Journal Title
XX(X):1–18

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

A new generalization of the HSS method for solving the continuous Sylvester equation

Davod Khojasteh Salkuyeh[1] and Mehdi Bastani[2]

Abstract

The two-parameter generalized Hermitian and skew-Hermitian splitting (TGHSS) iteration method is applied to solve the continuous Sylvester equation $AX + XB = C$, where $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$ and $C \in \mathbb{C}^{m\times n}$. The matrix-vector form of the equation is considered to present convergence properties of the TGHSS method. Some numerical examples are given to show the effectiveness and accuracy of the method. Comparison with other methods are also included.

Keywords

Continuous Sylvester equation, HSS method, GHSS method, TGHSS method, convergence

Introduction

In this paper, an iterative method is given to compute an approximate solution of the continuous Sylvester equation of the form

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

where $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$ and $C \in \mathbb{C}^{m\times n}$ are given complex matrices. Assume that

(A_1) A, B and C are large and sparse matrices;

(A_2) at least one of the matrices A and B is non-Hermitian;

(A_3) both A and B are positive semidefinite, and at least one of them is positive definite.

A necessary and sufficient condition for the continuous Sylvester equation (1) to have a unique solution, as under the assumptions (A_1)-(A_3), is that $\lambda(A) + \lambda(B) \neq 0$ for all $\lambda(A) \in \sigma(A)$ and $\lambda(B) \in \sigma(B)$, where $\sigma(M)$ denotes the spectrum of the matrix M (Gantmacher, 1959; Lancaster and Tismenetsky, 1985).

A special case of the continuous Sylvester equation is achieved for $B = A^T$ and $C = C^T$, and called continuous Lyapunov equation. The continuous Sylvester equation (1) and the proposed special case play an important role in many fields and have numerous applications in the control theory (Schaft, 2000), model reduction (Gallivan, 2004; Sorensen and Antoulas, 2002), signal processing (Anderson et al., 1986), image restoration (Sanches et al., 2008), analysis of bilinear systems (Lewis et al., 1990), stability of linear systems (Halanay and Răsvan, 1993), numerical methods for differential equations (Bai et al., 2006a), power systems (Ilic, 1989) and so on.

There is an equivalent form for the continuous Sylvester equation (1) of the form

$$\mathcal{A}x = c, \tag{2}$$

[1]Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran
[2]Department of Mathematics, Faculty of Mathematical Sciences, University of Mohaghegh Ardabili, Ardabil, Iran

Corresponding author:
Faculty of Mathematical Sciences, University of Guilan, P.O. Box 1914, Rasht, Iran.
Email: khojasteh@guilan.ac.ir

where $\mathcal{A} = I \otimes A + B^T \otimes I$, $x = \mathbf{vec}(X)$ and $c = \mathbf{vec}(C)$, with $\otimes$ being the Kronecker product symbol and B^T representing the transpose of the matrix B. It is noted that, for a matrix $X \in \mathbb{C}^{m\times n}$, the vector form of the matrix X is defined as $\mathbf{vec}(X) = (X_{*1}; X_{*2}; \ldots; X_{*n})$, where X_{*j} is the jth column of X. Note that the equivalent system (2) is costly to solve and can be ill-conditioned. So, solving (1) directly by an iterative method is preferred on solving (2).

In the literature, several papers have been presented to numerically solve equation (1). Bartels and Stewart (1972) have reduced the matrix A and B in (1) to lower and upper Schur form, respectively, to solve the proposed equation. The preconditioned conjugate gradient (PCG) method has been presented to solve continuous Sylvester equation by Evans and Galligani (1994). The proposed matrix equation has been solved by an iterative least-squares method in Ding and Chen (2005). Salkuyeh and Toutounian (2006) have applied modified global Arnoldi algorithm for solving large Sylvester matrix equations. A generalization of Cholesky factor alternating directional implicit (ADI) method has been presented by Benner et al. (2009). Beckermann (2011) has applied rational Galerkin projection method to solve the proposed matrix equation. A relaxation parameter has been introduced by Niu et al. (2011) to derive a relaxed gradient-based algorithm for solving Sylvester equations. Wang et al. (2012) have also applied a modified gradient based algorithm to solve (1). Hached (2015) has applied an improved version of the Davison-Man method to numerically solve the Sylvester matrix equations.

Bai et al. (2003) presented the Hermitian and skew-Hermitian splitting (HSS) iteration method to solve non-Hermitian positive definite linear systems. The proposed method is based on splitting the coefficient matrix A into Hermitian matrix $H = (A + A^H)/2$ and skew-Hermitian matrix $S = (A - A^H)/2$. The HSS iteration method has been widely used to solve non-Hermitian positive definite linear systems in various problems such as convection-diffusion problem (Bai and Yang, 2009; Bertaccini et al., 2005), saddle point problem (Bai and Golub, 2007; Bai et al., 2004; Benzi and Golub, 2004), image restoration problem (Aghazadeh et al., 2015a; Bai, 2009; Benzi and Ng, 2006) and so on. Bai (2011) applied the HSS method to solve continuous Sylvester equation with non-Hermitian and positive definite/semidefinite matrices. The positive-definite and skew-Hermitian splitting (PSS) iteration method has been established to solve positive-definite linear systems in Bai et al. (2005). Wang et al. (2013) used the PSS iteration method to approximate the solutions of the continuous Sylvester equation. Bai et al. (2007) and Bai and Ng (2012) generalized the HSS iteration method and constructed the normal and skew-Hermitian splitting (NSS) iteration method for solving large sparse non-Hermitian positive definite system of linear equations. The proposed method has been applied by Zheng and Ma (2014) to solve large sparse continuous Sylvester equation. Recently, Zhou et al. (2015) used the idea of generalized HSS (GHSS) method proposed by Benzi (2009) to solve the equation (1). In this paper, we use the idea of the two-parameter GHSS (TGHSS) method presented by Aghazadeh et al. (2015b) and apply it to solve the continuous Sylvester equation (1). Convergence properties of the proposed method is also presented.

Throughout the paper, $\|.\|$ stands for the Euclidean norm in $\mathbb{C}^n$, i.e. for $x \in \mathbb{C}^n$, $\|x\| = (x^H x)^{\frac{1}{2}}$. Similarly, for any matrix $A \in \mathbb{C}^{n\times n}$, $\|A\| = \max_{\|x\|=1} \|Ax\|$. The spectral radius and the spectrum of A are shown by $\rho(A)$ and $\sigma(A)$, respectively. Furthermore, for a matrix $X \in \mathbb{R}^{n\times n}$ whose eigenvalues are real, $\lambda_1(X)$ and $\lambda_n(X)$ stand for the largest and smallest eigenvalues of X, respectively. The conjugate transpose of $X \in \mathbb{C}^{n\times n}$ is denoted by X^*.

This paper is organized as follows. In Section 2, a brief description of the application of the HSS and GHSS methods is given to approximate the solutions of continuous Sylvester equation. TGHSS method is applied to solve continuous Sylvester equation in Section 3. Some convergence results are also presented in this section. In Section 4, four numerical tests are given to show the effectiveness and accuracy of the TGHSS method. Finally, some concluding remarks are presented in Section 5.

A brief description of HSS and GHSS methods

In this section, we briefly introduce the HSS and GHSS methods to solve continuous Sylvester equation (1).

It is well known that the matrices A and B admit the Hermitian and skew-Hermitian splittings

$$A = H(A) + S(A), \quad B = H(B) + S(B), \tag{3}$$

where $H(A) = (A + A^*)/2$, $S(A) = (A - A^*)/2$, $H(B) = (B + B^*)/2$ and $S(B) = (B - B^*)/2$. From equation (3), two following splittings can be written for any positive numbers α and β:

$$A = (\alpha I + H(A)) - (\alpha I - S(A)) = (\alpha I + S(A)) - (\alpha I - H(A)),$$

and

$$B = (\beta I + H(B)) - (\beta I - S(B)) = (\beta I + S(B)) - (\beta I - H(B)).$$

By using these splittings, Bai (2011) presented the following HSS iteration method to solve (1).

HSS iteration method:
Let $X^{(0)} \in \mathbb{C}^{m\times n}$ be an initial guess and α, β be two positive real numbers. $X^{(k+1)} \in \mathbb{C}^{m\times n}$ for $k = 0, 1, 2, \ldots$, is computed by using the following HSS iteration scheme until $\{X^{(k)}\}_{k=0}^{\infty}$ satisfies the stopping criterion:

$$\begin{cases} (\alpha I + H(A))X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\beta I + H(B)) = \\ \qquad\qquad (\alpha I - S(A))X^{(k)} + X^{(k)}(\beta I - S(B)) + C, \\ (\alpha I + S(A))X^{(k+1)} + X^{(k+1)}(\beta I + S(B)) = \\ \qquad\qquad (\alpha I - H(A))X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\beta I - H(B)) + C. \end{cases} \tag{4}$$

Bai (2011) has shown that if A and B are Hermitian positive semidefinite and either A or B is positive definite, then the iteration (4) is convergent for any positive numbers α and β. Benzi (2009) split the Hermitian part of the coefficient matrix into two Hermitian positive semidefinite matrices and presented the GHSS method to solve non-Hermitian positive definite linear systems. Then, the method has been used to solve equation (1) by Zhou et al. (2015). To this end, the Hemitian part of matrices A and B is decomposed as

$$H(A) = G(A) + K(A), \quad H(B) = G(B) + K(B), \tag{5}$$

where $G(A)$, $G(B)$, $K(A)$ and $K(B)$ are Hermitian positive semidefinite matrices. The GHSS iteration to solve the Sylvester equation (1) can be written as follows:

GHSS iteration method:
Let $X^{(0)} \in \mathbb{C}^{m\times n}$ be an initial guess and α, β be two positive numbers. $X^{(k+1)} \in \mathbb{C}^{m\times n}$ for $k = 0, 1, 2, \ldots$ is computed by using the following GHSS iteration scheme until $\{X^{(k)}\}_{k=0}^{\infty}$ satisfies the stopping criterion:

$$\begin{cases} (\alpha I + G(A))X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\beta I + G(B)) = \\ \qquad\qquad (\alpha I - S(A) - K(A))X^{(k)} + X^{(k)}(\beta I - S(B) - K(B)) + C, \\ (\alpha I + S(A) + K(A))X^{(k+1)} + X^{(k+1)}(\beta I + S(B) + K(B)) = \\ \qquad\qquad (\alpha I - G(A))X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\beta I - G(B)) + C. \end{cases} \tag{6}$$

Similar to the HSS method, convergence of the GHSS method can be investigated. Zhou et al. (2015) have shown that if $\mathcal{A} = \mathcal{G} + \mathcal{K} + \mathcal{S}$, where $\mathcal{G}$ and $\mathcal{K}$ are positive semidefinite Hermitian matrices, $\mathcal{S}$ is the skew-Hermitian matrix, and either $\mathcal{G}$ or $\mathcal{K}$ is positive definite, then the iteration (6) is convergent for any positive numbers α and β.

Two-parameter GHSS method

In Aghazadeh et al. (2015b), two-parameter GHSS (TGHSS) method has been introduced to solve non-Hermitian positive definite linear systems. Now, we use the idea of the proposed method to approximate the solutions of Sylvester equation (1).

Similar to the GHSS method, the Hermitian part of matrix A and B is split into Hermitian positive semidefinite matrices $G(A)$, $K(A)$, $G(B)$ and $G(K)$. Indeed, we have

$$\begin{aligned} A &= H(A) + S(A) = G(A) + K(A) + S(A), \\ B &= H(B) + S(B) = G(B) + K(B) + S(B). \end{aligned}$$

Then, two following splittings can be written for the matrix A and B,

$$\begin{aligned} A &= (\alpha_1 I + G(A)) - (\alpha_1 I - S(A) - K(A)), & A &= (\alpha_2 I + S(A) + K(A)) - (\alpha_2 I - G(A)), \\ B &= (\beta_1 I + G(B)) - (\beta_1 I - S(B) - K(B)), & B &= (\beta_2 I + S(B) + K(B)) - (\beta_2 I - G(B)), \end{aligned}$$

where α_1, α_2, β_1 and β_2 are positive numbers. Then, the continuous Sylvester equation (1) can be written as the following matrix equations:

$$\begin{cases} (\alpha_1 I + G(A))X + X(\beta_1 I + G(B)) = \\ \qquad\qquad (\alpha_1 I - S(A) - K(A))X + X(\beta_1 I - S(B) - K(B)) + C =: P(X), \\ (\alpha_2 I + S(A) + K(A))X + X(\beta_2 I + S(B) + K(B)) = \\ \qquad\qquad (\alpha_2 I - G(A))X + X(\beta_2 I - G(B)) + C =: Q(X). \end{cases} \tag{7}$$

From assumptions (A_1)-(A_3), it is easy to see that there is no common eigenvalue between the matrices $\alpha_1 I + G(A)$ and $\beta_1 I + G(B)$, as well as between the matrices $\alpha_2 I + K(A) + S(A)$ and $\beta_2 I + S(B) + K(B)$. Therefore, the proposed matrix equations in (7) have unique solutions for any given right-hand side matrices.

Now, from the above observations, the TGHSS method can be given to solve the continuous Sylvester equation (1) as follows:

TGHSS iteration method: Let $X^{(0)} \in \mathbb{C}^{m\times n}$ be an initial guess, $X^{(k+1)} \in \mathbb{C}^{m\times n}$ for $k = 0, 1, 2, \ldots$ is computed by using the following iteration scheme until $\{X^{(k)}\}_{k=0}^{\infty}$ satisfies the stopping criterion:

$$\begin{cases} (\alpha_1 I + G(A))X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\beta_1 I + G(B)) = P(X^{(k)}), \\ (\alpha_2 I + S(A) + K(A))X^{(k+1)} + X^{(k+1)}(\beta_2 I + S(B) + K(B)) = Q(X^{(k)}). \end{cases} \tag{8}$$

In continuation, convergence properties of the TGHSS method are investigated. To do so, we first recall the following lemma which has been introduced by Bai (2011); Zheng and Ma (2014)

Lemma 1. *Let $X^{(0)}$ be a given initial matrix, $A, B, C \in \mathbb{C}^{m\times n}$, and two splittings of the matrices A and B are given as $A_i = M_i(A) - N_i(A)$ and $B = M_i(B) - N_i(B)$ (i=1,2), respectively. If $X^{(k)}$ is a two-step iteration sequence defined by*

$$\begin{cases} M_1(A)X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}M_1(B) = N_1(A)X^{(k)} + X^{(k)}N_1(B) + C, \\ M_2(A)X^{(k+1)} + X^{(k+1)}M_2(B) = N_2(A)X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}N_2(B) + C, \end{cases} \tag{9}$$

$k = 0, 1, \ldots$, then

$$x^{(k+1)} = \mathcal{T}x^{(k)} + g, \tag{10}$$

where $x^{(k)} = \boldsymbol{vec}(X^{(k)})$, $x^{(k+1)} = \boldsymbol{vec}(X^{(k+1)})$,

$$\begin{aligned} \mathcal{T} \;=\; & [I \otimes M_2(A) + M_2(B)^T \otimes I]^{-1}[I \otimes N_2(A)^T + N_2(B)^T \otimes I] \\ & \times [I \otimes M_1(A) + M_1(B)^T \otimes I]^{-1}[I \otimes N_1(A)^T + N_1(B)^T \otimes I], \end{aligned}$$

and

$$\begin{aligned} g \;=\; & [I \otimes M_2(A) + M_2(B)^T \otimes I]^{-1} \\ & \times [I + [I \otimes N_2(A)^T + N_2(B)^T \otimes I][I \otimes M_1(A) + M_1(B)^T \otimes I]^{-1}]\boldsymbol{vec}(C). \end{aligned}$$

Moreover, if the spectral radius $\rho(\mathcal{T}) < 1$, then the iterative sequence $\{X^{(k)}\}_{k=0}^{\infty}$ converges to the unique solution $X^* \in \mathbb{C}^{m\times n}$ of equation (1) for all initial vectors $X^{(0)} \in \mathbb{C}^{m\times n}$.

Lemma 2. *(Lancaster and Tismenetsky, 1985) If the matrix $X \in \mathbb{C}^{m\times n}$ is the solution of the Sylvester equation (1), then $x = \boldsymbol{vec}(X)$ is the solution of Eq. (2).*

Aghazadeh et al. (2015b) have presented the following lemma which plays an essential role in proving of the convergence property of the TGHSS method. Similar versions of the next lemma have been presented by Johnson and Krukier (2009); Bai and Hadjidimos (2014).

Lemma 3. *Let $A \in \mathbb{C}^{n\times n}$ and $H = \frac{1}{2}(A + A^*)$ be positive definite. Let also α and β be two nonnegative numbers. Then*

(i) *If $0 \le \alpha \le \beta + \lambda_n/2$, then $\|(\alpha I - A)(\beta I + A)^{-1}\| \le 1$;*

(ii) *If $0 \le \alpha < \beta + \lambda_n/2$ such that α and β are not simultaneously equal to zero, then $\|(\alpha I - A)(\beta I + A)^{-1}\| < 1$, where λ_n is the smallest eigenvalue of H. Suppose that, in addition, H be positive semidefinite.*

(iii) *If $0 \le \alpha \le \beta$, then $\|(\alpha I - A)(\beta I + A)^{-1}\| \le 1$;*

(iv) *If $0 \le \alpha < \beta$, then $\|(\alpha I - A)(\beta I + A)^{-1}\| < 1$.*

In the next theorem, some sufficient conditions are given for the convergence of the TGHSS iteration method for solving the Sylvester matrix equation (1).

Theorem 4. *Let α_1, α_2, β_1 and β_2 be nonnegative numbers such that*

$$\gamma_1 = \alpha_1 + \beta_1 \neq 0, \qquad \gamma_2 = \alpha_2 + \beta_2 \neq 0.$$

Let also $\mathcal{A} = \mathcal{H} + \mathcal{S} = (\mathcal{G} + \mathcal{K}) + \mathcal{S}$, where $\mathcal{G}$ and $\mathcal{K}$ are Hermitian positive semidefinite matrices and $\mathcal{S}$ is a skew-Hermitian matrix with

$$\begin{aligned}\mathcal{A} &= I \otimes A + B^T \otimes I,\\ \mathcal{G} &= I \otimes G(A) + G(B)^T \otimes I,\\ \mathcal{K} &= I \otimes K(A) + K(B)^T \otimes I,\end{aligned}$$

and

$$\mathcal{S} = I \otimes S(A) + S(B)^T \otimes I.$$

Then, the TGHSS iteration (8) converges to the unique solution of the continuous Sylvester equation (1) if one of the following conditions holds:

(i) *$\mathcal{G}$ is positive definite, $\mathcal{K}$ is positive semidefinite matrix and*

$$\gamma_1 < \gamma_2 \le \gamma_1 + 2\lambda_n(\mathcal{G}) \quad or \quad \gamma_1 \le \gamma_2 < \gamma_1 + 2\lambda_n(\mathcal{G}).$$

(ii) *$\mathcal{G}$ is positive semidefinite, $\mathcal{K}$ is positive definite matrix and*

$$\gamma_2 \le \gamma_1 < \gamma_2 + \frac{1}{2}\lambda_n(\mathcal{K}) \quad or \quad \gamma_2 < \gamma_1 \le \gamma_2 + \frac{1}{2}\lambda_n(\mathcal{K}).$$

(iii) *$\mathcal{G}$ is positive definite, $\mathcal{K}$ is positive definite matrix and*

$$\begin{aligned}\gamma_1 &< \gamma_2 + \frac{1}{2}\lambda_n(\mathcal{K}) \le \gamma_1 + 2\lambda_n(\mathcal{G}) + \frac{1}{2}\lambda_n(\mathcal{K})\\ or \quad \gamma_1 &\le \gamma_2 + \frac{1}{2}\lambda_n(\mathcal{K}) < \gamma_1 + 2\lambda_n(\mathcal{G}) + \frac{1}{2}\lambda_n(\mathcal{K}).\end{aligned}$$

Proof. Let

$$\begin{aligned}&M_1(A) = \alpha_1 I + G(A), && N_1(A) = \alpha_1 I - K(A) - S(A),\\ &M_2(A) = \alpha_2 I + K(A) + S(A), && N_2(A) = \alpha_2 I - G(A),\\ &M_1(B) = \beta_1 I + G(B), && N_1(B) = \beta_1 I - K(B) - S(B),\\ &M_2(B) = \beta_2 I + K(B) + S(B), && N_2(B) = \beta_2 I - G(B).\end{aligned}$$

From Lemma 1, it can be seen that

$$x^{(k+1)} = \mathcal{T}_{\gamma_1,\gamma_2} x^{(k)} + g,$$

where

$$\mathcal{T}_{\gamma_1,\gamma_2} = (\gamma_2 I + \mathcal{K} + \mathcal{S})^{-1} (\gamma_2 I - \mathcal{G}) (\gamma_1 I + \mathcal{G})^{-1} (\gamma_1 I - \mathcal{K} - \mathcal{S})$$

and

$$g = (\gamma_2 I + \mathcal{K} + \mathcal{S})^{-1} \left(I + (\gamma_2 I - \mathcal{G}) (\gamma_1 I + \mathcal{G})^{-1} \right) \mathbf{vec}(c).$$

By similarity transformation, it can be seen that the eigenvalues distribution of $\mathcal{T}_{\gamma_1,\gamma_2}$ is the same as that of

$$\hat{\mathcal{T}}_{\gamma_1,\gamma_2} = (\gamma_2 I - \mathcal{G}) (\gamma_1 I + \mathcal{G})^{-1} (\gamma_1 I - \mathcal{S} - \mathcal{K}) (\gamma_2 I + \mathcal{S} + \mathcal{K})^{-1}. \tag{11}$$

Since $\hat{\mathcal{T}}_{\gamma_1,\gamma_2}$ is similar to $\mathcal{T}_{\gamma_1,\gamma_2}$, the eigenvalues distribution of two proposed matrices are the same. Therefore, in the sequel, the eigenvalues distribution of the matrix $\hat{\mathcal{T}}_{\gamma_1,\gamma_2}$ is considered to investigate the convergence of the TGHSS method. Now, let

$$\mathcal{P}_1 = (\gamma_2 I - \mathcal{G})(\gamma_1 I + \mathcal{G})^{-1}, \quad \mathcal{P}_2 = (\gamma_1 I - \mathcal{S} - \mathcal{K})(\gamma_2 I + \mathcal{S} + \mathcal{K})^{-1}. \tag{12}$$

From equations (11) and (12), it follows that

$$\rho(\hat{\mathcal{T}}_{\gamma_1,\gamma_2}) \leq \|\mathcal{P}_1\| \|\mathcal{P}_2\|.$$

We prove only case (i) and the two other cases can be proved similarly. Hence, suppose that $\mathcal{G}$ and $\mathcal{K}$ are positive definite and positive semidefinite matrices, respectively. Then,

$$\|\mathcal{P}_1\| = \|(\gamma_2 I - \mathcal{G})(\gamma_1 I + \mathcal{G})^{-1}\| = \max_{\lambda \in \sigma(\mathcal{G})} \frac{|\gamma_2 - \lambda|}{\gamma_1 + \lambda}.$$

Now there exists a $\hat{\gamma} > 0$ such that

$$\|\mathcal{P}_1\| = \begin{cases} \frac{\lambda_1(\mathcal{G}) - \gamma_2}{\lambda_1(\mathcal{G}) + \gamma_1}, & 0 < \gamma_2 \leq \hat{\gamma}, \\ \frac{\gamma_2 - \lambda_n(\mathcal{G})}{\lambda_n(\mathcal{G}) + \gamma_1}, & \hat{\gamma} \leq \gamma_2. \end{cases} \tag{13}$$

From the relation (13), it is easy to see that if

$$\gamma_2 \leq \gamma_1 + 2\lambda_n(G), \tag{14}$$

then $\|\mathcal{P}_1\| \leq 1$. On the other hand, from the fourth part of Lemma 3, if $\gamma_1 < \gamma_2$ then $\|\mathcal{P}_2\| < 1$. From the discussions of the two cases above, we see that if $\gamma_1 < \gamma_2 \leq \gamma_1 + 2\lambda_n(\mathcal{G})$ then $\rho(\mathcal{T}_{\gamma_1,\gamma_2}) = \rho(\hat{\mathcal{T}}_{\gamma_1,\gamma_2}) < 1$, and the convergence of the TGHSS method is achieved. Furthermore, a similar argument can be used to show that if

$$\gamma_2 < \gamma_1 + 2\lambda_n(\mathcal{G}), \tag{15}$$

then $\gamma_1 \leq \gamma_2$ is a sufficient condition for the convergence of the proposed method.

As seen in Theorem 4, the spectral radius of iteration matrix has been bounded as:

$$\rho(\mathcal{T}_{\gamma_1,\gamma_2}) \leq \|P_1\| \|\mathcal{P}_2\| \leq \max_{\omega \in \sigma(\mathcal{G})} \frac{|\gamma_2 - \omega|}{\gamma_1 + \omega} = \max_{\nu_i \in \sigma(G(A))} \max_{\mu_j \in \sigma(G(B))} \frac{|\gamma_2 - (\nu_i + \mu_j)|}{\gamma_1 + (\nu_i + \mu_j)} = \sigma(\gamma_1, \gamma_2),$$

In the next theorem, which can be proved similar to Corollary 2.3 in Aghazadeh et al. (2015b), the proposed upper bound for spectral radius of iteration matrix is minimized for a fix value of γ_1. Indeed, the optimal value of γ_2 which minimizes the upper bound is given for a fix value of γ_1.

Theorem 5. *Let the assumptions of Theorem 4 be fulfilled. Moreover, let*

$$\mathfrak{B}_1(\gamma) = \frac{\gamma(\lambda_n(\mathcal{G}) + \lambda_1(\mathcal{G})) + 2\lambda_n(G)\lambda_1(\mathcal{G})}{2\gamma + \lambda_n(\mathcal{G}) + \lambda_1(\mathcal{G})},$$

and

$$\mathfrak{B}_2(\gamma) = \frac{\lambda_1^2(\mathcal{G}) + \gamma(\lambda_1(\mathcal{G}) - \lambda_n(\mathcal{G})) - \lambda_1(\mathcal{G})\lambda_n(\mathcal{G})}{(2\gamma + \lambda_1(\mathcal{G}) + \lambda_n(\mathcal{G}))(\gamma + \lambda_1(\mathcal{G}))}.$$

Then, the convergence factor is given by the spectral radius $\rho(\mathcal{T}_{\gamma_1,\gamma_2})$ of the matrix $\mathcal{T}_{\gamma_1,\gamma_2}$, which is bounded as

$$\rho(\mathcal{T}_{\gamma_1,\gamma_2}) \leq \max_{\nu_i \in \sigma(G(A))} \max_{\mu_j \in \sigma(G(B))} \frac{|\gamma_2 - (\nu_i + \mu_j)|}{\gamma_1 + (\nu_i + \mu_j)} = \sigma(\gamma_1, \gamma_2),$$

(i) *If $\mathcal{G}$ is positive definite, $\mathcal{K}$ is positive semidefinite matrix and γ_1 is set to be γ_1^*, then*

$$\gamma_2^{opt} = \begin{cases} \gamma_1^*, & \gamma_1^* > \sqrt{\lambda_1(\mathcal{G})\lambda_n(\mathcal{G})}, \\ \mathfrak{B}_1(\gamma_1^*), & \textit{otherwise}, \end{cases}$$

where γ_2^{opt} is the optimal value of γ_2. Moreover, the corresponding upper bounds are given by

$$\sigma(\gamma_1^*, \gamma_2^{opt}) = \begin{cases} \frac{\gamma_1^* - \lambda_n(\mathcal{G})}{\alpha^* + \lambda_n(\mathcal{G})}, & \gamma_1^* > \sqrt{\lambda_1(\mathcal{G})\lambda_n(\mathcal{G})}, \\ \mathfrak{B}_2(\gamma_1^*), & \textit{otherwise}. \end{cases}$$

(ii) *If $\mathcal{G}$ is positive semidefinite, $\mathcal{K}$ is positive definite matrix, and γ_2 is set to be γ_2^*, then*

$$\gamma_1^{opt} = \gamma_2^* + \frac{1}{2}\lambda_n(\mathcal{K}),$$

where γ_1^{opt} is the optimal value of γ_1. Moreover,

$$\sigma(\gamma_1^{opt}, \gamma_2^*) = \begin{cases} \frac{\lambda_1(\mathcal{G}) - \gamma_2^*}{\lambda_1(\mathcal{G}) + \gamma_2^* + \lambda_n(\mathcal{K})/2}, & \gamma_2^* \leq \mathfrak{B}_1(\gamma_1^{opt}), \\ \frac{\gamma_2^*}{\gamma_2^* + \lambda_n(\mathcal{K})/2}, & \gamma_2^* \geq \mathfrak{B}_1(\gamma_1^{opt}). \end{cases}$$

(iii) *Suppose that $\mathcal{G}$ and $\mathcal{K}$ are positive definite matrices and γ_1 is set to be the prescribed value γ_1^*. Moreover, let*

$$\Delta = \sqrt{\lambda_n(\mathcal{K})^2 + 4(\lambda_1(\mathcal{G}) + \lambda_n(\mathcal{G}))\lambda_n(\mathcal{K}) + 16\lambda_n(\mathcal{G})\lambda_1(\mathcal{G})},$$

then

$$\gamma_2^{opt} = \begin{cases} \gamma_1^* - \frac{1}{2}\lambda_n(\mathcal{K}), & \gamma_1^* > \frac{1}{4}(\lambda_n(\mathcal{K}) + \sqrt{\Delta}), \\ \mathfrak{B}_1(\gamma_1^*), & \textit{otherwise}, \end{cases}$$

where γ_2^{opt} is the optimal value of γ_2. In addition, we have

$$\sigma(\gamma_1^*, \gamma_2^{opt}) = \begin{cases} \frac{\gamma_1^* - \lambda_n(\mathcal{K})/2 - \lambda_n(\mathcal{G})}{\gamma_1^* + \lambda_n(\mathcal{G})}, & \gamma_1^* > \frac{1}{4}(\lambda_n(\mathcal{K}) + \sqrt{\Delta}), \\ \mathfrak{B}_2(\gamma_1^*), & \textit{otherwise}. \end{cases}$$

It is noted that the computation of the optimal parameter in the HSS method and its derivatives are essentially problem-based. However, some results have been given in this directions in the literature (Bai, 2009; Bai et al., 2006b).

In each half-step of the TGHSS iteration method we need to solve a Sylvester matrix equation. For large Sylvester matrix equations, it is impractical and highly cost prohibitive. The inexact HSS (IHSS) iteration method has been presented to solve large sparse non-Hermitian positive definite linear systems in Bai et al. (2008) and Bai (2011). Indeed, since the HSS method is costly and impractical in actual implementations, the IHSS method has been proposed

to overcome this disadvantage and further improve the performance of the HSS method. To improve computing efficiency of the TGHSS iteration method we also use inexact TGHSS (ITGHSS) iteration, that is to solve the two subproblems iteratively. In the first half-step of the ITGHSS iteration the corresponding Sylvester equation is solved by the Global CG (Gl-CG) algorithm (Guennouni et al., 2002) and in the second half-step by the Global GMRES (Gl-GMRES) algorithm (Guennouni et al., 2002). Convergence of the method can be stated similar to that of the IHSS method presented in Bai (2011) and omitted here.

ITGHSS iteration method: Let $X^{(0)} \in \mathbb{C}^{m\times n}$ be an initial guess., $X^{(k+1)} \in \mathbb{C}^{m\times n}$ for $k = 0, 1, 2, \ldots$ is computed by using the following iteration scheme until $\{X^{(k)}\}_{k=0}^{\infty}$ satisfies the stopping criterion:
Set $R^{(k)} = C - AX^{(k)} - X^{(k)}B$ and approximately solve

$$(\alpha_1 I + G(A))Z^{(k)} + Z^{(k)}(\beta_1 I + G(B)) = R^{(k)}$$

by the Gl-CG algorithm such that the residual matrix $P^{(k)} = R^{(k)} - (\alpha_1 I + G(A))Z^{(k)} - Z^{(k)}(\beta_1 I + G(B))$ satisfies $\|P^{(k)}\|_F \leq \eta_k \|R^{(k)}\|_F$ and then set

$$X^{(k+\frac{1}{2})} = X^{(k)} + Z^{(k)},$$

Set $R^{(k+\frac{1}{2})} = C - AX^{(k+\frac{1}{2})} - X^{(k)+\frac{1}{2}}B$ and approximately solve

$$(\alpha_2 I + S(A) + K(A))Z^{(k+\frac{1}{2})} + Z^{(k+\frac{1}{2})}(\beta_2 I + S(B) + K(B)) = R^{(k+\frac{1}{2})},$$

by the Gl-GMRES algorithm such that the residual matrix $Q^{k+\frac{1}{2}} = R^{(k+\frac{1}{2})} - (\alpha_2 I + S(A) + K(A))Z^{(k+\frac{1}{2})} - Z^{(k+\frac{1}{2})}(\beta_2 I + S(B) + K(B))$ satisfies $\|Q^{(k+\frac{1}{2})}\|_F \leq \tau_k \|R^{(k+\frac{1}{2})}\|_F$ and then set

$$X^{(k+1)} = X^{(k+\frac{1}{2})} + Z^{(k+\frac{1}{2})}.$$

Numerical examples

In this section, some numerical examples are given to illustrate the effectiveness of the TGHSS and ITGHSS iteration methods for solving the Sylvester matrix equation $AX + XB = C$. All the computations have been implemented in MATLAB 8.1 software on a PC with Core i7, 2.67GHz CPU and 4.00GB RAM. In all the examples the initial guess is set to be the zero matrix and the iteration is stopped as

$$\frac{\|R^{(k)}\|_F}{\|R^{(0)}\|_F} \leq 10^{-6},$$

where $R^{(k)} = C - AX^{(k)} - X^{(k)}B$. In all the examples, C is an $m \times n$ matrix whose entries are all equal to one.

In the half-steps of the PSS, GHSS and TGHSS methods we use the Bartels and Stewart method to solve the inner Sylvester matrix equations, for further details see Bartels and Stewart (1972). In the inexact version of the iteration methods, i.e., IGHSS and TGHSS methods, the Gl-CG and Gl-GMRES iterations are used to solve the Sylvester matrix equations arisen in the first half-step and the second half-step, respectively. The inner iterations are terminated if the current residuals satisfy

$$\frac{\|P^{(k)}\|_F}{\|R^{(k)}\|_F} \leq \eta_k = \max\{0.1\delta^k, 10^{-6}\}, \qquad \frac{\|Q^{(k+\frac{1}{2})}\|_F}{\|R^{(k+\frac{1}{2})}\|_F} \leq \tau_k = \max\{0.1\delta^k, 10^{-6}\},$$

where δ is a control tolerance. In all the examples, we choose $\delta = 0.9$ to implement the iterative methods, and the maximum number of outer iterations and total number of inner iterations are set to be 1000.

Example 1. In this example, we consider the continuous Sylvester equation with the matrices (see Bai (2011))

$$A = B = M + \frac{100}{(n+1)^2} I + 2rN,$$

Table 1. The optimal values of parameter α for PSS and GHSS methods in Example 1.

n	PSS			GHSS		
	$r = 0.01$	$r = 0.1$	$r = 1.0$	$r = 0.01$	$r = 0.1$	$r = 1.0$
8	2.54	2.57	2.67	1.20	1.20	1.50
16	1.53	1.65	1.75	0.34	0.35	1.25
32	1.23	1.24	1.25	0.80	0.88	1.05
64	1.05	1.11	0.99	0.80	0.88	0.98
128	1.02	1.03	0.96	0.80	0.89	0.95
256	0.96	1.02	0.96	0.80	0.86	0.95

Table 2. The optimal values of parameters (α_1, α_2) for TGHSS method in Example 1.

n	$r = 0.01$	$r = 0.1$	$r = 1.0$
8	$(1.25, 1.35)$	$(0.35, 1.15)$	$(1.50, 1.55)$
16	$(0.35, 0.65)$	$(0.35, 1.15)$	$(0.70, 1.45)$
32	$(0.09, 0.65)$	$(0.08, 1.15)$	$(0.45, 1.15)$
64	$(0.02, 0.65)$	$(0.01, 1.02)$	$(0.40, 1.02)$
128	$(0.01, 0.65)$	$(0.01, 1.05)$	$(0.40, 1.02)$
256	$(0.005, 0.6)$	$(0.005, 0.95)$	$(0.40, 1.02)$

Table 3. IT and CPU time for GHSS, PSS and TGHSS methods with $r = 0.01$ in Example 1.

n	PSS		GHSS		TGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	11	0.0185	3	0.0143	3	0.141
16	17	0.0552	3	0.0152	3	0.0150
32	21	0.05	15	0.04	3	0.02
64	24	0.18	19	0.17	3	0.04
128	24	1.17	19	1.03	3	0.16
256	24	7.34	19	6.01	3	0.98

Table 4. IT and CPU time for GHSS, PSS and TGHSS methods with $r = 0.1$ in Example 1.

n	PSS		GHSS		TGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	11	0.0185	3	0.0145	3	0.0144
16	16	0.0542	6	0.0162	4	0.0160
32	21	0.05	16	0.04	4	0.02
64	23	0.17	20	0.16	5	0.05
128	24	1.33	21	1.15	4	0.21
256	23	7.03	20	6.43	4	1.32

where I is the identity matrix and $M, N \in \mathbb{R}^{n\times n}$ are the tridiagonal matrices given by

$$M = \text{tridiag}(-1, 2.3, -1), \qquad N = \text{tridiag}(0.5, 0, -0.5).$$

To apply the iterative methods, we consider $G(A) = G(B) = M$, $K(A) = K(B) = \frac{100}{(n+1)^2} I$ and $S(A) = S(B) = 2rN$. We consider $\alpha = \beta$ for the PSS method, $\alpha = \beta$ for the GHSS method and $\alpha_1 = \beta_1$, $\alpha_2 = \beta_2$ for the TGHSS method to find optimal values. The optimal values of unknown parameters in the PSS, GHSS and TGHSS methods and their inexact version are determined experimentally. These values are given in Tables 1-2. The outer iteration numbers and the CPU time for the convergence of the PSS, GHSS and TGHSS methods are given for $r = 0.01, 0.1, 1.0$ in Tables 3-5. This tables show that the TGHSS method is more effective than the GHSS and PSS methods.

Inexact version of the GHSS, PSS and TGHSS methods have also been applied to solve the continuous Sylvester equation. The optimal values of the parameters in the IPSS, the IGHSS and the ITGHSS methods are given in Tables 6-7. Number of outer iterations, CPU time and total iteration number of Gl-GMRES and Gl-CG methods are reported in Tables 8-10. From Tables 11-13 it is affirmed that ITGHSS method is superior to the IGHSS and the IPSS methods.

Example 2. In this example, the matrices A and B in equation (1) are defined as (see Bai (2011))

$$A = \text{diag}(1, 2, \ldots, n) + rL^T,$$

Table 5. IT and CPU time for GHSS, PSS and TGHSS methods with $r = 1.0$ in Example 1.

n	PSS		GHSS		TGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	10	0.0179	6	0.0173	6	0.0165
16	14	0.0375	11	0.0203	10	0.0203
32	18	0.04	17	0.04	13	0.03
64	21	0.20	21	0.19	15	0.13
128	21	1.20	22	1.21	15	0.74
256	21	7.01	22	7.11	15	4.87

Table 6. The optimal values of parameter α for IPSS and IGHSS methods in Example 1.

n	IPSS			IGHSS		
	$r = 0.01$	$r = 0.1$	$r = 1.0$	$r = 0.01$	$r = 0.1$	$r = 1.0$
8	2.39	2.45	2.60	1.20	1.16	1.61
16	1.86	1.86	1.83	0.47	0.43	1.22
32	1.49	1.55	1.70	0.78	0.83	1.09
64	1.37	1.44	1.52	0.80	0.87	1.10
128	1.28	1.35	1.40	0.80	0.82	1.11
256	2.11	2.12	2.14	0.75	1.03	1.09

Table 7. The optimal values of parameters (α_1, α_2) for ITGHSS method in Example 1.

n	$r = 0.01$	$r = 0.1$	$r = 1.0$
8	$(1.16, 1.40)$	$(1.15, 1.35)$	$(1.55, 1.38)$
16	$(0.34, 1.38)$	$(0.39, 1.38)$	$(1.24, 1.38)$
32	$(0.08, 1.65)$	$(0.09, 1.37)$	$(0.70, 1.28)$
64	$(0.02, 2.11)$	$(0.01, 1.69)$	$(0.40, 1.17)$
128	$(0.01, 1.81)$	$(0.01, 1.34)$	$(0.35, 1.13)$
256	$(0.01, 2.05)$	$(0.01, 1.38)$	$(0.38, 1.20)$

Table 8. IT and CPU time for IPSS, IGHSS and ITGHSS methods with $r = 0.01$ in Example 1.

n	IPSS		IGHSS		ITGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	11	0.0170	4	0.0117	4	0.0106
16	18	0.0225	8	0.0149	4	0.0118
32	25	0.04	17	0.03	4	0.02
64	28	0.14	21	0.08	4	0.03
128	28	0.40	22	0.27	4	0.07
256	48	3.44	23	1.61	4	0.38

Table 9. IT and CPU time for IPSS, IGHSS and ITGHSS methods with $r = 0.1$ in Example 1.

n	IPSS		IGHSS		ITGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	11	0.0168	4	0.0117	4	0.0111
16	18	0.0209	9	0.0157	5	0.0120
32	26	0.04	16	0.03	5	0.02
64	30	0.12	21	0.08	5	0.03
128	30	0.40	23	0.29	5	0.08
256	48	3.05	24	1.68	5	0.42

and

$$B = 2^{-t}I + \mathrm{diag}(1, 2, \ldots, n) + rL^T + 2^{-t}L,$$

Table 10. IT and CPU time for IPSS, IGHSS and ITGHSS methods with $r = 1.0$ in Example 1.

n	IPSS		IGHSS		ITGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	11	0.0225	7	0.0153	7	0.0140
16	11	0.0288	12	0.0171	11	0.0160
32	22	0.16	19	0.03	14	0.03
64	30	0.24	23	0.09	16	0.06
128	30	0.41	24	0.31	16	0.21
256	48	3.71	25	1.72	16	1.11

Table 11. Total number of iterations for IPSS method in Example 1.

n	$r = 0.01$		$r = 0.1$		$r = 1.0$	
	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG
8	11	19	11	19	11	21
16	18	37	18	36	11	25
32	25	94	26	94	22	46
64	28	106	30	104	30	62
128	28	106	30	101	30	62
256	48	49	48	49	48	49

Table 12. Total number of iterations for IGHSS method in Example 1.

n	$r = 0.01$		$r = 0.1$		$r = 1.0$	
	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG
8	4	9	4	9	7	14
16	8	18	9	19	12	22
32	17	37	16	47	19	34
64	21	46	21	59	23	46
128	22	51	23	56	24	48
256	23	47	24	48	25	49

Table 13. Total number of iterations for ITGHSS method in Example 1.

n	$r = 0.01$		$r = 0.1$		$r = 1.0$	
	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG
8	4	8	4	9	7	12
16	4	14	4	16	11	22
32	4	21	5	23	14	40
64	4	22	5	27	16	48
128	4	24	5	28	16	48
256	4	23	5	28	16	47

where t is a problem parameter with non-negative values which is specified in actual computations and

$$L = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 1 & \cdots & 1 & 0 \end{pmatrix}.$$

Let $\tilde{L} = L^T + L$, $\tilde{D}_1 = \text{diag}(1, 2, \ldots, n)$ and $\tilde{D}_2 = \text{diag}(1, 1, \ldots, 1)$. The Hermitian part of the matrix A and B can be split as follows:

$$H(A) = G(A) + K(A) = \frac{r}{2}\left(\tilde{L} + \frac{1}{r}\tilde{D}_1\right) + \frac{1}{2}\tilde{D}_1,$$
$$H(B) = G(B) + K(B) = \frac{1}{2}\left(r\left(\tilde{L} + \frac{1}{r}\tilde{D}_1\right) + 2^{-t}(\tilde{L} + \tilde{D}_2)\right) + \frac{1}{2}\left(\tilde{D}_1 + 2^{-t}\tilde{D}_2\right).$$

Table 14. The optimal values of parameter γ for PSS and GHSS methods in Example 2.

n	IPSS			IGHSS		
	$r = 0.01$	$r = 0.1$	$r = 1.0$	$r = 0.01$	$r = 0.1$	$r = 1.0$
8	7	7	13	4	4	7
16	12	11	17	6	6	6
32	19	17	23	10	11	9
64	31	34	34	18	19	15
128	57	52	60	30	29	25
256	93	82	86	51	48	46

Table 15. The optimal values of parameters (γ_1, γ_2) for TGHSS method in Example 2.

n	$r = 0.01$	$r = 0.1$	$r = 1.0$
8	$(2, 6)$	$(2, 6)$	$(2, 6)$
16	$(9, 2)$	$(9, 3)$	$(3, 11)$
32	$(18, 3)$	$(19, 3)$	$(15, 6)$
64	$(33, 4)$	$(34, 5)$	$(34, 5)$
128	$(70, 5)$	$(60, 5)$	$(70, 5)$
256	$(136, 10)$	$(110, 10)$	$(140, 12)$

Table 16. IT and CPU time for PSS, GHSS and TGHSS methods with $r = 0.01$ in Example 2.

n	PSS		GHSS		TGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	14	0.0248	7	0.0220	6	0.0207
16	17	0.0317	9	0.0283	7	0.0238
32	21	0.15	11	0.09	7	0.04
64	25	0.24	13	0.19	8	0.11
128	27	1.37	14	0.84	8	0.45
256	32	9.01	15	5.06	8	2.67

Table 17. IT and CPU time for PSS, GHSS and TGHSS methods with $r = 0.1$ in Example 2.

n	PSS		GHSS		TGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	13	0.0245	7	0.0225	6	0.0210
16	16	0.0305	9	0.0295	7	0.0242
32	19	0.13	10	0.08	7	0.04
64	20	0.19	11	0.16	8	0.11
128	25	1.27	15	0.94	10	0.69
256	31	8.91	19	6.41	10	3.41

We first test the PSS, the GHSS and the TGHSS iteration methods and compare their numerical results. To do so, these methods have been applied to solve the equation (1). The used values of $\gamma = \alpha + \beta$, $\gamma_1 = \alpha_1 + \beta_1$ and $\gamma_2 = \alpha_2 + \beta_2$ are given in Tables 14-15. The number of iterations and CPU time of the methods are given for $r = 0.01, 0.1, 1.0$ in Tables 16-18. The IPSS, IGHSS and ITGHSS methods have also been applied to solve the continuous Sylvester equation. The optimal values of the parameters in these methods are given in Tables 19-20. Number of outer iterations, CPU time and total iteration number of Gl-GMRES and Gl-CG are reported in Tables 21-26. As the numerical results show, the TGHSS and ITGHSS are more effective than the GHSS and IGHSS methods to solve the equation (1).

Example 3. In this example, we consider equation (1) with

$$A = \begin{pmatrix} 3.2 & 1 & & & 1 \\ 2 & 3.2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & 3.2 & 1 \\ 1 & & & 2 & 3.2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 4.2 & 1 & & & 1 \\ 3 & 4.2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 3 & 4.2 & 1 \\ 1 & & & 3 & 4.2, \end{pmatrix}.$$

Table 18. IT and CPU time for PSS, GHSS and TGHSS methods with $r = 1.0$ in Example 2.

n	PSS		GHSS		TGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	19	0.0251	7	0.0229	6	0.0219
16	21	0.0319	9	0.0271	8	0.0264
32	24	0.15	12	0.09	10	0.06
64	28	0.31	15	0.20	11	0.15
128	31	1.38	18	1.09	12	0.77
256	35	10.16	20	6.92	12	4.22

Table 19. The optimal values of parameter γ for IPSS and IGHSS methods in Example 2.

n	IPSS			IGHSS		
	$r = 0.01$	$r = 0.1$	$r = 1.0$	$r = 0.01$	$r = 0.1$	$r = 1.0$
8	6	5	8	4	4	4
16	9	9	12	5	6	6
32	14	14	21	8	8	12
64	26	28	63	14	15	29
128	51	45	93	26	23	42
256	92	92	173	43	45	84

Table 20. The optimal values of parameter γ for ITGHSS method in Example 2.

n	$r = 0.01$	$r = 0.1$	$r = 1.0$
8	$(6, 2)$	$(2, 6)$	$(3, 8)$
16	$(10, 2)$	$(9, 2)$	$(10, 24)$
32	$(18, 2)$	$(16, 3)$	$(9, 33)$
64	$(43, 4)$	$(46, 5)$	$(18, 65)$
128	$(67, 6)$	$(68, 7)$	$(35, 131)$
256	$(127, 9)$	$(130, 13)$	$(70, 260)$

Table 21. IT and CPU time for IPSS, IGHSS and ITGHSS methods with $r = 0.01$ in Example 2.

n	IPSS		IGHSS		ITGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	14	0.0180	8	0.0149	6	0.0138
16	17	0.0234	10	0.0186	7	0.0173
32	21	0.04	13	0.03	8	0.02
64	25	0.19	15	0.13	10	0.08
128	30	1.41	17	0.93	10	0.55
256	34	11.52	19	8.33	11	4.92

Table 22. IT and CPU time for IPSS, IGHSS and ITGHSS methods with $r = 0.1$ in Example 2.

n	IPSS		IGHSS		ITGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	14	0.0178	8	0.0152	6	0.0140
16	17	0.0240	10	0.0191	7	0.0174
32	21	0.04	13	0.03	8	0.02
64	25	0.19	15	0.13	11	0.09
128	28	1.31	18	0.91	10	0.57
256	30	10.59	19	8.28	11	4.28

The GHSS, TGHSS, IGHSS and ITGHSS methods have been implemented with the following splitting for the Hermitian part of the matrices A and B:

$$H(A) = \frac{1}{10}H(A) + \frac{9}{10}H(A) = G(A) + K(A),$$
$$H(B) = \frac{1}{10}H(B) + \frac{9}{10}H(B) = G(B) + K(B).$$

Table 23. IT and CPU time for IPSS, IGHSS and ITGHSS methods with $r = 1.0$ in Example 2.

n	IPSS		IGHSS		ITGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	20	0.0270	8	0.0160	6	0.0141
16	25	0.0297	10	0.0192	7	0.0174
32	31	0.06	12	0.03	6	0.02
64	37	0.27	13	0.13	6	0.06
128	40	1.75	13	0.83	6	0.32
256	41	14.82	13	5.44	6	2.54

Table 24. Total number of iterations for IPSS method in Example 2.

n	$r = 0.01$		$r = 0.1$		$r = 1.0$	
	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG
8	14	19	14	16	20	55
16	17	25	17	31	25	87
32	21	36	21	43	31	117
64	25	50	25	59	37	106
128	30	59	28	67	40	131
256	34	76	30	76	41	143

Table 25. Total number of iterations for IGHSS method in Example 2.

n	$r = 0.01$		$r = 0.1$		$r = 1.0$	
	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG
8	8	15	8	15	8	20
16	10	18	10	20	10	24
32	13	24	13	22	12	30
64	15	20	15	30	13	31
128	17	33	18	29	13	32
256	19	33	19	32	13	32

Table 26. Total number of iterations for ITGHSS method in Example 2.

n	$r = 0.01$		$r = 0.1$		$r = 1.0$	
	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG	Gl-GMRES	Gl-CG
8	6	10	6	12	6	14
16	7	13	7	14	7	13
32	8	15	8	16	6	15
64	10	16	11	19	6	14
128	11	20	10	21	6	14
256	14	22	11	24	6	14

Similar to previous examples, the values of unknown parameters have been experimentally determined and given in Tables 27. The numerical results of the PSS, GHSS and TGHSS methods have been listed in Table 28. The IPSS, IGHSS and ITGHSS methods have also been applied to approximate the solutions of equation (1). The optimal values of the parameters in the IPSS, the IGHSS and the ITGHSS methods are presented in Table 29. Number of outer iterations, total iteration number of the Gl-GMRES and the Gl-CG algorithms and the CPU time have been reported in Table 30. As the numerical results show, the TGHSS and ITGHSS method are reliable and effective methods to solve continuous Sylvester equation and the performance of these methods are better than the GHSS/PSS and IGHSS/IPSS algorithms, respectively.

Example 4. Consider the convection-diffusion equation (see Robbe and Sadkane (2002))

$$-\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+2c_1\frac{\partial u}{\partial x}+2c_2\frac{\partial u}{\partial y}-2c_3u=f(x,y),\qquad (x,y)\in\Omega, \tag{16}$$

with the homogeneous Dirichlet boundary conditions, where c_1, c_2 and c_3 are constants, and $\Omega=[0,1]\times[0,1]$. The five-point stencil and centered finite differences discretization is applied to discretize of the equation (16) on a uniform

Table 27. The optimal values of unknown parameters for PSS, GHSS and TGHSS methods in Example 3.

n	PSS	GHSS	TGHSS
	(α, β)	(α, β)	$(\alpha_1, \beta_1, \alpha_2, \beta_2)$
8	$(1.7, 2.4)$	$(0.7, 0.4)$	$(1.6, 0.5, 0.7, 0.5)$
16	$(1.7, 2.4)$	$(0.8, 0.5)$	$(1.2, 0.6, 0.7, 0.5)$
32	$(1.7, 2.7)$	$(0.8, 0.5)$	$(1.6, 0.4, 0.7, 0.5)$
64	$(1.7, 3.2)$	$(0.8, 0.5)$	$(2.5, 0.5, 0.8, 0.5)$
128	$(1.5, 4.5)$	$(0.7, 0.5)$	$(4.1, 3.5, 0.7, 0.5)$
256	$(1.5, 4.9)$	$(0.7, 0.5)$	$(4.1, 3.5, 0.8, 0.6)$

Table 28. IT and CPU time for PSS, GHSS and TGHSS methods in Example 3.

n	PSS		GHSS		TGHSS	
	IT	CPU	IT	CPU	IT	CPU
8	21	0.0226	7	0.0171	5	0.0152
16	23	0.0362	6	0.0283	6	0.0213
32	22	0.0668	6	0.0343	5	0.0288
64	20	0.1139	6	0.0782	5	0.0633
128	17	0.6034	6	0.4666	4	0.2815
256	15	3.6291	6	2.1935	3	1.0963

Table 29. The optimal values of unknown parameters for IPSS, IGHSS and ITGHSS methods in Example 3.

n	IPSS	IGHSS	ITGHSS
	(α, β)	(α, β)	$(\alpha_1, \beta_1, \alpha_2, \beta_2)$
8	$(6.5, 7.5)$	$(0.1, 1.1)$	$(4.3, 2.5, 0.1, 0.3)$
16	$(6.5, 7.5)$	$(0.1, 1.1)$	$(4.3, 2.5, 0.1, 0.4)$
32	$(6.6, 7.5)$	$(0.1, 1.1)$	$(4.3, 2.5, 0.1, 0.3)$
64	$(6.5, 7.5)$	$(0.1, 1.2)$	$(4.3, 2.5, 0.1, 0.9)$
128	$(6.5, 7.5)$	$(0.1, 1.2)$	$(4.3, 2.5, 0.1, 0.8)$
256	$(6.5, 7.5)$	$(0.1, 1.2)$	$(4.3, 2.5, 0.1, 1.2)$

Table 30. IT and CPU time for IPSS, IGHSS and ITGHSS methods in Example 3.

n	IPSS				IGHSS				ITGHSS			
	IT	Gl-GMRES	Gl-CG	CPU	IT	Gl-GMRES	Gl-CG	CPU	IT	Gl-GMRES	Gl-CG	CPU
8	38	38	74	0.0294	6	6	8	0.0121	5	5	5	0.0107
16	33	33	71	0.0299	6	6	7	0.0126	5	5	5	0.0111
32	26	26	56	0.0386	6	6	7	0.0145	5	5	5	0.0130
64	20	20	44	0.0811	5	5	6	0.0251	4	4	4	0.0205
128	16	16	35	0.1824	5	5	6	0.0731	4	4	4	0.0530
256	12	12	26	0.7543	5	5	6	0.3279	3	3	3	0.1803

grid with $n \times n$ interior nodes (with step size $h = 1/(n+1)$) which leads to the Sylvester equation (1) with:

$$A = \text{tridiag}(-1 - c_1 h, 2 - c_3 h^2, -1 + c_1 h), \tag{17}$$
$$B = \text{tridiag}(-1 - c_2 h, 2 - c_3 h^2, -1 + c_2 h). \tag{18}$$

The ITGHSS method is applied to solve this example for $n = 256$, $c_1 = c_2 = 1$ and $c_3 = -1000$. To this end, the following splitting is considered to use in the proposed method:

$$A = (H(A) - \lambda_{\min}(H(A))I) + \lambda_{\min}(H(A))I + S(A) = G(A) + K(A) + S(A),$$
$$B = (H(B) - \lambda_{\min}(H(B))I) + \lambda_{\min}(H(B))I + S(B) = G(B) + K(B) + S(B).$$

where $\lambda_{\min}(H(A))$ and $\lambda_{\min}(H(B))$ are the minimum eigenvalue of $H(A)$ and $H(B)$, respectively. Number of outer iterations, total iteration number of Gl-GMRES and Gl-CG and CPU time are given for various values of α_1, β_1, α_2 and β_2 in Tables 31-34.

Table 31. Results of ITGHSS for $n = 250$, $(\beta_1, \alpha_2, \beta_2) = (0.05, 0.5, 0.5)$ and various values of α_1 in Example 4.

α_1	Its.	Gl-GMRES	Gl-CG	CPU
0.01	16	16	246	2.01
0.05	30	30	233	2.74
0.10	46	46	260	3.72
0.15	60	60	292	4.44
0.20	73	73	324	5.32
0.25	86	86	357	6.32
0.30	96	96	379	6.85
0.35	107	107	402	7.38
0.40	117	117	418	8.04
0.45	126	126	436	8.73

Table 32. Results of ITGHSS for $n = 250$, $(\alpha_1, \alpha_2, \beta_2) = (0.01, 0.5, 0.5)$ and various values of β_1 in Example 4.

β_1	Its.	Gl-GMRES	Gl-CG	CPU
0.01	23	23	1812	9.51
0.05	16	16	246	2.01
0.10	33	33	232	2.94
0.15	49	49	266	3.92
0.20	63	63	299	4.69
0.25	76	76	332	5.53
0.30	88	88	366	6.36
0.35	99	99	389	7.09
0.40	109	109	402	7.93
0.45	119	119	422	8.32

Table 33. Results of ITGHSS for $n = 250$, $(\alpha_1, \beta_1, \beta_2) = (0.05, 0.05, 0.2)$ and various values of α_2 in Example 4.

α_2	Its.	Gl-GMRES	Gl-CG	CPU
0.1	26	26	250	2.57
0.2	27	27	246	2.58
0.3	28	28	244	2.59
0.4	29	29	241	2.79
0.5	29	29	230	2.56
0.6	29	29	232	2.58
0.7	30	30	235	2.65
0.8	30	30	233	2.64
0.9	30	30	230	2.61
1.0	30	30	232	2.63

Table 34. Results of ITGHSS for $n = 250$, $(\alpha_1, \beta_1, \alpha_2) = (0.05, 0.05, 0.15)$ and various values of β_2 in Example 4.

β_2	Its.	Gl-GMRES	Gl-CG	CPU
0.1	27	27	263	2.55
0.2	28	28	261	2.79
0.3	28	28	250	2.60
0.4	28	28	238	2.55
0.5	29	29	235	2.70
0.6	29	29	234	2.68
0.7	30	30	236	2.71
0.8	30	30	237	2.71
0.9	30	30	230	2.66
1.0	30	30	239	2.72

Conclusion

In this paper, the two-parameter generalized Hermitian and skew-Hermitian splitting (TGHSS) iteration method has been applied to solve the continuous Sylvester equation. Some convergence properties of the proposed method has been presented. The effectiveness and performance of the proposed method have been investigated in four examples. As the numerical results show, the TGHSS method is reliable and effective method to solve the continuous Sylvester equation.

Acknowledgments

The authors are grateful to anonymous referees and editor of the journal for their valuable comments and suggestions. The work of Davod Khojasteh Salkuyeh is partially supported by University of Guilan.

References

Anderson BDO, Agathoklis P, Jury EI, et al. (1986) Stability and the matrix Lyapunov equation for discrete 2-dimensional system. *IEEE Transactions on Circuits and Systems* 33: 261–267.

Aghazadeh N, Bastani M and Salkuyeh DK (2015a) Generalized Hermitian and skew-Hermitian splitting iterative method for image restoration. *Applied Mathematical Modelling* 39: 6126–6138.

Aghazadeh N, Salkuyeh DK and Bastani M (2015b) Two-parameter generalized Hermitian and skew-Hermitian splitting iteration method. *International Journal of Computer Mathematics*. DOI:10.1080/00207160.2015.1019873.

Bai Z-Z (2009) Optimal parameters in the HSS-like methods for saddle-point problems. *Numerical Linear Algebra with Applications* 16: 447–479.

Bai Z-Z (2011) On Hermitian and skew-Hermitian spliting iteration methods for continuous Sylvester equations. *Journal of Computational Mathematics* 29: 185–198.

Bai Z-Z, Deng Y-B and Gao Y-H (2006a) Iterative orthogonal direction methods for Hermitian minimum norm solutions of two consistent matrix equations. *Numerical Linear Algebra with Applications* 13: 801–823.

Bai Z-Z and Golub GH (2007) Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. *IMA Journal of Numerical Analysis* 27: 1–23.

Bai Z-Z, Golub GH and Li C-K (2006b) Optimal parameter in Hermitian and skew-Hermitian splitting method for certain two-by-two block matrices. *SIAM Journal on Scientific Computing* 28: 583–603.

Bai Z-Z, Golub GH, Lu L-Z, et al. (2005) Block triangular and skew-Hermitian splitting methods for positive-definite linear systems. *SIAM Journal on Scientific Computing* 26: 844–863.

Bai Z-Z, Golub GH and Ng MK (2003) Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *SIAM Journal on Matrix Analysis and Applications* 24: 603–626.

Bai Z-Z, Golub GH and Ng MK (2007) On successive-overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations. *Numerical Linear Algebra with Applications* 14: 319–335.

Bai Z-Z, Golub GH and Ng MK (2008) On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *Linear Algebra and its Applications* 428: 413–440.

Bai Z-Z, Golub GH and Pan J-Y (2004) Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. *Numerische Mathematik* 98: 1–32.

Bai Z-Z and Hadjidimos A (2014) Optimization of extrapolated Cayley transform with non-Hermitian positive definite matrix. *Linear Algebra and its Applications* 463: 322–339.

Bai Z-Z and Ng MK (2012) Erratum. *Numerical Linear Algebra with Applications* 19: 891.

Bai Z-Z and Yang X (2009) On HSS-based iteration methods for weakly nonlinear systems. *Applied Numerical Mathematics* 59: 2923–2936.

Bartels RH and Stewart GW (1972) Solution of the matrix equation $AX + XB = C$. *Communications of the ACM* 15: 820–826.

Beckermann B (2011) An error analysis for rational Galerkin projection applied to the sylvester equation. *SIAM Journal on Numerical Analysis* 49: 2430–2450.

Benner P, Li RC and Truhar N (2009) On the ADI method for Sylvester equations. *Journal of Computational and Applied Mathematics* 233: 1035–1045.

Benzi M (2009) A generalization of the Hermitian and skew-Hermitian splitting iteration. *SIAM Journal on Matrix Analysis and Applications* 31: 360–374.

Benzi M and Golub GH (2004) A preconditioner for generalized saddle point problems. *SIAM Journal on Matrix Analysis and Applications* 26: 20–41.

Benzi M and Ng MK (2006) Preconditioned iterative methods for weighted Toeplitz least squares problems. *SIAM Journal on Matrix Analysis and Applications* 27: 1106–1124.

Bertaccini D, Golub GH, Serra-Capizzano S, et al. (2005) Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection-diffusion equation. *Numerische Mathematik* 99: 441–484.

Ding F and Chen T (2005) Iterative least-squares solutions of coupled Sylvester matrix equations. *Systems & Control Letters* 54: 95–107.

Guennouni AE, Jbilou K and Riquet AJ (2002) Block Krylov subspace methods for solving large Sylvester equations. *Numerical Algorithms* 29: 75–96.

Evans DJ and Galligani E (1994) A parallel additive preconditioner for conjugate gradient method for $AX + XB = C$. *Parallel Computing* 20: 1055–1064.

Gallivan K, Vandendorpe A and Dooren PV (2004) Sylvester equations and projection-based model reduction. *Journal of Computational and Applied Mathematics* 162: 213–229.

Gantmacher FR (1959) *The Theory of Matrices*. New York: Chelsea Publishing Company.

Hached M (2015) A note on the Davison-Man method for Sylvester matrix equations. *Computational and Applied Mathematics*. DOI: 10.1007/s40314-015-0244-1.

Halanay A and Răsvan V (1993) *Applications of Lyapunov Methods in Stability*. Berlin: Kluwer Academic Publishers.

Ilic M-D (1989) New approaches to voltage monitoring and control. *IEEE Control Systems Magazine* 9: 5–11.

Johnson CR and Krukier LA (2009) General resolution of a convergence question of L. Krukier. *Numerical Linear Algebra with Applications* 16: 949-950.

Lancaster P and Tismenetsky M (1985) *The Theory of Matrices*. 2nd edn. Orlando: Academic Press.

Lewis F-L, Mertzios V-G, Vachtsevanos G, et al. (1990) Analysis of bilinear systems using Walsh functions. *IEEE Transactions on Automatic Control* 35: 119–123.

Niu Q, Wang X and Lu L-Z (2011) A relaxed gradient based algorithm for solving Sylvester equations. *Asian Journal of Control* 13: 461–464.

Robbé M and Sadkane M (2002) A convergence analysis of GMRES and FOM methods for Sylvester equations. *Numerical Algorithms* 30: 71–89.

Salkuyeh DK and Toutounian F (2006) New approaches for solving large Sylvester equations. *Applied Mathematics and Computation* 173: 9–18.

Sanches JM, Nascimento JC and Marques JS (2008) Medical image noise reduction using the Sylvester-Lyapunov equation. *IEEE Transactions on Image Processing* 17: 1522–1539.

Schaft AVD (2000) L_2*-Gain and Passivity Techniques in Nonlinear Control*. 2nd edn. London: Springer-Verlag.

Sorensen D and Antoulas A (2002) The Sylvester equation and approximate balanced reduction. *Linear Algebra and its Applications* 351: 671–700.

Wang X, Dai L and Liao D (2012) A modified gradient based algorithm for solving Sylvester equations. *Applied Mathematics and Computation* 218: 5620–5628.

Wang X, Li W-W and Mao L-Z (2013) On positive-definite and skew-Hermitian splitting iteration methods for continuous Sylvester equation $AX + XB = C$. *Computers & Mathematics with Applications* 66: 2352–2361.

Zheng Q-Q and Ma C-F (2014) On normal and skew-Hermitian splitting iteration methods for large sparse continuous Sylvester equations. *Journal of Computational and Applied Mathematics* 268: 145–154.

Zhou R, Wang X and Tang X-B (2015) A generalization of the Hermitian and skew-Hermitian splitting iteration method for solving Sylvester equations. *Applied Mathematics and Computation* 271: 609–617.