Filomat 27:8 (2013), 1437–1446
DOI 10.2298/FIL1308437Z

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A Global Transpose-free Method with Quasi-minimal Residual Strategy for the Sylvester Equations

Ke Zhang[a], Chuanqing Gu[b]

[a]*Department of Mathematics, Shanghai University, P. R. China*
[b]*Department of Mathematics, Shanghai University, P. R. China*

Abstract. The Sylvester equation has numerous applications in control and system theory, power system, linear algebra, model reduction and signal processing. In the present paper, a global transpose-free quasi-minimal residual method for the Sylvester equations is proposed. The resulting method, with short-term recurrences, does not involve the multiplication with the transpose of the coefficient matrices, and often exhibits smoother convergence behavior than some other existing global methods. Finally, several numerical examples are reported by comparing the proposed method with some global methods.

1. Introduction

Consider the Sylvester equations of the form

$$AX + XB = C, \tag{1}$$

where $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{m\times n}$ are given matrices. Throughout this paper, we assume that (1) has a unique solution, i.e., matrices A and $-B$ have no eigenvalue in common [4, 8].

The Sylvester equation (1) arises in many problems concerning control and system theory, signal processing and power systems. Conventionally, Eq.(1) can be solved equivalently by converting it into a linear system in the Kronecker product form, that is, $\mathtt{Ax} = \mathtt{b}$, where $\mathtt{A} = I \otimes A + B^T \otimes I$, $\mathtt{x}$ and $\mathtt{b}$ are augmented vectors by stacking the columns of X and C respectively, with $\otimes$ the Kronecker product and B^T the transpose of B. This approach, however, can be encountered with the problems concerning computational cost and stability.

Benchmark methods for the Sylvester equation (1) are the Bartels-Stewart and the Hessenberg-Schur methods [2, 9]. Such methods are usually classified as direct methods and often implemented for solving systems of reasonable small size. When the system (1) becomes large and sparse, however, the iterative methods may be preferred. These iterative methods include, to name just a few of them, the ADI method [3, 14, 16], the Smith's method [11, 18], the matrix sign function method [13], the HSS method [1] and the

2010 *Mathematics Subject Classification*. 65F10; 65F30; 15A24
Keywords. Sylvester equation; Quasi-minimal residual; CGS; Krylov subspaces; Global methods
Received: 20 January 2013; Accepted: 11 May 2013
Communicated by Professor Dragana Cvetkovic Ilic.
Research supported by Shanghai Natural Science Foundation (10ZR1410900), Key Disciplines of Shanghai Municipality (S30104) and Innovation Program of Shanghai Municipal Education Commission(13ZZ068).
Corresponding author: Chuanqing Gu
Email addresses: xznuzk123@126.com (Ke Zhang), cqgu@shu.edu.cn (Chuanqing Gu)

gradient-based methods [6, 10]. For more about numerical methods for the Sylvester equation (1), we refer to Datta's monograph [4].

Recently, by using the global oblique and orthogonal projections of the initial matrix residual onto a matrix Krylov subspace, Jbilou etc. proposed the global version methods for matrix equations [12]. Datta etc. presented a global Arnoldi method and corresponding theory for solving multi-input Sylvester-observer equations from the construction of the Luenberger observer in control theory [5]. Using the global techniques, some new methods for matrix equations, for instance, the global conjugate squared method (Gl-CGS) and its generalization [20], and the global bi-conjugate residual method (Gl-BCR) [21], have been proposed following [12]. Though some of these methods were originally contrived for linear systems with multiple right-hand sides, they can also be applied for the Sylvester equation in a similar way. One possible drawback of Gl-CGS is that its convergence curves are sometimes rather erratic. To circumvent this difficulty, we generate in this paper a smooth global method with an approach analogous to the derivation of the TFQMR method [7] from the conjugate gradient squared (CGS) method [19] for general linear systems. The resulting method does not involve the matrix multiplication with the transpose of A or B in (1) and often yields a much smoother convergence than Gl-CGS.

This paper is organized as follows. In Section 2, we sketch out the conjugate gradient squared method and its global variant. In Section 3, we present our idea by rewriting the relations of residuals and the corresponding auxiliary matrices in the Gl-CGS method, which results in a new iterative method by minimizing the norm of the quasi-residual. In Section 4, some numerical examples are used to validate the effectiveness of the proposed method. Finally, some concluding remarks are given in Section 5.

2. Preliminaries and notation

In this section, we introduce some preliminary results about the conjugate gradient squared method (CGS), its global variant and notation used in the global methods, the results of which will be employed in the next section to derive the new algorithm of this paper. The CGS method was originally proposed by Sonneveld [19] to solve the general nonsymmetric linear systems. Further, Zhang etc. have extended the CGS algorithm to solve linear systems with multiple right-hand sides, i.e., $MX = N$, which yields the global CGS algorithm (Gl-CGS) [20]. To clarify the derivation of our new algorithm in Section 3, we give the Gl-CGS algorithm below. For more detailed description of Gl-CGS, we refer to [20].

Algorithm 1 Gl-CGS algorithm.

Choose an initial guess X_0 and $\widetilde{R}_0$, $R_0 = N - MX_0$, $P_0 = W_0 = R_0$.
for $j = 0, 1, \cdots$ until convergence **do**
  $\alpha_j = \frac{<R_j, \widetilde{R}_0>_F}{<MP_j, \widetilde{R}_0>_F}$;
  $Q_j = W_j - \alpha_j MP_j$;
  $X_{j+1} = X_j + \alpha_j(W_j + Q_j)$;
  $R_{j+1} = R_j - \alpha_j M(W_j + Q_j)$;
  $\beta_j = \frac{<R_{j+1}, \widetilde{R}_0>_F}{<R_j, \widetilde{R}_0>_F}$;
  $W_{j+1} = R_{j+1} + \beta_j Q_j$;
  $P_{j+1} = W_{j+1} + \beta_j(Q_j + \beta_j P_j)$;
end for

Next we introduce some notation which has been used in the derivation of the global Krylov subspace methods [12]. The matrix Frobenius norm and 2-norm are denoted by $\|\cdot\|_F$ and $\|\cdot\|_2$ respectively. For matrices X and Y with conforming dimensions, we define the inner product $< X, Y >_F = \mathrm{tr}(X^T Y)$, where $\mathrm{tr}(\cdot)$ represents the trace of a square matrix. Additionally, X is said to be F-orthogonal to Y if $< X, Y >_F = 0$.

Denote by $\mathcal{K}_s(\mathcal{A}, R_0)$ the *matrix Krylov subspace*

$$\mathcal{K}_s(\mathcal{A}, R_0) = \left\{ \sum_{i=0}^{s-1} \alpha_i \mathcal{A}^i(R_0); \alpha_i \in \mathbb{R} \right\},$$

where the operator $\mathcal{A}$ is defined as: $\mathcal{A} : X \to AX + XB$ with $X \in \mathbb{R}^{m\times n}$, and $R_0 = C - \mathcal{A}(X_0)$ is the initial residual. With the operator $\mathcal{A}$, Algorithm 1 can also be extended to solve the Sylvester equation (1) by replacing the matrix M with $\mathcal{A}$.

Let $\overline{\mathbf{U}}_s = [U_0, U_1, \cdots U_{s-1}] \in \mathbb{R}^{m\times ns}$, where $U_i \in \mathbb{R}^{m\times n}$, $i = 0, 1, \cdots, s-1$. For $\mathbf{b} = [b_0, b_1, \cdots, b_{s-1}]^T \in \mathbb{R}^s$, as in [12], the notation $*$ denotes the product

$$\overline{\mathbf{U}}_s * \mathbf{b} = \sum_{i=0}^{s-1} b_i U_i. \tag{2}$$

Further, for matrix $G \in \mathbb{R}^{s\times s}$ we define

$$\overline{\mathbf{U}}_s * G = [\overline{\mathbf{U}}_s * G_{\cdot,1}, \overline{\mathbf{U}}_s * G_{\cdot,2}, \cdots, \overline{\mathbf{U}}_s * G_{\cdot,s}], \tag{3}$$

where $G_{\cdot,j}$ stands for the jth column of the matrix G. It can be easily verified that

$$\overline{\mathbf{U}}_s * (\mathbf{a} + \mathbf{b}) = (\overline{\mathbf{U}}_s * \mathbf{a}) + (\overline{\mathbf{U}}_s * \mathbf{b}), \quad (\overline{\mathbf{U}}_s * G) * \mathbf{a} = \overline{\mathbf{U}}_s * (G\mathbf{a}), \tag{4}$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^s$, and $\overline{\mathbf{U}}_s$ is defined in (2).

3. A global transpose-free quasi-minimal residual method

It is known that the CGS method [19] for the general linear systems may exhibit erratic convergence behavior. Unfortunately, the disadvantage can also be carried over to the Gl-CGS method [20]. In this section, we give a *global transpose-free quasi-minimal residual method* (Gl-TFQMR) for the Sylvester equation (1). Like the Gl-CGS method, Gl-TFQMR involves no matrix multiplication with the transpose of the coefficient matrices A and B, but often yields much smoother convergence behavior.

In the Gl-CGS method, the iterates X_s can be updated in two half-steps, namely, $X_{s+1/2} = X_s + \alpha_s W_s$ and $X_{s+1} = X_{s+1/2} + \alpha_s Q_s$. For convenience, we double the subscripts to avoid indices that are multiple of 1/2. It follows that $X_{2s+2} = X_{2s} + \alpha_{2s}(W_{2s} + Q_{2s})$. Correspondingly, the two half-steps are now updated by

$$X_{2s+1} = X_{2s} + \alpha_{2s} W_{2s} \text{ and } X_{2s+2} = X_{2s+1} + \alpha_{2s} Q_{2s}. \tag{5}$$

With an approach analogous to [7, 17], we define W_s and α_s for odd s by $W_{2j+1} = Q_{2j}$ and $\alpha_{2j+1} = \alpha_{2j}$. This results in a unified form of (5) for either even or odd values of s, i.e., $X_s = X_{s-1} + \alpha_{s-1} W_{s-1}$. Let $y_s = [\alpha_0, \alpha_1, \cdots, \alpha_{s-1}]^T \in \mathbb{R}^s$, and $\overline{\mathbf{W}}_s = [W_0, W_1, \cdots, W_{s-1}] \in \mathbb{R}^{m\times ns}$, then the approximate solutions X_s can be updated by

$$X_s = X_{s-1} + \alpha_{s-1} W_{s-1} = X_0 + \overline{\mathbf{W}}_s * y_s. \tag{6}$$

It should be noted that the iterates X_s above for odd s do not exist in the Gl-CGS algorithm. With (1) and (6), the residual matrices are given by

$$\begin{aligned} R_s &= C - \mathcal{A}(X_s) = C - \mathcal{A}(X_0 + \overline{\mathbf{W}}_s * y_s) = R_0 - \mathcal{A}(\overline{\mathbf{W}}_s * y_s) \\ &= R_{s-1} - \alpha_{s-1}\mathcal{A}(W_{s-1}). \end{aligned} \tag{7}$$

The relation (7) yields $\mathcal{A}(W_{s-1}) = \alpha_{s-1}^{-1}(R_{s-1} - R_s)$, where $s = 0, 1, \cdots$. It can be interpreted in a compact form

$$\begin{aligned} & [\mathcal{A}(W_0), \mathcal{A}(W_1), \cdots, \mathcal{A}(W_{s-1})] \\ = \; & [\alpha_0^{-1}(R_0 - R_1), \alpha_1^{-1}(R_1 - R_2), \cdots, \alpha_{s-1}^{-1}(R_{s-1} - R_s)] \\ = \; & [R_0, R_1, \cdots, R_{s-1}, R_s] * \mathbf{F}_s \equiv \overline{\mathbf{R}}_{s+1} * \mathbf{F}_s, \end{aligned} \tag{8}$$

where $\overline{\mathbf{R}}_{s+1} \in \mathbb{R}^{m\times(s+1)n}$, and the matrix $\mathbf{F}_s \in \mathbb{R}^{(s+1)\times s}$ has the form

$$\mathbf{F}_s = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -1 & 1 & & & \vdots \\ & -1 & 1 & \cdots & \\ & & \ddots & \ddots & \vdots \\ & & & -1 & 1 \\ & & & & -1 \end{pmatrix} \cdot \begin{pmatrix} \alpha_0^{-1} & & & \\ & \alpha_1^{-1} & & \\ & & \ddots & \\ & & & \alpha_{s-1}^{-1} \end{pmatrix}. \tag{9}$$

It follows from (7), (8) and (4) that

$$\begin{aligned} R_s &= R_0 - \mathcal{A}(\overline{\mathbf{W}}_s * y_s) = R_0 - \mathcal{A}(\sum_{i=0}^{s-1} \alpha_i W_i) = R_0 - \sum_{i=0}^{s-1} \alpha_i \mathcal{A}(W_i) \\ &= R_0 - [\mathcal{A}(W_0), \mathcal{A}(W_1), \cdots, \mathcal{A}(W_{s-1})] * y_s \\ &= R_0 - (\overline{\mathbf{R}}_{s+1} * \mathbf{F}_s) * y_s = R_0 - \overline{\mathbf{R}}_{s+1} * (\mathbf{F}_s y_s) \\ &= \overline{\mathbf{R}}_{s+1} * e_1^{(s+1)} - \overline{\mathbf{R}}_{s+1} * (\mathbf{F}_s y_s) = \overline{\mathbf{R}}_{s+1} * (e_1^{(s+1)} - \mathbf{F}_s y_s), \end{aligned} \tag{10}$$

where $e_1^{(s+1)}$ stands for the first canonical basis vector in $\mathbb{R}^{s+1}$.

Obviously, if $\overline{\mathbf{R}}_{s+1}$ in (10) is F-orthogonal, then to minimize $\|R_s\|_F$ is equivalent to minimize $\|e_1^{(s+1)} - \mathbf{F}_s y_s\|_2$, as done in [12]. Generally, however, $\overline{\mathbf{R}}_{s+1}$ can be dense and not F-orthogonal. A compromise, therefore, is to minimize $\|e_1^{(s+1)} - \mathbf{F}_s y_s\|_2$, which turns to be a least squares problem. To improve the conditioning of $\overline{\mathbf{R}}_{s+1}$, a scaling matrix can be chosen such that the F-norm of each R_i in $\overline{\mathbf{R}}_{s+1}$, say, will be reduced to one. Such scaling can be achieved through defining an $s+1$ by $s+1$ diagonal matrix $\mathbf{\Sigma}_{s+1} \equiv \mathrm{diag}[\sigma_0, \sigma_1, \cdots, \sigma_s]$ with $\sigma_i = \|R_i\|_F$. By combining (3), (4) and (10), it follows that

$$\begin{aligned} R_s &= \overline{\mathbf{R}}_{s+1} * e_1^{(s+1)} - \overline{\mathbf{R}}_{s+1} * (\mathbf{F}_s y_s) \\ &= (\overline{\mathbf{R}}_{s+1} * \mathbf{\Sigma}_{s+1}^{-1}) * (\sigma_0 e_1^{(s+1)}) - (\overline{\mathbf{R}}_{s+1} * \mathbf{\Sigma}_{s+1}^{-1}) * (\mathbf{\Sigma}_{s+1} \mathbf{F}_s y_s) \\ &\equiv (\overline{\mathbf{R}}_{s+1} * \mathbf{\Sigma}_{s+1}^{-1}) * (\mathbf{g}_{s+1} - \overline{\mathbf{H}}_s y_s), \end{aligned} \tag{11}$$

where $\mathbf{g}_{s+1} \equiv \sigma_0 e_1^{(s+1)}$ and $\overline{\mathbf{H}}_s \equiv \mathbf{\Sigma}_{s+1} \mathbf{F}_s \in \mathbb{R}^{(s+1)\times s}$. Based on the above discussion, therefore, we have to solve the following least squares problem

$$\tau_s = \|\mathbf{g}_{s+1} - \overline{\mathbf{H}}_s y_s\|_2 = \min_{y\in\mathbb{R}^s} \|\mathbf{g}_{s+1} - \overline{\mathbf{H}}_s y\|_2. \tag{12}$$

From (9) and definitions of $\mathbf{\Sigma}_{s+1}$ and $\mathbf{F}_s$, we conclude that the matrix $\overline{\mathbf{H}}_s$ has full rank and thus y_s is uniquely defined in (12). To contrive the implementation of Gl-TFQMR method, we need the auxiliary iterates

$$\widetilde{X}_s = X_0 + \overline{\mathbf{W}}_s * \widetilde{y}_s, \tag{13}$$

where $\widetilde{y}_s = \mathbf{H}_s^{-1} \mathbf{g}_s$, the tildes are hereafter used to denote the Gl-CGS iterates, and $\mathbf{H}_s \in \mathbb{R}^{s\times s}$ is obtained by deleting the last row of $\overline{\mathbf{H}}_s$. With the definition of $\mathbf{H}_s$ in (11), it holds that $\sigma_s = \|\mathbf{g}_{s+1} - \overline{\mathbf{H}}_s \widetilde{y}_s\|_2$; see [7, p.474-475] for a simple derivation of this relation.

Next we proceed to devise some recurrence relations in the Gl-TFQMR method. Firstly, we introduce the following Lemma.

Lemma 1. Let the $s+1$ by s upper Hessenberg matrix

$$\overline{\mathbf{H}}_s = \begin{pmatrix} \mathbf{H}_s \\ h_{s+1,s} e_s^T \end{pmatrix} = \begin{pmatrix} \overline{\mathbf{H}}_{s-1} & \star \\ 0 & h_{s+1,s} \end{pmatrix},$$

be of full column rank s, where $e_s = [0, 0, \cdots, 0, 1]^T$, $\sigma_0 > 0$ and $s \geq 1$. For $k = s-1, s$, assume that $y_k \in \mathbb{R}^k$ is the solution of the least squares problem

$$\tau_k \equiv \min_{y \in \mathbb{R}^k} \|\mathbf{g}_{k+1} - \overline{\mathbf{H}}_k y\|_2, \text{ where } \mathbf{g}_{k+1} = \sigma_0 e_1^{(k+1)} \in \mathbb{R}^{k+1}. \tag{14}$$

Suppose that $\mathbf{H}_s$ is nonsingular, and set $\widetilde{y}_s \equiv \mathbf{H}_s^{-1}\mathbf{g}_s$. Then we get

$$y_s = (1 - c_s^2)\begin{pmatrix} y_{s-1} \\ 0 \end{pmatrix} + c_s^2 \widetilde{y}_s, \quad \tau_s = \tau_{s-1}\theta_s c_s, \tag{15}$$

where $\theta_s = \tau_{s-1}^{-1}\|\mathbf{g}_{s+1} - \overline{\mathbf{H}}_s \widetilde{y}_s\|_2 = \sigma_s \tau_{s-1}^{-1}$ and $c_s = (1 + \theta_s^2)^{-1/2}$.

Proof. The proof is similar to that of Lemma 4.1 in [7], so it is omitted.□

By using (6), (13) and the first equation of (15), we obtain that

$$\begin{aligned} X_s &= X_0 + \overline{\mathbf{W}}_s * [(1 - c_s^2)\begin{pmatrix} y_{s-1} \\ 0 \end{pmatrix} + c_s^2 \widetilde{y}_s] \\ &= X_0 + (1 - c_s^2)\overline{\mathbf{W}}_{s-1} * y_{s-1} + c_s^2 \overline{\mathbf{W}}_s * \widetilde{y}_s \\ &= X_0 + \overline{\mathbf{W}}_{s-1} * y_{s-1} + c_s^2(\widetilde{X}_s - X_0 - \overline{\mathbf{W}}_{s-1} * y_{s-1}) \\ &= (1 - c_s^2)X_{s-1} + c_s^2 \widetilde{X}_s. \end{aligned} \tag{16}$$

From (16) it follows that $\widetilde{X}_s - X_{s-1} = c_s^{-2}(X_s - X_{s-1})$. Let $D_s = \alpha_{s-1}^{-1}(\widetilde{X}_s - X_{s-1}) = c_s^{-2}\alpha_{s-1}^{-1}(X_s - X_{s-1}) = \eta_s^{-1}(X_s - X_{s-1})$, where $\eta_s \equiv c_s^2 \alpha_{s-1}$. Thus $X_s = X_{s-1} + \eta_s D_s$. It can be deduced from (13) that

$$\widetilde{X}_s = \widetilde{X}_{s-1} + \alpha_{s-1} W_{s-1}. \tag{17}$$

We now try to carry out an iterative formula for D_s. By (16), (17), one obtains that

$$\begin{aligned} D_s &= \alpha_{s-1}^{-1}(\widetilde{X}_s - X_{s-1}) = \alpha_{s-1}^{-1}(\widetilde{X}_s - \widetilde{X}_{s-1} + \widetilde{X}_{s-1} - X_{s-1}) \\ &= W_{s-1} + \alpha_{s-1}^{-1}(\widetilde{X}_{s-1} - X_{s-2} + X_{s-2} - X_{s-1}) \\ &= W_{s-1} + \alpha_{s-1}^{-1}(1 - c_{s-1}^2)(\widetilde{X}_{s-1} - X_{s-2}). \end{aligned} \tag{18}$$

Therefore, from Lemma 1, the recurrence for D_s is expressed by

$$D_s = W_{s-1} + \alpha_{s-1}^{-1}\theta_{s-1}^2 \eta_{s-1} D_{s-1}, \tag{19}$$

where $c_s = (1 + \theta_s^2)^{-1/2}$, as defined in (15).

With (15), (17) and (19), we have generated the relations for D_s, θ_s, c_s, τ_s and η_s respectively. Some other relations, based on Gl-CGS, must be carried out to realize the actual implementation of Gl-TFQMR. It should be noted that the matrices R_s are no longer the actual residuals in the algorithm, we hereby replace it by L_s. From (7), it follows that

$$L_s = L_{s-1} - \alpha_{s-1}\mathcal{A}(W_{s-1}).$$

Here we use $\mathcal{A}(W_i)$ to update the matrices $V_{2j} = \mathcal{A}(P_{2j})$. Imposing the linear operator $\mathcal{A}$ on both sides of the equation in line 9 of Algorithm 1, we have

$$\mathcal{A}(P_{2j}) = \mathcal{A}(W_{2j}) + \beta_{2j-2}[\mathcal{A}(Q_{2j-2}) + \beta_{2j-2}\mathcal{A}(P_{2j-2})].$$

Therefore, it holds that $V_{2j} = \mathcal{A}(W_{2j}) + \beta_{2j-2}[\mathcal{A}(W_{2j-1}) + \beta_{2j-2}V_{2j-2}]$ by assuming $Q_{2j} = W_{2j+1}$. Furthermore, we have the following relations

$$W_{s+1} = W_s - \alpha_s V_s \text{ for even } s, \quad W_{s+1} = L_{s+1} + \beta_{s-1} W_s \text{ for odd } s.$$

Based on the above process, the Gl-TFQMR method for solving the Sylvester equation (1) can be presented in Algorithm 2.

Algorithm 2 Global transpose-free quasi-minimal method (Gl-TFQMR).

Choose X_0, set $L_0 = W_0 = R_0 = C - \mathcal{A}(X_0)$, $V_0 = \mathcal{A}(W_0)$, $D_0 = \mathbf{0}$, $\theta_0 = \eta_0 = 0$, and $\tau_0 = \|R_0\|_F$. Choose $\widetilde{R}_0$ such that $\rho_0 =< \widetilde{R}_0, R_0 >_F \neq 0$.
for $s = 0, 1, \cdots$ until convergence **do**
  if s is even **then**
    $\alpha_{s+1} = \alpha_s = \rho_s / < V_s, \widetilde{R}_0 >_F$;
    $W_{s+1} = W_s - \alpha_s V_s$;
  end if
  $L_{s+1} = L_s - \alpha_s \mathcal{A}(W_s)$;
  $D_{s+1} = W_s + \alpha_s^{-1}\theta_s^2 \eta_s D_s$;
  $\theta_{s+1} = \tau_s^{-1}\|L_{s+1}\|_F$, $c_{s+1} = (1 + \theta_{s+1}^2)^{-1/2}$;
  $\tau_{s+1} = \tau_s \theta_{s+1} c_{s+1}$, $\eta_{s+1} = \alpha_s c_{s+1}^2$;
  $X_{s+1} = X_s + \eta_{s+1} D_{s+1}$;
  if s is odd **then**
    $\rho_{s+1} =< L_{s+1}, \widetilde{R}_0 >_F$, $\beta_{s-1} = \rho_{s+1}/\rho_{s-1}$;
    $W_{s+1} = L_{s+1} + \beta_{s-1} W_s$;
    $V_{s+1} = \mathcal{A}(W_{s+1}) + \beta_{s-1}[\mathcal{A}(W_s) + \beta_{s-1} V_{s-1}]$;
  end if
end for

Table 1 Iterations and CPU time (in sec.) for Example 1.

m	Iterations (time)			
	Gl-GMRES(10)	Gl-GMRES(20)	Gl-GMRES(50)	Gl-TFQMR
200	30(0.4780)	21(0.5404)	21(0.5961)	14(0.2949)
400	57(7.537)	37(7.709)	29(8.646)	24(4.382)
600	71(28.496)	55(33.714)	37(39.896)	28(15.457)
800	95(81.856)	69(91.688)	44(119.635)	37(43.584)
1000	119(309.548)	82(283.364)	57(370.902)	44(109.843)

A few comments are in order. If $B = \mathbf{0}$ in the Sylvester equation (1), then Algorithm 2 can be adopted to solve the linear systems with multiple right-hand sides, namely, $AX = C$; furthermore, if $n = 1$, then Algorithm 2 reduces to the classical TFQMR method for solving the general linear system [7, 17]. Like the Gl-CGS method, the Gl-TFQMR method presented here can break down. Although such breakdowns are very rare in practice, it is crucial to incorporate look-ahead and composite steps techniques which are not pursued in this paper; see, for instance, [17] for more about the look-ahead technique. It should also be noted that in Algorithm 2 there is no explicit formula for the residual matrices R_s (corresponding to X_s), the F-norm of which are often used to check the convergence of the algorithm. However, a result at no extra cost is available, namely,

$$\|R_s\|_F \leq \|\overline{\mathbf{R}}_{s+1} * \mathbf{\Sigma}_{s+1}^{-1}\|_F \|\mathbf{g}_{s+1} - \overline{\mathbf{H}}_s y_s\|_2 = \tau_s (s+1)^{1/2}, \tag{20}$$

where the relations (11), (14) and the inequality $\|(\overline{\mathbf{R}}_{s+1} * \mathbf{\Sigma}_{s+1}^{-1}) * (\mathbf{g}_{s+1} - \overline{\mathbf{H}}_s y_s)\|_F \leq \|\overline{\mathbf{R}}_{s+1} * \mathbf{\Sigma}_{s+1}^{-1}\|_F \|\mathbf{g}_{s+1} - \overline{\mathbf{H}}_s y_s\|_2$ have been used. Therefore, we can examine the convergence of Algorithm 2 by using $\tau_s(s+1)^{1/2}$, the upper bound of $\|R_s\|_F$ in (20), instead of computing the actual values of $\|R_s\|_F$.

4. Numerical Examples

In this section, we give some numerical examples to demonstrate the effectiveness of Gl-TFQMR method. Some recently proposed global methods, such as Gl-GMRES [12], Gl-CGS, G-GlCGS2 [20] and Gl-CRS [21],

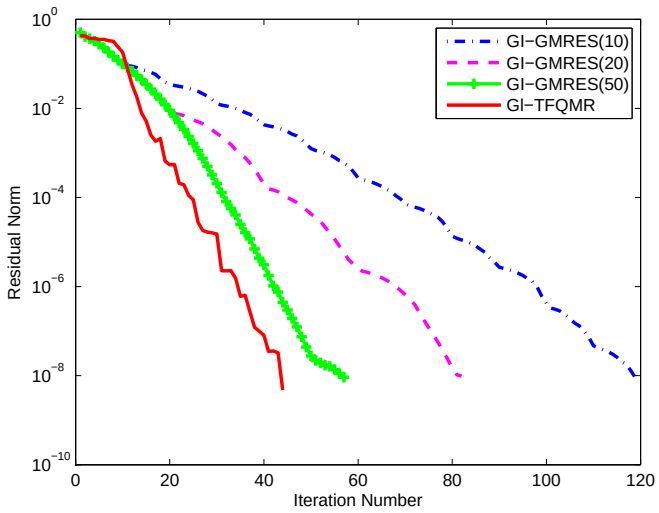

Fig. 1. Convergence histories of Gl-TFQMR and Gl-GMRES(k) with $m = 1000$, $k = 10, 20$, and 50.

are used for comparison. Though originally proposed for linear systems with multiple right-hand sides, Gl-CGS, G-GlCGS2 and Gl-CRS, as stated in [20, 21], belong to the global methods and can be applied to the Sylvester equation (1). All examples are terminated as soon as $\|R_k\|_F/\|R_0\|_F \leq 1e-8$, where R_k and R_0 denote respectively the true residuals at the kth and the initial step. To report the true relative residual norms, we compute the kth residual for Gl-TFQMR by $R_k = C - AX_k - X_kB$ instead of using (20). The maximum iterative step is set to be 500 and the initial matrix X_0 is chosen to be zero. All computations are carried out using Matlab 7.6.0.

Example 1. In this example, we compare Gl-TFQMR with Gl-GMRES(k), where k denotes the integer number of iterations between restarts. The matrices in the Sylvester equation (1) are given by the Matlab code [6]:

- A = triu(rand(m, m), 1) + diag(10 + diag(rand(m)))';
- B = A'; C = rand(m,m)+eye(m)*2; C = C+C';

The function *rand* in the above code returns a random matrix with coefficients uniformly distributed on the unit interval. We consider different values of m in this example. For this example, as seen from Table 1 and Fig. 1, Gl-TFQMR outperforms Gl-GMRES(k) with less iterations and CPU time. For small values of m, say $m = 200$, Gl-TFQMR shows better performance by a small margin. However, memory and computational requirements grow as m increases for Gl-GMRES. Yet Gl-TFQMR can reach the required accuracy with relatively less iterations and time; see Table 1 for details.

Example 2. In this example, we compare the performances of Gl-TFQMR with Gl-CGS, G-GlCGS2 and Gl-CRS. Three groups of matrices taken from the Matrix Market collection [15] are discussed: A_1=utm1700a, B_1=fs7601; A_2=rdb200, B_2=A_2; A_3=watt2, B_3=*rand*(35), C_i are generated by *rand* with conforming dimension for $i = 1, 2$, and C_3=**1**, where **1** is the matrix with all entries 1.

The orders, the numbers of nonzero elements and the application disciplines of these test matrices are listed in Table 2. From Fig.2-Fig.4 and Table 3, we have the following observations for this example. (a) Gl-CGS usually converges with less CPU time but its convergence curves are rather erratic. Such oscillating behavior may possibly destroy the convergence, which is shown in Fig.4 (from step 350 to step 431). On the contrary, Gl-TFQMR shows much smoother convergence curves than Gl-CGS. Peaks that appear in the curves of Gl-CGS vanish in those of Gl-TFQMR while the original convergence property of Gl-CGS is retained in Gl-TFQMR, which validates that Gl-TFQMR can be a remedy for the erratic behavior of Gl-CGS. (b) Except the CPU time, Gl-TFQMR, G-GlCGS2 and Gl-CRS outperform Gl-CGS regarding iteration number. (c) In terms of convergence curves, iteration steps and time, Gl-CRS shows moderate performance

Table 2 Test matrices: the order (m) and the number of nonzero elements (nnz).

Matrix	m	nnz	Application discipline
fs7601	760	5739	Chemical kinetics
rdb200	200	1120	Chemical engineering
utm1700a	1700	21313	Nuclear physics (plasmas)
watt2	1856	11550	Petroleum engineering

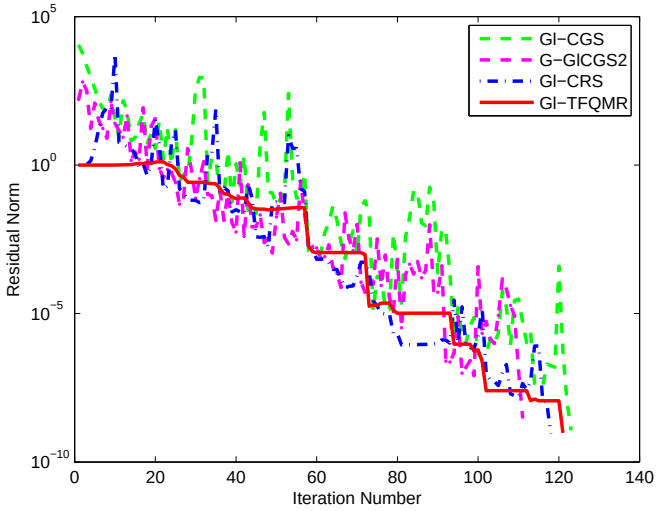

Fig.2. Convergence histories for A_1.

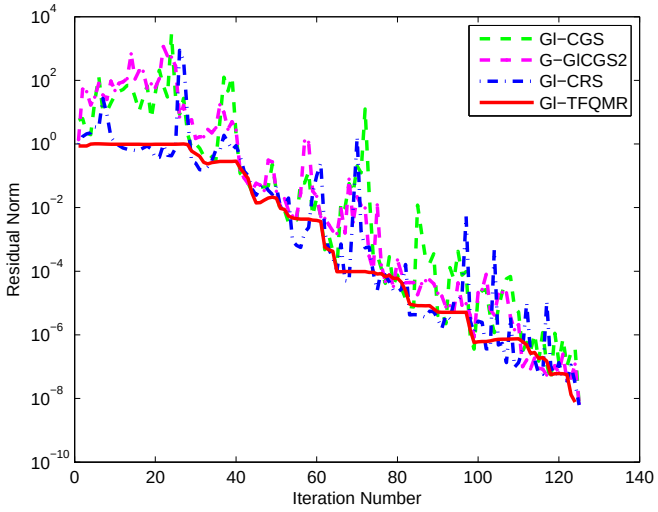

Fig.3. Convergence histories for A_2.

in this example. As for the comparison between Gl-TFQMR and G-GlCGS2, Gl-TFQMR presents smoother convergence with less CPU time than G-GlCGS2 in all three testing groups, while G-GlCGS2 converges with less iteration steps than Gl-TFQMR in all but the second testing group. The final true relative residual norms (TRR) are also reported in Table 3; see Table 3 for details.

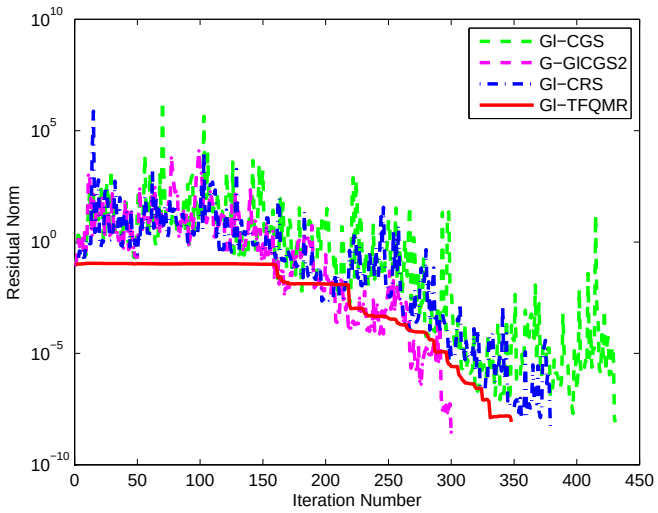

Fig.4. Convergence histories for A_3.

Table 3 Iterations, CPU time and TRR for Example 2.

Matrix	Iter/Time/TRR			
	Gl-CGS	G-GlCGS2	Gl-CRS	Gl-TFQMR
A_1	123/134.701/1.2$e-9$	111/219.716/3.2$e-9$	118/147.270/9.1$e-10$	121/119.370/9.6$e-10$
A_2	125/1.581/9.5$e-9$	125/2.8/9.5$e-9$	125/2.230/9.2$e-9$	124/2.005/9.9$e-9$
A_3	431/4.835/5.8$e-9$	300/6.781/5.5$e-9$	379/7.780/5.7$e-9$	348/6.556/5.9$e-9$

Table 4 Iterations, CPU time and TRR for Example 3.

(m,n)	Iter/Time/TRR		
	G-GlCGS2	Gl-CRS	Gl-TFQMR
(1000, 50)	24/0.523/9.3$e-9$	21/0.341/4.4$e-9$	21/0.260/9.8 − 9
(1000, 500)	61/45.152/3.6$e-9$	55/24.259/9.9$e-9$	57/20.509/8.6$e-9$
(1000, 700)	69/88.022/6.4$e-9$	60/49.152/9.8$e-9$	63/35.617/9.7$e-9$
(2000, 50)	23/1.018/3.0$e-9$	21/0.681/5.7$e-9$	21/0.933/7.9$e-9$
(2000, 500)	68/96.394/5.5$e-9$	60/50.393/7.5$e-9$	62/42.466/8.5$e-9$
(2000, 700)	75/187.092/7.6$e-9$	68/95.416/7.8$e-9$	71/76.473/8.9$e-9$
(5000, 50)	23/2.748/2.2$e-9$	21/2.054/6.7$e-9$	21/2.024/8.9$e-9$
(5000, 500)	71/245.272/6.9$e-9$	63/127.852/9.8$e-9$	66/112.924/8.8$e-9$
(5000, 700)	82/508.868/7.8$e-9$	73/255.861/9.8$e-9$	77/209.703/9.6$e-9$

Example 3. In this example, we consider the Sylvester eqaution (1) with A = tridiag$(-1 + 10/(m + 1), 2, -1 + 10/(m + 1))$, B = tridiag$(-1 + 10/(n + 1), 2, -1 + 10/(n + 1))$, and C= *rand*(m, n).

The numerical results are reported in Table 4 for different m and n. An overall conclusion observed from Table 4 is that Gl-CRS and Gl-TFQMR show better performance than G-GlCGS2 in both iterations and time. As for Gl-CRS and Gl-TFQMR, they perform similarly for small values n, for example, $n = 50$. Nevertheless, Gl-TFQMR converges to the required accuracy, though with some additional marginal iterations, using less CPU time than Gl-CRS as n increases.

5. Conclusions

In this paper, we present a global transpose-free method with quasi-minimal residual technique for solving the Sylvester equation. The resulting method can be attractive for the reason that it is transpose free, with short-term recurrence and often has much smoother convergence than some other existing global methods. Numerical results illustrate that the presented method not only has the smooth convergence but preserves the fast convergence of the Gl-CGS method. The future work may include the theoretical analysis of the convergence of Gl-TFQMR.

Acknowledgments. We are deeply indebted to Professor Dragana Cvetkovic Ilic and the referee for their insightful comments which have helped us improve a previous version of this paper.

References

[1] Z. Bai, On Hermitian and skew-Hermitian splitting iteration methods for continuous Sylvester eqautions, J. Comput. Math., 29(2) (2011) 185–198.
[2] R. Bartels and G. Stewart, Solution of the matrix equation $AX + XB = C$: Algorithm 432, Commun. ACM, 15 (1972) 820–826.
[3] P. Benner, R. Li and N. Truhar, On the ADI method for Sylvester eqautions, J. Comput. Appl. Math., 233 (2009) 1035–1045.
[4] B. Datta, Numerical methods for linear control systems, Academic Press, 2003.
[5] B. Datta, M. Heyouni and K. Jbilou, The global Arnoldi process for solving the Sylvester-observer equation, Comput. App. Math., 29(3) (2010) 527–544.
[6] F. Ding and T. Chen, Gradient based iterative algorithms for solving a class of matrix equations, IEEE Trans. Autom. Control, 50(8) (2005) 1216–1221.
[7] R. W. Freund, A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems, SIAM J. Sci. Comput., 14(2) (1993) 470–482.
[8] F. R. Gantmacher, The Theory of Matrices, Chelsea, New York, 1959.
[9] G. H. Golub, S. Nash and C. F. Van Loan, A Hessenberg-Schur method for the matrix problem $AX + XB = C$, IEEE Trans. Autom. Control, 24(6) (1979) 909–913.
[10] C. Gu and H. Xue, A shift-splitting hierarchical identification method for solving Lyapunov matrix equations, Linear Alg. Appl., 430(5-6) (2009) 1517–1530.
[11] S. Gugercin, D. C. Sorensen and A. C. Antoulas, A modified low-rank smith method for large-scale Lyapunov equations, Numer. Algorithms, 32(1) (2003) 27–55.
[12] K. Jbilou, A. Messaoudi and H. Sadok, Global FOM and GMRES algoritms for matrix equations, Appl. Numer. Math., 31 (1999) 49–63.
[13] L. Jódar, An algorithm for solving generalized algebraic Lyapunov equations in Hilbert space, applications to boundary value problems, Proc. Edinburgh Math. Society, 31 (1988) 99–105.
[14] J.-R. Li and J. White, Low-rank solution of Lyapunov eqautions, SIAM J. Matrix Anal. Appl., 24(2002) 260–280.
[15] Matrix Market, http://math.nist.gov/MatrixMarket/.
[16] T. Penzl, A cyclic low-rank smith method for large sparse Lyapunov equations, SIAM J. Sci. Comput., 21(2000) 1401–1418.
[17] Y. Saad, Iterative methods for sparse linear systems, (2nd ed.), SIAM, Philadelphia, 2003.
[18] R.A. Smith, Matrix equation $XA + BX = C$, SIAM J. Appl. Math., 16 (1968) 198–201.
[19] P. Sonneveld, CGS: a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Statist. Comput., 10 (1989) 36–52.
[20] J. Zhang, H. Dai and Jing Zhao, Generalized global conjugate gradient squared algorithm, Appl. Math. Comput., 216 (2010) 3694–3706.
[21] J. Zhang, H. Dai and Jing Zhao, A new family of global methods for linear systems with multiple right-hand sides, J. Comput. Appl. Math., 236 (2011) 1562–1575.