matrix $H_m$ by Algorithm 1.
5. Solve $H_my_m = \| R_0 \|_F e_1^{(m)}$ for $y_m = (y_1^{(m)}, y_2^{(m)}, ..., y_m^{(m)})$.
6. Compute $X_m = X_0 + \sum_{i=1}^m y_i^{(m)} V_i$ and set $X_0 = X_m$.
7. Go to 2.

- The Global Generalized Minimal Residual Method
Starting with an initial guess $X_0$ and the corresponding residual $R_0$, the global
generalized minimal residual method for solving matrix equations, constructs, at step $m$, the new approximation $X_m$ of the matrix equations (2) which satisfies

$$X_m - X_0 = Z_m \in K_m(S, R_0)$$

with the F-orthogonality relation

$$R_m \perp_F SK_m(S, R_0).$$

By the same manner as in GL-FOM method we can summarize the GL-GMRES algorithm as follows. (See [10]).

Algorithm 3. Restarted global GMRES algorithm (GL-GMRES).
1. Choose an initial approximate solution $X_0$ and a tolerance $\epsilon$.
2. Compute $R_0 = G - SX_0$ and set $V_1 = R_0/\| R_0 \|_F$.
3. If $\| R_0 \|_F < \epsilon$ then stop.
4. Generate F-Orthonormal basis $V_1, V_2, ..., V_m$ of $K_m(S, R_0)$ and
   the matrix $\tilde{H}_m$ by Algorithm 1.
5. Compute $y_m$, the minimizer of the problem $\min_{y \in \mathbb{R}^m} \| R_0 \|_F e_1^{(m+1)} - \tilde{H}_my \|_2$
6. Compute $X_m = X_0 + \sum_{i=1}^m y_i^{(m)} V_i$ and set $X_0 = X_m$.
7. Go to 2.

3 Iterative Methods for Solving Sylvester Matrix Equations

The main idea of this methods is to write the Sylvester matrix equations as a
block linear systems and then use some suitable algorithms. In [15] Marliny
Monsalve proposed these methods as follows:
• Define \( U = AX \),
• Set \( U = XB + C \).
Or
• Define \( U = XB \),
• Set \( U = AX - C \).
These generate two iterative algorithms:

**Algorithm 4. Using: \( U = AX \).**
1. Choose an initial approximate solution \( X_0 \in \mathbb{R}^{n \times p} \).
2. Compute \( U_0 = C + X_0B \).
3. For \( k = 0, 1, ..., \) until convergence
4. Solve \( AX_{k+1} = U_{k+1} \).
5. Set \( U_{k+1} = C + X_{k+1}B \).
6. End For.

**Algorithm 5. Using: \( U = XB \).**
1. Choose an initial approximate solution \( X_0 \in \mathbb{R}^{n \times p} \).
2. Compute \( U_0 = AX_0 - C \).
3. For \( k = 0, 1, ..., \) until convergence
4. Solve \( X_{k+1}B = U_{k+1} \).
5. Set \( U_{k+1} = AX_{k+1} - C \).
6. End For.

These algorithms require the solution of one block linear system and one matrix-matrix product per iteration. Also, these algorithms allow, depending on the dimension of \( A \) and \( B \), to discriminate easily which algorithms is the most convenient.

**Theorem 1.** Let \( A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{p \times p}, C \in \mathbb{R}^{n \times p} \), and \( \| \cdot \| \) be an induced norm. If \( \| A^{-1} \| \| B \| < 1 \) then equation (2) has a unique solution, and the sequence \( \{ X_k \}_{k \geq 0} \) generate by algorithm 4 converge to the solution. Moreover, if \( \| B^{-1} \| \| A \| < 1 \) then equation (2) has a unique solution, and the sequence \( \{ X_k \}_{k \geq 0} \) generate by algorithm 5 converge to the solution.

**Proof.** See [15].

Let \( R_k \), be the residual matrix at iteration \( k \), obtained by algorithm 4 or algorithm 5. The computation of \( R_k \), from the definition

\[
R_k = C - AX_k + X_kB
\]

is very expensive, since this require two matrix-matrix products per iteration. The following result provides an equivalent and less expensive way to calculate the residual.

**Theorem 2.** Let \( \{ X_k \}_{k \geq 0} \) be the sequence generated by algorithm 4. The following expressions for the residual matrix are equivalent.
\( R_k = C - AX_k + X_kB. \)
\( R_k = U_k - U_{k-1}. \)
\( R_k = (X_k - X_{k-1})B. \)

**Proof.** See [15].

Also, by algorithm 5 the following expressions are equivalent, see [15].
\( R_k = C - AX_k + X_kB. \)
\( R_k = U_k - U_{k-1} - 1. \)
\( R_k = A(X_k - X_{k-1}). \)

Finally, from theorem 2 we have the following corollary which yields a more convenient stopping criterion.

**Corollary 1.** Let \( \{X_k\}_{k \geq 0} \) be the sequence generated by algorithm 4 or by algorithm 5. If \( \|X_k - X_{k-1}\| \to 0 \), then \( \|R_k\| \to 0 \).

**Proof.** See [15].

### 4 The Equivalent Forms of Sylvester Matrix Equations

For algorithm 4, the matrix of coefficients of the internal system is of dimension \( n \), while for algorithm 5, is of dimension \( p \), therefore if \( n > p \) it is convenient to use algorithm 5, otherwise it is convenient to use algorithm 4.

Now we can consider the following three cases for matrices \( A \) and \( B \).

**case a:** If \( n \gg p \) and \( B \) is a singular matrix.

In this case, because the condition \( \|B^{-1}\| \|A\| < 1 \) does not satisfy, theorem 1 does not guarantee convergence. Therefore, we do not propose to use algorithm 5 to obtain an approximate solution of Sylvester matrix equations (1). To remedy this difficulty we propose to consider the Sylvester matrix equation (1) in the following equivalent form

\[
AX - X(B + \tau I) = C - \tau X,
\]

where \( \tau \) is a scalar such that the matrix \( B + \tau I \) be a nonsingular and

\[
\| (B + \tau I)^{-1}\| \|A\| \ll 1.
\]

Thus we can compute an approximated solution of equation (1) by applying algorithm 5 for Sylvester matrix equations

\[
AX - XB_\tau = C_\tau(X),
\]

where

\[
B_\tau = B + \tau I \quad \text{and} \quad C_\tau(X) = C - \tau X.
\]
**Case b:** If \( p \gg n \) and \( A \) is a singular matrix.

In this case, because the condition \( \| A^{-1} \| \| B \| < 1 \) does not satisfy, theorem 1 does not guarantee convergence. Therefore, we do not propose to use algorithm 4 to obtain an approximate solution of Sylvester matrix equations (1). To remedy this difficulty we propose to consider the Sylvester matrix equation (1) in the following equivalent form

\[
(A - \tau I)X - XB = C - \tau X,
\]

where \( \tau \) is a scalar such that the matrix \( A - \tau I \) be a nonsingular and

\[
\| (A - \tau I)^{-1} \| \| B \| \ll 1.
\]

Thus we can compute an approximated solution of equation (1) by applying algorithm 5 for Sylvester matrix equations

\[
A_{\tau}X - XB = C_{\tau}(X), \tag{4}
\]

where

\[
A_{\tau} = A - \tau I.
\]

**Case c:** If \( B = A \) or \( B = A^T \).

In this case, theorem 1 does not guarantee convergence, since \( \| A^{-1} \| \| A \| = k(A) \geq 1 \). Unfortunately, this include the well-known Lyapunov matrix equations. To remedy this difficulty we propose to consider the Sylvester matrix equation (1) in the following equivalent form

\[
(A - \tau I)X - X(B + \tau I) = C - 2\tau X,
\]

where \( \tau \) is an appropriate scalar such that

\[
\| (A - \tau I)^{-1} \| \| (B + \tau I) \| \ll 1.
\]

Thus we can compute an approximated solution of equation (1) by applying algorithm 5 for Sylvester matrix equations

\[
A_{\tau}X - XB_{\tau} = C_{\tau}(X), \tag{5}
\]

where

\[
C_{\tau}(X) = C - 2\tau X.
\]
5 Numerical Experiments

In this section, we present the performance of the Algorithm 4 and Algorithm 5. For solving the internal block linear systems in Algorithms 4 and 5 we use GL-FOM(m) and GL-GMRES(m) methods. For all the examples, we have used the stopping criterion
\[ \| R_k = C - AX_k + X_kB \|_F \leq 10^{-6}, \]
and the maximum number of iterations allowed, is set to 1000. In addition, we have used the following stopping criterion for stopping the GL-FOM(m) and GL-BGS(m) algorithms which has been used at step 4 of Algorithms 4 and 5:
\[ \| \tilde{R}_k \|_F \leq 10^{-10}, \]
where
\[ \tilde{R}_k = U_k - AX_k \text{ or } \tilde{R}_k = U_k - X_kB. \]

For all the experiments, the initial guess is \( X_0 = 0 \). The right-hand-side matrix \( C \) is chosen so that \( X = (x_{ij}) \) with \( x_{ij} = 1 \), \( 1 \leq i \leq n, 1 \leq j \leq p \), solves equation (1). We use the matrices \( A \) and \( B \) as follows:
\[ A = -\text{tridiag}(-1 + \frac{10}{n+1}, 2, -1 + \frac{10}{n+1}) \]
and
\[ B = \text{tridiag}(-1 + \frac{10}{p+1}, 2, -1 + \frac{10}{p+1}), \]
where \( n \) and \( p \) are the order of matrices \( A \) and \( B \), respectively. The results obtained by the Algorithms 4 and 5 are reported in Tables 1-4 with different values of \( m, n, p \) and \( \tau \). The results of the case \( \tau = 0 \) correspond to the results of the Algorithms 4 and 5 for solving equation (1).

Table 1. \( m = 5, p = 10 \).
Number of iterations (and CPU Times, second) of the GL-FOM(m).

<table>
<thead>
<tr>
<th>( n ) ( \tau )</th>
<th>0</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>†</td>
<td>7(0.65)</td>
<td>6(0.55)</td>
<td>5(0.45)</td>
<td>5(0.45)</td>
<td>5(0.45)</td>
</tr>
<tr>
<td>600</td>
<td>†</td>
<td>7(1.22)</td>
<td>6(1.05)</td>
<td>5(0.86)</td>
<td>5(0.85)</td>
<td>5(0.85)</td>
</tr>
<tr>
<td>800</td>
<td>†</td>
<td>7(2.17)</td>
<td>6(1.85)</td>
<td>5(1.53)</td>
<td>5(1.53)</td>
<td>5(1.53)</td>
</tr>
<tr>
<td>1000</td>
<td>†</td>
<td>7(3.33)</td>
<td>6(2.87)</td>
<td>5(2.41)</td>
<td>5(2.41)</td>
<td>5(2.41)</td>
</tr>
<tr>
<td>1200</td>
<td>†</td>
<td>7(4.72)</td>
<td>6(4.05)</td>
<td>5(3.38)</td>
<td>5(3.38)</td>
<td>5(3.38)</td>
</tr>
<tr>
<td>1400</td>
<td>†</td>
<td>7(6.42)</td>
<td>6(5.42)</td>
<td>5(4.62)</td>
<td>5(4.62)</td>
<td>5(4.62)</td>
</tr>
<tr>
<td>1600</td>
<td>†</td>
<td>7(8.24)</td>
<td>6(6.75)</td>
<td>5(5.72)</td>
<td>5(5.72)</td>
<td>5(5.72)</td>
</tr>
<tr>
<td>1800</td>
<td>†</td>
<td>7(10.2)</td>
<td>6(8.85)</td>
<td>5(7.21)</td>
<td>5(7.21)</td>
<td>5(7.21)</td>
</tr>
<tr>
<td>2000</td>
<td>†</td>
<td>7(12.6)</td>
<td>6(10.9)</td>
<td>5(9.15)</td>
<td>5(9.15)</td>
<td>5(9.15)</td>
</tr>
</tbody>
</table>

†=no solution has been obtained after 1000 iterations by GL-FOM(m).
### Table 2. \( m = 5, \ p = 10 \).
Number of iterations (and CPU Times, second) of the GL-GMRES(m).

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \tau )</th>
<th>0</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>( \dagger )</td>
<td>7(0.67)</td>
<td>6(0.57)</td>
<td>5(0.46)</td>
<td>5(0.46)</td>
<td>5(0.46)</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>( \dagger )</td>
<td>7(1.29)</td>
<td>6(1.09)</td>
<td>5(0.88)</td>
<td>5(0.88)</td>
<td>5(0.88)</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>( \dagger )</td>
<td>7(2.22)</td>
<td>6(2.01)</td>
<td>5(1.57)</td>
<td>5(1.57)</td>
<td>5(1.57)</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>( \dagger )</td>
<td>7(3.47)</td>
<td>6(2.93)</td>
<td>5(2.46)</td>
<td>5(2.46)</td>
<td>5(2.46)</td>
<td></td>
</tr>
<tr>
<td>1200</td>
<td>( \dagger )</td>
<td>7(4.92)</td>
<td>6(4.12)</td>
<td>5(3.42)</td>
<td>5(3.42)</td>
<td>5(3.42)</td>
<td></td>
</tr>
<tr>
<td>1400</td>
<td>( \dagger )</td>
<td>7(6.51)</td>
<td>6(5.56)</td>
<td>5(4.66)</td>
<td>5(4.66)</td>
<td>5(4.66)</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>( \dagger )</td>
<td>7(8.57)</td>
<td>6(6.97)</td>
<td>5(5.82)</td>
<td>5(5.82)</td>
<td>5(5.82)</td>
<td></td>
</tr>
<tr>
<td>1800</td>
<td>( \dagger )</td>
<td>7(10.4)</td>
<td>6(8.95)</td>
<td>5(7.32)</td>
<td>5(7.32)</td>
<td>5(7.32)</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>( \dagger )</td>
<td>7(12.8)</td>
<td>6(11.1)</td>
<td>5(9.2)</td>
<td>5(9.2)</td>
<td>5(9.2)</td>
<td></td>
</tr>
</tbody>
</table>

\( \dagger \)=no solution has been obtained after 1000 iterations by GL-GMRES(m).

### Table 3. \( m = 5, \ p = 20 \).
Number of iterations (and CPU Times, second) of the GL-FOM(m).

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \tau )</th>
<th>0</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>( \dagger )</td>
<td>7(0.82)</td>
<td>6(0.75)</td>
<td>5(0.60)</td>
<td>5(0.60)</td>
<td>5(0.60)</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>( \dagger )</td>
<td>7(1.91)</td>
<td>6(1.64)</td>
<td>5(1.38)</td>
<td>5(1.38)</td>
<td>5(1.38)</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>( \dagger )</td>
<td>7(3.24)</td>
<td>6(2.92)</td>
<td>5(2.43)</td>
<td>5(2.43)</td>
<td>5(1.43)</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>( \dagger )</td>
<td>7(5.11)</td>
<td>6(4.39)</td>
<td>5(3.25)</td>
<td>5(3.25)</td>
<td>5(3.25)</td>
<td></td>
</tr>
<tr>
<td>1200</td>
<td>( \dagger )</td>
<td>7(7.22)</td>
<td>6(6.31)</td>
<td>5(5.18)</td>
<td>5(5.18)</td>
<td>5(5.18)</td>
<td></td>
</tr>
<tr>
<td>1400</td>
<td>( \dagger )</td>
<td>7(9.79)</td>
<td>6(8.21)</td>
<td>5(6.97)</td>
<td>5(6.97)</td>
<td>5(6.97)</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>( \dagger )</td>
<td>7(12.7)</td>
<td>6(11.2)</td>
<td>5(9.05)</td>
<td>5(9.05)</td>
<td>5(9.05)</td>
<td></td>
</tr>
<tr>
<td>1800</td>
<td>( \dagger )</td>
<td>7(15.8)</td>
<td>6(13.6)</td>
<td>5(11.4)</td>
<td>5(11.4)</td>
<td>5(11.4)</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>( \dagger )</td>
<td>7(19.6)</td>
<td>6(16.8)</td>
<td>5(14.1)</td>
<td>5(14.1)</td>
<td>5(14.1)</td>
<td></td>
</tr>
</tbody>
</table>

\( \dagger \)=no solution has been obtained after 1000 iterations by GL-FOM(m).

### Table 4. \( m = 5, \ p = 20 \).
Number of iterations (and CPU Times, second) of the GL-GMRES(m).

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \tau )</th>
<th>0</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>( \dagger )</td>
<td>7(0.89)</td>
<td>6(0.80)</td>
<td>5(0.65)</td>
<td>5(0.65)</td>
<td>5(0.65)</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>( \dagger )</td>
<td>7(2.01)</td>
<td>6(1.75)</td>
<td>5(1.51)</td>
<td>5(1.51)</td>
<td>5(1.51)</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>( \dagger )</td>
<td>7(3.51)</td>
<td>6(3.01)</td>
<td>5(2.53)</td>
<td>5(2.53)</td>
<td>5(1.53)</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>( \dagger )</td>
<td>7(5.25)</td>
<td>6(4.52)</td>
<td>5(3.75)</td>
<td>5(3.75)</td>
<td>5(3.75)</td>
<td></td>
</tr>
<tr>
<td>1200</td>
<td>( \dagger )</td>
<td>7(7.27)</td>
<td>6(6.35)</td>
<td>5(5.25)</td>
<td>5(5.25)</td>
<td>5(5.25)</td>
<td></td>
</tr>
<tr>
<td>1400</td>
<td>( \dagger )</td>
<td>7(9.95)</td>
<td>6(8.35)</td>
<td>5(7.07)</td>
<td>5(7.07)</td>
<td>5(7.07)</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>( \dagger )</td>
<td>7(12.8)</td>
<td>6(11.3)</td>
<td>5(9.85)</td>
<td>5(9.85)</td>
<td>5(9.85)</td>
<td></td>
</tr>
<tr>
<td>1800</td>
<td>( \dagger )</td>
<td>7(16.2)</td>
<td>6(13.5)</td>
<td>5(11.5)</td>
<td>5(11.5)</td>
<td>5(11.5)</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>( \dagger )</td>
<td>7(19.7)</td>
<td>6(16.9)</td>
<td>5(14.2)</td>
<td>5(14.2)</td>
<td>5(14.2)</td>
<td></td>
</tr>
</tbody>
</table>
\[ \hat{=} \text{no solution has been obtained after 1000 iterations by GL-FOM(m).} \]

## 6 Conclusion

We have proposed equivalent forms of Sylvester matrix equations for solving the Sylvester equations \( AX - XB = C \), by applying Algorithms 4 and 5. These iterative methods use the GL-FOM(m) or Gl-GMRES(m) algorithm in each step for solving an auxiliary block linear system. The numerical experiments show that the solution of Sylvester matrix equations can be obtained with high accuracy applying the Algorithms 4 and 5 for the equivalent forms of Sylvester matrix equations. In addition, we observe that as \( \tau \) increases, the number of iterations of the Algorithms 4 and 5 and the CPU-Times are decreases. Finding an optimal \( \tau \) is an open problem and can be concern of a future work.

## References


Received: April 16, 2007