Filomat 31:8 (2017), 2381–2390
DOI 10.2298/FIL1708381T

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

An Accelerated Jacobi-gradient Based Iterative Algorithm for Solving Sylvester Matrix Equations

Zhaolu Tian[a], Maoyi Tian[b,c], Chuanqing Gu[d], Xiaoning Hao[a]

[a] *College of Big Data Science, Taiyuan University of Technology, Taiyuan 030024, P.R.China*
[b] *Geomatics College, Shandong University of Science and Technology, Qingdao 266590, P.R.China*
[c] *Key Laboratory of Surveying and Mapping Technology on Island and Reef,National Administration of Surveying, Mapping and Geoinfomation*
[d] *Department of Mathematics, Shanghai University, Shanghai 200444, P.R.China*

Abstract. In this paper, an accelerated Jacobi-gradient based iterative (AJGI) algorithm for solving Sylvester matrix equations is presented, which is based on the algorithms proposed by Ding and Chen [6], Niu et al. [18] and Xie et al. [25]. Theoretical analysis shows that the new algorithm will converge to the true solution for any initial value under certain assumptions. Finally, three numerical examples are given to verify the efficiency of the accelerated algorithm proposed in this paper.

1. Introduction

Consider the iterative solution of the following Sylvester matrix equation :

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

where $A \in R^{m\times m}$, $B \in R^{n\times n}$, $C \in R^{m\times n}$ are known matrices, and $X \in R^{m\times n}$ is the matrix to be determined.

Eq.(1) plays an important role in some fields of applied mathematics and control theory [1, 5, 11]. Eq.(1) is mathematically equivalent to the following linear equation

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

where $\mathcal{A} = I \otimes A + B^T \otimes I$, and vectors x and c contain the concatenated columns of the matrices X and C, respectively. It is well-known that Eq.(1) admits a unique solution if and only if A and $-B$ possess no common eigenvalues [14]. However, this is a numerically poor way to determine the solution X of Eq.(1), as the dimensions of matrices increase greatly and may be ill-conditioned, so this approach is only applicable for small sized Sylvester matrix equations.

2010 *Mathematics Subject Classification*. Primary 15A24; Secondary 65F30, 65F35
Keywords. Sylvester matrix equations, Jacobi-gradient based algorithm, Accelerated, Convergence
Received: 28 November 2015; Accepted: 15 April 2016
Communicated by Dijana Mosić
Corresponding author: Zhaolu Tian
This work is supported by the Key Laboratory of Surveying and Mapping Technology on Island and Reef, National Administration of Surveying, Mapping and Geoinfomation (2013A03), National key scientific instrument and equipment development projects (2013YQ120343), the National Natural Science Foundation of China (11401422) and the National Natural Science Foundation of China (11371243)
Email address: tianzhaolu2004@126.com (Zhaolu Tian)

When the matrices A and B are small and dense, the algorithms [3, 14] are attractive, which consist in transforming A and B into triangular or Hessenberg form by an orthogonal similarity transformation, and then solving the resulting matrix equation directly by a back-substitution process. When the matrices A and B are large and sparse, iterative methods are more efficient to solve Eq.(1) such as the Smith's method [19], the alternating-direction implicit (ADI) method [4, 16, 21], the HSS and corresponding methods [2, 23, 27, 28], the SSHI method [15], and the KPIM method [20], etc.

Recently, Ding et al.in [6, 9, 10, 12, 22, 24, 26] presented a gradient based iterative (GI) algorithm for solving a class of matrix equations by applying the so-called hierarchical identification principle [7, 8], which regards the unknown matrix as the system parameter matrix to be identified, and then construct a recursive formula to approximate the unknown solution. Niu et al. [18] proposed a relaxed gradient based iterative (RGI) algorithm for solving Sylvester matrix equations. The numerical experiments show that the convergent behavior of Niu's algorithm is better than Ding's algorithm when the relaxation factor is chosen appropriately. Xie et al. [25] proposed an accelerated gradient based iterative (AGBI) algorithm, on the basis of the information generated in the previous half-step, the authors introduce a relaxation factor to improve the RGI algorithm.

In [13], Fan, Gu and Tian realized the matrix multiplication in GI algorithm would cost large time and spaces if the matrices A and B are large and dense, so the authors presented a Jacobi-gradient iterative (JGI) algorithm.In order to improve the convergent rate of the JGI algorithm, in this paper, we propose an accelerated Jacobi-gradient based iterative (AJGI) algorithm for solving Sylvester matrix equations based on the algorithms [6, 13, 18, 25], the convergence condition of the AJGI algorithm is analyzed, and three numerical examples are given to compare the AJGI algorithm with the algorithms in [6, 13, 17, 18, 25].

2. GI algorithm, RGI algorithm, AGBI algorithm and JGI algorithm

We firstly recall the GI algorithm proposed by Ding et al. [6] for solving Eq.(1). Regarding Eq.(1) as the following two matrix equations:

$$AX = C - XB\ ,\ XB = C - AX. \tag{3}$$

From (3) Ding et al. presented the GI algorithm.

Algorithm 1: The gradient based iterative (GI) algorithm

Step 1: Given any two initial approximate solution block vectors $X_1(0), X_2(0)$, and then
$X(0) = [X_1(0) + X_2(0)]/2$

Step 2: For $k = 1, 2, \cdots$, until converges, do:

Step 3: $X_1(k) = X(k-1) + \mu A^T[C - AX(k-1) - X(k-1)B]$

Step 4: $X_2(k) = X(k-1) + \mu[C - AX(k-1) - X(k-1)B]B^T$

Step 5: $X(k) = [X_1(k) + X_2(k)]/2$

Step 6: End

It is shown in [6] that the GI algorithm converges as long as

$$0 < \mu < \frac{2}{\lambda_{\max}(AA^T) + \lambda_{\max}(B^TB)},$$

where $\lambda_{\max}(AA^T)$ is the largest eigenvalue of AA^T.

In [18], Niu at al. proposed a relaxed gradient based iterative algorithm for solving Eq.(1) by introducing a relaxed factor $\hat{\omega}$.

Algorithm 2: The relaxed gradient based iterative (RGI) algorithm

Step 1: Given two initial approximate solution block vectors $X_1(k-1)$ and $X_2(k-1)$

Step 2: For $k = 1, 2, \cdots$, until converges, do:

Step 3: $X(k-1) = \hat{\omega}X_1(k-1) + (1-\hat{\omega})X_2(k-1)$

Step 4: $X_1(k) = X(k-1) + (1-\hat{\omega})\mu A^T[C - AX(k-1) - X(k-1)B]$

Step 5: $X_2(k) = X(k-1) + \hat{\omega}\mu[C - AX(k-1) - X(k-1)B]B^T$

Step 6: End

The RGI algorithm has been proved to be convergent when

$$0 < \mu < \frac{1}{\hat{\omega}(1-\hat{\omega})(\lambda_1+\lambda_2+\lambda_3)},$$

where $\lambda_1 = \lambda_{\max}(AA^T)$, $\lambda_2 = \lambda_{\max}(B^TB)$, and $\lambda_3 = \sigma_{\max}(BA^T)$ denotes the largest singular value of matrix BA^T.

By using the information of $X_1(k)$ to update $X(k-1)$, Xie et al. [25] proposed the following accelerated gradient based iterative (AGBI) algorithm.

Algorithm 3: The accelerated gradient based iterative (AGBI) algorithm

Step 1: Given two initial approximate solution block vectors $X_1(k-1)$ and $X_2(k-1)$

Step 2: For $k = 1, 2, \cdots$, until converges, do:

Step 3: $X(k-1) = (1-\bar{\omega})X_1(k-1) + \bar{\omega}X_2(k-1)$

Step 4: $X_1(k) = X(k-1) + \bar{\omega}\mu A^T[C - AX(k-1) - X(k-1)B]$

Step 5: $X(k-1) = (1-\bar{\omega})X_1(k) + \bar{\omega}X_2(k-1)$

Step 6: $X_2(k) = X(k-1) + (1-\bar{\omega})\mu[C - AX(k-1) - X(k-1)B]B^T$

Step 7: End

From Theorem 3.4 [25], we conclude that the AGBI algorithm will be convergent when

$$0 < \mu < \min\left\{\frac{2}{\bar{\omega}\|A\|^2}, \frac{2}{(1-\bar{\omega})\|B\|^2}\right\}.$$

In order to reduce computational cost and save storage space in GI algorithm, the authors in [13] presented the Jacobi-gradient iterative (JGI) algorithm to solve the Lyapunov matrix equation

$$AX + XA^T = Q,$$

and obtained its convergent theorem. Now, we will use the JGI algorithm to solve Eq.(1).

Let $A = D_1 + F_1$ and $B = D_2 + F_2$, where D_1, D_2 are the diagonal parts of A and B, respectively. Based on (3), we have

$$D_1X = C - XB - F_1X,$$

$$XD_2 = C - AX - XF_2.$$

Similar to the GI algorithm, the Jacobi-gradient iterative algorithm can be expressed as follows:

Algorithm 4: The Jacobi-gradient iterative (JGI) algorithm

Step 1: Given any two initial approximate solution block vectors $X_1(0), X_2(0)$, and then
$X(0) = [X_1(0) + X_2(0)]/2$

Step 2: For $k = 1, 2, \cdots$, until converges, do:

Step 3: $X_1(k) = X(k-1) + \mu D_1[C - AX(k-1) - X(k-1)B]$

Step 4: $X_2(k) = X(k-1) + \mu[C - AX(k-1) - X(k-1)B]D_2$

Step 5: $X(k) = [X_1(k) + X_2(k)]/2$

Step 6: End

3. The Accelerated Jacobi-Gradient Based Iterative (AJGI) Algorithm

In Algorithm 4, we use the information of $X(k-1)$ instead of $X_2(k-1)$ to update $X(k-1)$, and introduce two relaxation factors ω_1 and ω_2, then obtain the following accelerated Jacobi-gradient based iterative (AJGI) algorithm.

Algorithm 5: The accelerated Jacobi-gradient based iterative (AJGI) algorithm

Step 1: Given two initial approximate solution block vectors $X_1(k-1)$ and $X_2(k-1)$

Step 2: For $k = 1, 2, \cdots$, until converges, do:

Step 3: $X(k-1) = [X_1(k-1) + X_2(k-1)]/2$

Step 4: $X_1(k) = X(k-1) + (1-\omega_1)\mu D_1[C - AX(k-1) - X(k-1)B]$

Step 5: $\hat{X}(k-1) = (1-\omega_2)X(k-1) + \omega_2 X_1(k)$
Step 6: $X_2(k) = \hat{X}(k-1) + \omega_1\mu[C - A\hat{X}(k-1) - \hat{X}(k-1)B]D_2$
Step 7: End

Theorem 3.1. *If the Sylvester matrix equation (1) is consistent and has a unique solution X, then the iterative sequences $X(k)$ generated by Algorithm 4 converge to X, i.e., $\lim_{k\to\infty} X(k) = X$; or the error $X(k) - X$ converge to zero for any initial value $X(0)$ when μ satisfies*

$$\|I - \mu D_1 A\|_2 + \|I - \mu B D_2\|_2 + \mu\tau_1\sigma_2 + \mu\tau_2\sigma_1 < 2,$$

where $\tau_1 = \|D_1\|_2, \tau_2 = \|D_2\|_2, \sigma_1 = \|A\|_2$, and $\sigma_2 = \|B\|_2$, respectively.

Proof. Define the error matrices

$$\tilde{X}(k) := X(k) - X,$$
$$\tilde{X}_1(k) := X_1(k) - X,\ \tilde{X}_2(k) := X_2(k) - X$$

According to Algorithm 4, we can obtain

$$\begin{aligned}\tilde{X}(k) &= X_1(k)/2 + X_2(k)/2 - X\\ &= \tilde{X}_1(k)/2 + \tilde{X}_2(k)/2,\end{aligned}$$

$$\begin{aligned}\tilde{X}_1(k) &= X(k-1) + \mu D_1[C - AX(k-1) - X(k-1)B] - X\\ &= \tilde{X}(k-1) - \mu D_1 A\tilde{X}(k-1) - \mu D_1\tilde{X}(k-1)B,\end{aligned}$$

$$\begin{aligned}\tilde{X}_2(k) &= X(k-1) + \mu[C - AX(k-1) - X(k-1)B]D_2 - X\\ &= \tilde{X}(k-1) - \mu[A\tilde{X}(k-1) + \tilde{X}(k-1)B]D_2\\ &= \tilde{X}(k-1) - \mu A\tilde{X}(k-1)D_2 - \mu\tilde{X}(k-1)BD_2.\end{aligned}$$

then

$$\begin{aligned}\tilde{X}(k) &= \tilde{X}_1(k)/2 + \tilde{X}_2(k)/2\\ &= \tilde{X}(k-1) - \tfrac{1}{2}\mu D_1 A\tilde{X}(k-1) - \tfrac{1}{2}\mu D_1\tilde{X}(k-1)B\\ &\quad - \tfrac{1}{2}\mu A\tilde{X}(k-1)D_2 - \tfrac{1}{2}\mu\tilde{X}(k-1)BD_2.\end{aligned}$$

Taking the 2-norm of $\tilde{X}(k)$, then we have

$$\begin{aligned}\|\tilde{X}(k)\|_2 &\le \tfrac{1}{2}(\|I - \mu D_1 A\|_2 + \|I - \mu B D_2\|_2\\ &\quad + \mu\|D_1\|_2\|B\|_2 + \mu\|D_2\|_2\|A\|_2)\|\tilde{X}(k-1)\|_2\\ &= \tfrac{1}{2}(\|I - \mu D_1 A\|_2 + \|I - \mu B D_2\|_2 + \mu\tau_1\sigma_2 + \mu\tau_2\sigma_1)\|\tilde{X}(k-1)\|_2,\end{aligned}$$

where $\tau_1 = \|D_1\|_2 = \max\limits_{1\le i\le m}|d_1(ii)|$, $\tau_2 = \|D_2\|_2 = \max\limits_{1\le i\le n}|d_2(ii)|$, σ_1 and σ_2 are the largest singular values of A and B, respectively.

Let

$$q = \frac{1}{2}(\|I - \mu D_1 A\|_2 + \|I - \mu B D_2\|_2 + \mu\tau_1\sigma_2 + \mu\tau_2\sigma_1),$$

we get

$$\|\tilde{X}(k)\|_2 \le q\|\tilde{X}(k-1)\|_2 \le \cdots \le q^k\|\tilde{X}(0)\|_2.$$

Therefore, if

$$\|I - \mu D_1 A\|_2 + \|I - \mu B D_2\|_2 + \mu\tau_1\sigma_2 + \mu\tau_2\sigma_1 < 2, \tag{4}$$

then

$$\tilde{X}(k) \to 0 \text{ as } k \to \infty.$$

This completes the proof of Theorem 3.1. □

Theorem 3.2. *If the Sylvester matrix equation (1) is consistent and has a unique solution X, then the iterative sequences $X(k)$ generated by Algorithm 5 converge to X, i.e., $\lim_{k\to\infty} X(k) = X$; or the error $X(k) - X$ converge to zero for any initial value $X(0)$ when μ satisfies*

$$\|I - (1-\omega_1)\mu D_1 A\|_2 + (1-\omega_1)\mu\tau_1\sigma_2 + \tau\|I - \omega_1\mu B D_2\|_2 + \omega_1\mu\tau\tau_2\sigma_1 < 2$$

and $0 < \omega_1 < 1,\ \omega_2 > 0$.

Proof. Define the error matrices

$$\bar{X}(k) := X(k) - X,\ \bar{\hat{X}}(k) := \hat{X}(k) - X,$$
$$\bar{X}_1(k) := X_1(k) - X,\ \bar{X}_2(k) := X_2(k) - X$$

From Algorithm 5 it is easy to get

$$\begin{aligned}\bar{X}(k) &= \tfrac{1}{2}X_1(k) + \tfrac{1}{2}X_2(k) - X\\ &= \tfrac{1}{2}\bar{X}_1(k) + \tfrac{1}{2}\bar{X}_2(k),\end{aligned}$$

$$\begin{aligned}\bar{X}_1(k) &= X(k-1) + (1-\omega_1)\mu D_1[C - AX(k-1) - X(k-1)B] - X\\ &= \bar{X}(k-1) - (1-\omega_1)\mu D_1 A\bar{X}(k-1) - (1-\omega_1)\mu D_1\bar{X}(k-1)B,\end{aligned}$$

$$\begin{aligned}\bar{X}_2(k) &= \hat{X}(k-1) + \omega_1\mu[C - A\hat{X}(k-1) - \hat{X}(k-1)B]D_2 - X\\ &= \bar{\hat{X}}(k-1) - \omega_1\mu[A\bar{\hat{X}}(k-1) + \bar{\hat{X}}(k-1)B]D_2\\ &= \bar{\hat{X}}(k-1) - \omega_1\mu A\bar{\hat{X}}(k-1)D_2 - \omega_1\mu\bar{\hat{X}}(k-1)BD_2.\end{aligned}$$

Since

$$\bar{\hat{X}}(k-1) = \bar{X}(k-1) - \mu\omega_2(1-\omega_1)D_1A\bar{X}(k-1) - \mu\omega_2(1-\omega_1)D_1\bar{X}(k-1)B,$$

then

$$\|\bar{\hat{X}}(k-1)\|_2 \le (\|I - \mu\omega_2(1-\omega_1)D_1A\|_2 + \mu\omega_2(1-\omega_1)\tau_1\sigma_2)\|\bar{X}(k-1)\|_2 = \tau\|\bar{X}(k-1)\|_2$$

with $0 < \omega_1 < 1,\ \omega_2 > 0$.

Since $0 < \omega_1 < 1$, then we have

$$\begin{aligned}\|\bar{X}(k)\|_2 &= \|\tfrac{1}{2}\bar{X}_1(k) + \tfrac{1}{2}\bar{X}_2(k)\|_2\\ &\le \tfrac{1}{2}\|\bar{X}_1(k)\|_2 + \tfrac{1}{2}\|\bar{X}_2(k)\|_2\\ &\le \tfrac{1}{2}(\|I - (1-\omega_1)\mu D_1A\|_2 + (1-\omega_1)\mu\tau_1\sigma_2)\|\bar{X}(k-1)\|_2\\ &+\tfrac{1}{2}(\tau\|I - \omega_1\mu BD_2\|_2 + \omega_1\mu\tau\tau_2\sigma_1)\|\bar{X}(k-1)\|_2.\end{aligned}$$

Let

$$p = \frac{1}{2}(\|I - (1-\omega_1)\mu D_1A\|_2 + (1-\omega_1)\mu\tau_1\sigma_2) + \frac{1}{2}(\tau\|I - \omega_1\mu BD_2\|_2 + \omega_1\mu\tau\tau_2\sigma_1),$$

Then we have

$$\|\bar{X}(k)\|_2 \le p\|\bar{X}(k-1)\|_2 \le \cdots \le p^k\|\bar{X}(0)\|_2.$$

Therefore, if

$$\|I - (1-\omega_1)\mu D_1A\|_2 + (1-\omega_1)\mu\tau_1\sigma_2 + \tau\|I - \omega_1\mu BD_2\|_2 + \omega_1\mu\tau\tau_2\sigma_1 < 2, \tag{5}$$

then

$$\bar{X}(k) \to 0 \text{ as } k \to \infty.$$

This completes the proof of Theorem 3.2 □

Similar to Algorithm 3, we can get another accelerated Jacobi-gradient based iterative algorithm.
Algorithm 6:
Step 1: Given two initial approximate solution block vectors $X_1(k-1)$ and $X_2(k-1)$
Step 2: For $k = 1, 2, \cdots$, until converges, do:
Step 3: $X(k-1) = \omega_1 X_1(k-1) + (1-\omega_1)X_2(k-1)$
Step 4: $X_1(k) = X(k-1) + (1-\omega_1)\mu A^T[C - AX(k-1) - X(k-1)B]$
Step 5: $\dot{X}(k-1) = (1-\omega_2)X(k-1) + \omega_2 X_1(k)$
Step 6: $X_2(k) = \dot{X}(k-1) + \omega_1\mu[C - A\dot{X}(k-1) - \dot{X}(k-1)B]B^T$
Step 7: End

Theorem 3.3. *If the Sylvester matrix equation (1.1) is consistent and has a unique solution X, then the iterative sequences $X(k)$ generated by Algorithm 6 converge to X, i.e., $\lim_{k\to\infty} X(k) = X$; or the error $X(k) - X$ converge to zero for any initial value $X(0)$ when $0 < \omega_1 < 1$, $\omega_2 > 0$ and μ satisfies*

$$\omega_1(\|I - (1-\omega_1)\mu D_1 A\|_2 + (1-\omega_1)\mu\tau_1\sigma_2) + \tau(1-\omega_1)(\|I - \omega_1\mu B D_2\|_2 + \omega_1\mu\tau_2\sigma_1) < 1. \tag{6}$$

Proof. This proof is similar to that of Theorem 3.2. □

Remark 3.4. *In Theorems 3.1, 3.2 and 3.3, we can choose relative large u, ω_1 not satisfying the inequalities (4), (5) and (6), but the JGI and AJGI algorithms also converge to the true solutions.This is because that the inequalities are just the sufficient conditions but not the necessary conditions, and we magnify the inequalities too large during the proofs. However, it is difficult to find the optimal values of the parameters μ, ω_1 and ω_2.*

Remark 3.5. *In fact, the AJGI algorithm doesn't increase the computational cost obviously compared with the JGI algorithm, since the matrix $\hat{X}(k-1)$ in Algorithm 5 can been written equivalently as follows:*

$$\hat{X}(k-1) = X(k-1) + \omega_2(1-\omega_1)\mu D_1[C - AX(k-1) - X(k-1)B],$$

where $X(k-1)$ and $D_1[C - AX(k-1) - X(k-1)B]$ have been computed in previous steps, so $\hat{X}(k-1)$ can be obtained at a fraction of the cost. We can get the similar conclusion from Algorithm 6.

4. Numerical examples

In this section, three examples are given to illustrate the effectiveness of the AJGI algorithm. The numerical experiments are performed in Matlab R2010 on an Intel dual core processor (3.20 GHZ, 4 GB RAM). We use three iteration parameters to test the five algorithms in this paper with the iteration step (denoted as IT), the computing time in seconds (denoted as CPU) and the relative residual (denoted as e_k).

Example 4.1. *[6] Consider the following sylvester matrix equation*

$$AX + XB = C$$

with

$$A = \begin{bmatrix} 1 & 1 \\ 2 & -4 \end{bmatrix},\ B = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix},\ C = \begin{bmatrix} 3 & 10 \\ -12 & -8 \end{bmatrix}.$$

In Fig.1, the convergence curves by five iterative solvers are recorded. From Fig.1 and Table 1, we find that the AJGI algorithm converges faster than GI, JGI, RGI and AGBI algorithms, respectively, and the AJGI algorithm outperforms other four algorithms in both iteration step and CPU time.

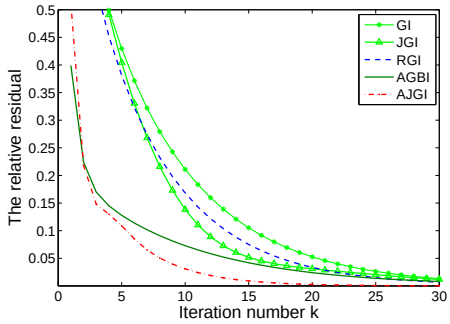

Figure 1: Comparison of convergence curves for GI, JGI, RGI, AGBI and AJGI algorithms

Table 1: Numerical results of Example 1.

Method	IT	CPU	e_k
GI	215	3.424×10^{-3}	$9.3416e-014$
JGI	256	1.862×10^{-3}	$9.1605e-014$
RGI	185	2.847×10^{-3}	$9.2090e-014$
AGBI	163	3.022×10^{-3}	$8.9345e-014$
AJGI	97	0.922×10^{-3}	$8.2423e-014$

Example 4.2. *[6] In this example, the experimental matrices contain a variable α, we choose $m = n = 30$. From Figs.2, 3 and Table 2 we find that the AJGI algorithm converges faster than GI, JGI, RGI and AGBI algorithms, respectively, and the larger the relaxed factor ω_2 is, the faster the convergence of the AJGI algorithm. However, if ω_2 is too large, the algorithm may diverge. How to choose the optimal convergence factor ω_2 is still an open problem.*

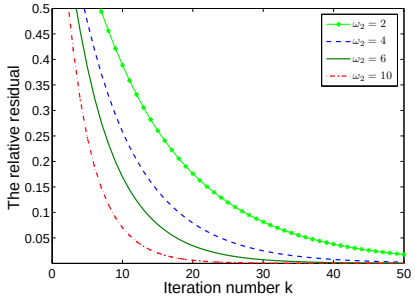

Figure 2: Convergence curves for GI, JGI, RGI, AGBI and AJGI algorithms with $\alpha = 4$

Table 2: Numerical results of Example 2.

Method	IT	CPU	e_k
GI	508	6.437×10^{-2}	$9.7462e-014$
JGI	321	3.757×10^{-2}	$9.4241e-014$
RGI	440	5.640×10^{-2}	$9.6562e-014$
AGBI	401	6.047×10^{-2}	$9.6299e-014$
AJGI	160	2.429×10^{-2}	$8.7991e-014$

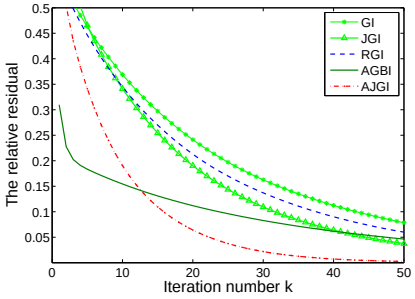

Figure 3: Convergence curves for AJGI algorithm with $\omega_2 = 2, 4, 6, 10$ and $\alpha = 2.5$

Example 4.3. *Suppose that $AX + XB = C$, where A, B and C are 60×60 matrices and generated in Matlab as follows:*
rand ('state',0);
A=triu(rand(m,n),1)+diag(α+diag(rand(m)))′;
*$B = A'$; C=rand(m,m)+eye(m)*2 ; $C = C + C'$. In this example, we choose $\alpha = 6$, and the systems is very ill-conditioned. Here, we will also compare the AJGI algorithm with the SSJGI algorithm [17]. The SSJGI algorithm improves the JGI algorithm by introducing a parameter θ, and it converges faster than the JGI algorithm by choosing appropriate values of θ.*

According to Figs.4, 5 and Table 3, we can see that the AJGI algorithm converges faster than GI, JGI, RGI, AGBI, AJGI and SSJGI algorithms, respectively.

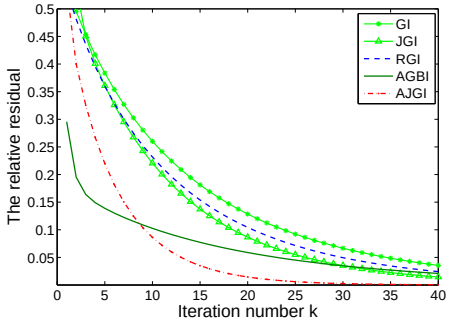

Figure 4: Convergence curves for GI, JGI, RGI, AGBI and AJGI algorithms

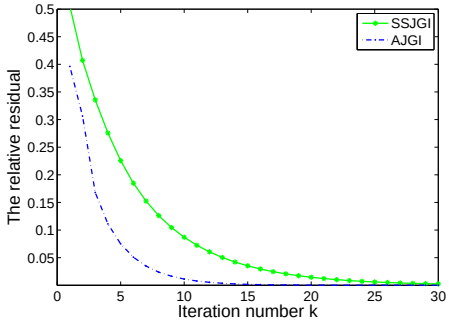

Figure 5: Convergence curves for AJGI and SSJGI algorithms with $\theta = 12$

Table 3: Numerical results of Example 3.

Method	IT	CPU	e_k
GI	847	0.360472	$9.8006e-014$
JGI	499	0.206684	$9.4557e-014$
RGI	735	0.319426	$9.6559e-014$
AGBI	397	0.194878	$9.4300e-014$
SSJGI	161	0.067995	$9.4300e-014$
AJGI	75	0.036669	$6.5804e-014$

5. Conclusions

In this paper, we propose an accelerated Jacobi-gradient based iterative (AJGI) algorithm for solving Sylvester matrix equation $AX + XB = C$, and show that the iterative solution converges to the exact solution for any initial value under certain assumptions. Moreover, the AJGI algorithm can be used to determine the iterative solutions of nonlinear matrix equations, e.g., the Riccati equations. Finally, three numerical examples are given to illustrate that the AJGI algorithm outperforms other five algorithms mentioned in this paper in both iteration step and CPU time.

Acknowledgements

The authors are grateful to thank the anonymous referee for their recommendations and valuable suggestions and Professor Dijana Mosic for the communication.

References

[1] A.Andrew, Eigenvectors of certain matrices,Linear Algebra Appl. 7(2) (1973) 157-162.
[2] Z.Z.Bai, On Hermitian and skew-Hermitian splitting iteration methods for continuous Sylvester equations, J. Comput. Math. 29(2) (2011) 185-198.
[3] R. Bartels, G.W. Stewart, Solution of the matrix equation AX + XB = C, Comm. ACM. 15(9) (1972) 820-826.
[4] D. Calvetti, L. Reichel, Application of ADI iterative methods to the restoration of noisy images, SIAM J. Matrix Anal. Appl. 17(1) (1996)165-186.
[5] W.Chen, X.Wang, T.Zhong,The structure of weighting coefficient matrices of Harmonic differential quadrature and its application,Commun.Numer.Methods Eng. 12(8) (1996) 455-460.
[6] F.Ding, T.Chen, Gradient based iterative algorithms for solving a class of matrix equations, IEEE Trans.Autom.Control. 50(8) (2005)1216-1221.
[7] F. Ding, T. Chen, Hierarchical gradient-based identi cation of multivariable discretetime systems, Automatica. 41(2) (2005) 315-325.
[8] F. Ding, T. Chen, Hierarchical least squares identi cation methods for multivarible systems, IEEE Trans. Autom. Control. AC-50(3)(2005) 397-402.
[9] F. Ding, T.Chen, Iterative least squares solutions of coupled Sylvester matrix equations, Syst.Control.Lett. 54(2) (2005) 95-107.
[10] F. Ding, P.X. Liu, J. Ding, Iterative solutions of the generalized Sylvester matrix equation by using the hierarchical identification principle, Appl. Math.Comput. 197(1) (2008) 41-50.
[11] F.Ding, X.H.Wang, Q.J.Chen, Y.S.Xiao, Recursive least squares parameter estimation for a class of output nonlinear systems based on the model decomposition, Circuits Syst. Signal Process. 35 (2016). doi: 10.1007/s00034-015-0190-6.
[12] F.Ding, H.Zhang, Gradient-based iterative algorithm for a class of the coupled matrix equations related to control systems, Iet.Control.Theory.A. 8(15) (2014) 1588-1595.
[13] W. Fan, C. Gu, Z. Tian, Jacobi-gradient iterative algorithms for Sylvester matrix equations, Proceedings of 14th Conference of the International Linear Algebra Society, Shanghai University, Shanghai, China, July 16-20, 2007.
[14] G.H. Golub, S. Nash, C.F. Van Loan, A HessenbergCSchur method for the matrix equation AX + XB = C, IEEE Trans. Automat. Control. 24(6) (1979) 909-913.
[15] C. Gu, H. Xue, A shift-splitting hierarchical identi cation method for solving Lyapunov matrix equations, Linear Alg. Appl. 430(5-6)(2009) 1517-1530.
[16] D.Y. Hu, L.Reichel, Krylov-subspace methods for the Sylvester equation, Linear Alg. Appl. 172(92) (1992), 283-313.
[17] S.K.Li, T.Z.Huang, A shift-splitting Jacobi-gradient algorithm for Lyapunov matrix equations arising from control theory, J. Comput. Anal. Appl. 13(7)(2011) 1246-1257.
[18] Q.Niu, X.Wang, L.Z.Lu, A relaxed gradient based algorithm for solving Sylvester equations, Asian Journal of Control. 13 (3) (2011) 461-464.

[19] R.A. Smith, Matrix equation XA + BX = C, SIAM J. Appl. Math. 16(1) (1968) 198-201.
[20] Z.L.Tian, C.Q.Gu, A numerical algorithm for Lyapinov equations, Appl. Math. Comput. 202(1) (2008) 44-53.
[21] E.L. Wachspress, Iterative solution of the Lyapunov matrix equation, Appl. Math. Lett. 1(1) (1988) 87-90.
[22] Y.J.Wang, F.Ding, Iterative estimation for a nonlinear IIR filter with moving average noise by means of the data filtering technique, IMA J. Math. Control Inf. (2016). doi:10.1093/imamci/dnv067.
[23] X.Wang,W.W.Li,L.Z.Mao,On positive-definite and skew-Hermitian splitting iteration methods for continuous Sylvester equation AX+XB=C,Comput.Math.Appl. 66(11) (2013) 2352-2361.
[24] L.Xie,Y.J.Liu,H.Z.Yang, Gradient based and least squares based iterative algorithms for matrix equations $AXB + CX^TD = F$, Appl. Math. Comput. 217 (5) (2010) 2191-2199.
[25] Y.J.Xie, C.F.Ma, The accelerated gradient based iterative algorithm for solving a class of generalized Sylvester-transpose matrix equation, Appl. Math. Comput. 273(C) (2016) 1257-1269.
[26] H.M.Zhang, F.Ding, A property of the eigenvalues of the symmetric positive definite matrix and the iterative algorithm for coupled Sylvester matrix equations, J. Franklin Inst. 351(1)(2014) 340-357.
[27] Q.Q.Zheng, Ch.F.Ma, On normal and skew-Hermitian splitting iteration methods for large sparse continuous Sylvester equations, J.Comput.Appl.Math. 268(1) (2014)145-154.
[28] D.M.Zhou, G.L.Chen, Q.Y.Cai, On modifeid HSS iteration methods for continuous Sylvester eqautions, Appl. Math. Comput. 263(C)(2015) 84-93.