

## Research Article

# A Gradient Based Iterative Solutions for Sylvester Tensor Equations

Zhen Chen<sup>1</sup> and Linzhang Lu<sup>1,2</sup>

<sup>1</sup> School of Mathematics and Computer Science, Guizhou Normal University, Guiyang 550001, China

<sup>2</sup> School of Mathematical Sciences, Xiamen University, Xiamen 361005, China

Correspondence should be addressed to Zhen Chen; [zchen@gznu.edu.cn](mailto:zchen@gznu.edu.cn)

Received 27 December 2012; Accepted 13 February 2013

Academic Editor: Mohammad Younis

Copyright © 2013 Z. Chen and L. Lu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper is concerned with the numerical solution of the Sylvester tensor equation, which includes the Sylvester matrix equation as special case. By applying hierarchical identification principle proposed by Ding and Chen, 2005, and by using tensor arithmetic concepts, an iterative algorithm and its modification are established to solve the Sylvester tensor equation. Convergence analysis indicates that the iterative solutions always converge to the exact solution for arbitrary initial value. Finally, some examples are provided to show that the proposed algorithms are effective.

## 1. Introduction

This paper is concerned with the solution of the following equation:

$$\mathcal{X} \times_1 A + \mathcal{X} \times_2 B + \mathcal{X} \times_3 C = \mathcal{D}, \quad (1)$$

where matrices  $A \in R^{m \times m}$ ,  $B \in R^{n \times n}$ , and  $C \in R^{l \times l}$ , and tensor  $\mathcal{D} \in R^{m \times n \times l}$ . The properties of the operators  $\times_1$ ,  $\times_2$ , and  $\times_3$  are described in detail in Section 2. When  $\mathcal{X}$  is a 2-mode tensor, that is, a matrix, (1) reduces to

$$AX + XB^T = D. \quad (2)$$

It is just a Sylvester matrix equation, which has received much attention, often motivated by applications in control theory, model reduction, signal processing, filtering, and system identification [1–7]. So we call (1) a Sylvester tensor equation in this paper.

The simplest direct approach for solving the matrix equation (2) is Kronecker product method [8] in which the huge memory is needed, for example, if  $m = n = 100$ , the dimension of the new coefficient matrix is 10000. Alternative direct method is to firstly transform coefficient matrices into

special forms, such as Schur canonical form [9], Hessenberg-Schur form [1], and [10–12], and then to solve another matrix equation which may be readily computed. Such kinds of methods have been widely adopted. For example, 3D Schur decomposition was developed to solve the Sylvester tensor equation (1) (for more details see [13]). However, these methods require the transformations of coefficient matrices, which need considerable cost.

Iterative algorithms for solving matrix equations have been studied extensively, for example, [14–21]. For example, Ding and Chen proposed a few simple iterative methods for matrix equation in [22–24]. The methods, resembling the classical Jacobi and Gaussian iterations for linear systems, are easy to implement and cost little per step. Gradient based iterative algorithms and least squares based iterative algorithms have also been proposed in [25–27] for (coupled) matrix equations, which include the Sylvester matrix equations and Lyapunov matrix equations as a special form. However, to the best knowledge of the authors, there is hardly any iterative method for the Sylvester tensor equation in references up to now.

In this paper, we first extend the gradient based iterative algorithm for Sylvester matrix equation (2) to solve the

Sylvester matrix equation (1) based on previous work in [22]. The basic idea of such an algorithm is to regard the unknown tensor as the system parameter tensor by applying hierarchical identification principle. Then, we present a more efficient modified iterative method. The convergence properties of the two algorithms are proved. Numerical examples are given to show that the two iterations are effective.

The rest of this paper is organized as follows. In Section 2, we describe tensor notations and common operations used throughout the paper. An extension and modification of the gradient based iterative algorithms to the Sylvester tensor equation (1) are presented in Section 3. The convergence of the algorithms are analyzed in Section 4. Section 5 contains some numerical experiments to illustrate the theoretical results in this paper. Finally, some conclusions are outlined in Section 6.

## 2. Tensor Notations and Basic Operations

In this section, we briefly review some concepts and notations that are used throughout the paper. Matrices are denoted by capital letter, for example,  $A$ ,  $B$ , and  $C$ . Tensors are denoted by Euler script letter, for example,  $\mathcal{X}$ ,  $\mathcal{Y}$ , and  $\mathcal{Z}$ . The  $n \times n$  identity matrix is denoted by  $I_n$ . The order of a tensor is the number of dimensions, also known as ways or modes. As a special case, a vector is 1-mode tensor and a matrix is 2-mode tensor. The operator  $\text{vec}(\cdot)$  stacks the columns of a matrix to form a vector.  $\|\cdot\|$  denotes the Frobenius norm unless otherwise stated.

**2.1. Tensor-Matrix Multiplication.** An important operation for a tensor is the tensor-matrix multiplication [28–30]. The 1-mode product of a tensor  $\mathcal{X} \in R^{I \times J \times K}$  by a matrix  $A \in R^{P \times I}$ , denoted by  $\mathcal{X} \times_1 A$ , is a  $P \times J \times K$  tensor in which the entries are given by

$$(\mathcal{X} \times_1 A)(p, j, k) = \sum_{i=1}^I x_{ijk} a_{pi}. \quad (3)$$

Similarly, 2-mode multiplication of a tensor by a matrix  $B \in R^{Q \times J}$

$$(\mathcal{X} \times_2 B)(i, q, k) = \sum_{j=1}^J x_{ijk} b_{qj}. \quad (4)$$

3-mode multiplication is analogous and is omitted here.

It follows immediately from the definition that  $n$ -mode and  $m$ -mode multiplication commutes if  $m \neq n$ :

$$\mathcal{X} \times_m A \times_n B = \mathcal{X} \times_n B \times_m A. \quad (5)$$

If the modes are the same, then

$$\mathcal{X} \times_n A \times_n B = \mathcal{X} \times_n (BA). \quad (6)$$

**2.2. Inner Product and Tensor Norm.** The inner product of two tensors  $\mathcal{X}, \mathcal{Y} \in R^{I \times J \times K}$  is defined by

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K x_{ijk} y_{ijk}, \quad (7)$$

and the norm induced by this inner product is

$$\|\mathcal{X}\| = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}. \quad (8)$$

For two tensors  $\mathcal{X}$  and  $\mathcal{Y}$ , the computation of  $\|\mathcal{X} - \mathcal{Y}\|$  can be simplified as follows:

$$\|\mathcal{X} - \mathcal{Y}\|^2 = \|\mathcal{X}\|^2 - 2\langle \mathcal{X}, \mathcal{Y} \rangle + \|\mathcal{Y}\|^2. \quad (9)$$

Moreover,  $n$ -mode multiplication commutes with respect to the inner product, that is,

$$\langle \mathcal{X}, \mathcal{Y} \times_n A \rangle = \langle \mathcal{X} \times_n A^T, \mathcal{Y} \rangle. \quad (10)$$

Finally, the tensor norm of  $n$ -mode multiplication is mutually consistent, that is,

$$\|\mathcal{X} \times_n A\| \leq \|A\|_2 \|\mathcal{X}\|. \quad (11)$$

## 3. Iterative Algorithms

In this section, we apply the tensor arithmetic concepts and the hierarchical identification principle to solve the Sylvester tensor equation (1).

Firstly, we consider the existence and uniqueness of the solution of (1). By using the Kronecker product, we have

$$\begin{aligned} & (I_l \otimes I_n \otimes A + I_l \otimes B \otimes I_m + C \otimes I_n \otimes I_m) \text{vec}(\mathcal{X}) \\ & = \text{vec}(\mathcal{D}), \end{aligned} \quad (12)$$

which is equivalent to (1). This vectorized representation immediately leads to the following result.

**Lemma 1.** Equation (1) has a unique solution if and only if  $\lambda_i(A) + \lambda_j(B) + \lambda_k(C) \neq 0$  for any  $i, j$ , and  $k$ , where  $\lambda_i(A)$ ,  $\lambda_j(B)$  and  $\lambda_k(C)$  represent the eigenvalues of  $A$ ,  $B$ , and  $C$ , respectively.

In the case of the Sylvester matrix equation (2), this corresponds to the well-known condition  $\lambda_i(A) + \lambda_j(B) \neq 0$ , for any  $i$  and  $j$ .

In order to drive the iterative algorithms for solving (1), we need to introduce the following residual tensors:

$$\begin{aligned} \mathcal{E}_1 & := \mathcal{D} - \mathcal{X} \times_2 B - \mathcal{X} \times_3 C, \\ \mathcal{E}_2 & := \mathcal{D} - \mathcal{X} \times_3 C - \mathcal{X} \times_1 A, \\ \mathcal{E}_3 & := \mathcal{D} - \mathcal{X} \times_1 A - \mathcal{X} \times_2 B. \end{aligned} \quad (13)$$

According to the hierarchical identification principle [22], (1) is decomposed into three subsystems:

$$\mathcal{X} \times_1 A = \mathcal{E}_1, \quad \mathcal{X} \times_2 B = \mathcal{E}_2, \quad \mathcal{X} \times_3 C = \mathcal{E}_3. \quad (14)$$

Consider the following least squares problem:

$$\begin{aligned} J_1(\mathcal{X}) & = \frac{1}{2} \|\mathcal{X} \times_1 A - \mathcal{E}_1\|^2, \\ J_2(\mathcal{X}) & = \frac{1}{2} \|\mathcal{X} \times_2 B - \mathcal{E}_2\|^2, \\ J_3(\mathcal{X}) & = \frac{1}{2} \|\mathcal{X} \times_3 C - \mathcal{E}_3\|^2. \end{aligned} \quad (15)$$

Note that the derivatives of  $J_i$  ( $i = 1, 2, 3$ ) with respect to  $\mathcal{X}$  are derived in [30] and are given by

$$\begin{aligned} \frac{J_1(\mathcal{X})}{\partial \mathcal{X}} &= (\mathcal{X} \times_1 A - \mathcal{E}_1) \times_1 A^T, \\ \frac{J_2(\mathcal{X})}{\partial \mathcal{X}} &= (\mathcal{X} \times_2 B - \mathcal{E}_2) \times_2 B^T, \\ \frac{J_3(\mathcal{X})}{\partial \mathcal{X}} &= (\mathcal{X} \times_3 C - \mathcal{E}_3) \times_3 C^T. \end{aligned} \quad (16)$$

Minimizing  $J_i(\mathcal{X})$  ( $i = 1, 2, 3$ ) leads to the following recursive equations:

$$\begin{aligned} \mathcal{X}_1^k &= \mathcal{X}_1^{k-1} + \mu (\mathcal{E}_1 - \mathcal{X}_1^{k-1} \times_1 A) \times_1 A^T, \\ \mathcal{X}_2^k &= \mathcal{X}_2^{k-1} + \mu (\mathcal{E}_2 - \mathcal{X}_2^{k-1} \times_2 B) \times_2 B^T, \\ \mathcal{X}_3^k &= \mathcal{X}_3^{k-1} + \mu (\mathcal{E}_3 - \mathcal{X}_3^{k-1} \times_3 C) \times_3 C^T, \end{aligned} \quad (17)$$

where  $\mu$  is iterative step length and is given in the following section.

Substituting (13) into (17) and replacing the unknown variable  $\mathcal{X}$  in (13) with their estimate at time  $(k-1)$ , we have

$$\begin{aligned} \mathcal{X}_1^k &= \mathcal{X}_1^{k-1} + \mu \mathcal{R}_1^{k-1} \times_1 A^T, \\ \mathcal{X}_2^k &= \mathcal{X}_2^{k-1} + \mu \mathcal{R}_2^{k-1} \times_2 B^T, \\ \mathcal{X}_3^k &= \mathcal{X}_3^{k-1} + \mu \mathcal{R}_3^{k-1} \times_3 C^T, \end{aligned} \quad (18)$$

where  $\mathcal{R}_i^k = \mathcal{D} - \mathcal{X}_i^k \times_1 A - \mathcal{X}_i^k \times_2 B - \mathcal{X}_i^k \times_3 C$  ( $i = 1, 2, 3$ ). Taking the average of  $\mathcal{X}_1^k$ ,  $\mathcal{X}_2^k$ , and  $\mathcal{X}_3^k$ , we obtain Algorithm 1.

*Remark 2.* Note that when we compute  $\mathcal{X}_2^k$  in Algorithm 1, the  $\mathcal{X}_1^k$  has been computed. Similarly, when we compute  $\mathcal{X}_3^k$ , the  $\mathcal{X}_1^k$  and  $\mathcal{X}_2^k$  have been computed. Hence, we can fully take advantage of the information of  $\mathcal{X}_i^k$  ( $i = 1, 2, 3$ ) to update the  $\mathcal{X}^{k-1}$  and get a more efficient algorithm as in Algorithm 2.

#### 4. Convergence Analysis

In this section, we will discuss convergence properties of the algorithms in the previous section. The following theorem provides a sufficient condition to guarantee convergence of the Algorithm 1.

**Theorem 3.** *If the Sylvester tensor equation (1) has a unique solution  $\mathcal{X}^*$  and  $0 < \mu < 2/(\|A\|_2^2 + \|B\|_2^2 + \|C\|_2^2)$ , then the iterative solution  $\mathcal{X}^k$  given by the Algorithm 1 converges to  $\mathcal{X}^*$  for arbitrary initial value  $\mathcal{X}^0$ .*

*Proof.* Define the error tensors

$$\bar{\mathcal{X}}^k := \mathcal{X}^k - \mathcal{X}, \quad \bar{\mathcal{X}}_i^k := \mathcal{X}_i^k - \mathcal{X}, \quad (i = 1, 2, 3), \quad (19)$$

$$\mathcal{U}^k := \bar{\mathcal{X}}^{k-1} \times_1 A, \quad \mathcal{V}^k := \bar{\mathcal{X}}^{k-1} \times_2 B, \quad (20)$$

$$\mathcal{W}^k := \bar{\mathcal{X}}^{k-1} \times_3 C.$$

Using (1) and Algorithm 1, the following equalities are trivial:

$$\begin{aligned} \bar{\mathcal{X}}_1^k &= \bar{\mathcal{X}}^{k-1} - \mu \mathcal{L}^k \times_1 A^T, \\ \bar{\mathcal{X}}_2^k &= \bar{\mathcal{X}}^{k-1} - \mu \mathcal{L}^k \times_2 B^T, \\ \bar{\mathcal{X}}_3^k &= \bar{\mathcal{X}}^{k-1} - \mu \mathcal{L}^k \times_3 C^T, \end{aligned} \quad (21)$$

where  $\mathcal{L}^k = \mathcal{U}^k + \mathcal{V}^k + \mathcal{W}^k$ . Taking norm on both sides of (21), and using (9)–(11), we have

$$\begin{aligned} \|\bar{\mathcal{X}}_1^k\|^2 &\leq \|\bar{\mathcal{X}}^{k-1}\|^2 + \mu^2 \|A\|_2^2 \|\mathcal{L}^k\|^2 - 2\mu \langle \mathcal{U}^k, \mathcal{L}^k \rangle, \\ \|\bar{\mathcal{X}}_2^k\|^2 &\leq \|\bar{\mathcal{X}}^{k-1}\|^2 + \mu^2 \|B\|_2^2 \|\mathcal{L}^k\|^2 - 2\mu \langle \mathcal{V}^k, \mathcal{L}^k \rangle, \\ \|\bar{\mathcal{X}}_3^k\|^2 &\leq \|\bar{\mathcal{X}}^{k-1}\|^2 + \mu^2 \|C\|_2^2 \|\mathcal{L}^k\|^2 - 2\mu \langle \mathcal{W}^k, \mathcal{L}^k \rangle. \end{aligned} \quad (22)$$

According to (19) and (22), we have

$$\begin{aligned} \|\bar{\mathcal{X}}^k\|^2 &\leq \frac{\|\bar{\mathcal{X}}_1^k\|^2 + \|\bar{\mathcal{X}}_2^k\|^2 + \|\bar{\mathcal{X}}_3^k\|^2}{3} \\ &\leq \|\bar{\mathcal{X}}^{k-1}\|^2 + \frac{\mu^2}{3} \|\mathcal{L}^k\|^2 (\|A\|_2^2 + \|B\|_2^2 + \|C\|_2^2) \\ &\quad - \frac{2\mu}{3} \|\mathcal{L}^k\|^2 \\ &= \|\bar{\mathcal{X}}^{k-1}\|^2 - \frac{\mu}{3} \{2 - \mu (\|A\|_2^2 + \|B\|_2^2 + \|C\|_2^2)\} \|\mathcal{L}^k\|^2 \\ &\leq \|\bar{\mathcal{X}}^0\|^2 - \frac{\mu}{3} \{2 - \mu (\|A\|_2^2 + \|B\|_2^2 + \|C\|_2^2)\} \sum_{i=1}^k \|\mathcal{L}^i\|^2, \end{aligned} \quad (23)$$

that is,

$$\begin{aligned} \|\bar{\mathcal{X}}^k\|^2 &+ \frac{\mu}{3} \{2 - \mu (\|A\|_2^2 + \|B\|_2^2 + \|C\|_2^2)\} \sum_{i=1}^k \|\mathcal{L}^i\|^2 \\ &\leq \|\bar{\mathcal{X}}^0\|^2. \end{aligned} \quad (24)$$

Since  $0 < \mu < 2/(\|A\|_2^2 + \|B\|_2^2 + \|C\|_2^2)$ , we have

$$\sum_{i=1}^{\infty} \|\mathcal{L}^i\|^2 \leq \|\bar{\mathcal{X}}^0\|^2 < \infty. \quad (25)$$

This implies that as  $k \rightarrow \infty$ ,  $\mathcal{L}^k \rightarrow 0$ , that is,

$$\bar{\mathcal{X}}^{k-1} \times_1 A + \bar{\mathcal{X}}^{k-1} \times_2 B + \bar{\mathcal{X}}^{k-1} \times_3 C \rightarrow 0. \quad (26)$$

According to Lemma 1,  $\bar{\mathcal{X}}^k \rightarrow 0$  as  $k \rightarrow \infty$ . Thus, the theorem holds.  $\square$

Based on Theorem 3, we have the following theorem.

**Input:**  $A, B, C, \mathcal{D}$ , and  $\mu$ .  
**Output:**  $\mathcal{X}$ .  
Initialize  $\mathcal{X}^0$ .  
For  $k = 1, 2, \dots, k_{\max}$   
  Compute  $\mathcal{R}^{k-1} = \mathcal{D} - \mathcal{X}^{k-1} \times_1 A - \mathcal{X}^{k-1} \times_2 B - \mathcal{X}^{k-1} \times_3 C$ ,  
   $\mathcal{X}_1^k = \mathcal{X}^{k-1} + \mu \mathcal{R}^{k-1} \times_1 A^T$ ,  
   $\mathcal{X}_2^k = \mathcal{X}^{k-1} + \mu \mathcal{R}^{k-1} \times_2 B^T$ ,  
   $\mathcal{X}_3^k = \mathcal{X}^{k-1} + \mu \mathcal{R}^{k-1} \times_3 C^T$ ,  
   $\mathcal{X}^k = (\mathcal{X}_1^k + \mathcal{X}_2^k + \mathcal{X}_3^k)/3$ .  
End

ALGORITHM 1: The gradient based iterative algorithm (GI).

**Input:**  $A, B, C, \mathcal{D}$ , and  $\kappa$ .  
**Output:**  $\mathcal{X}$ .  
Initialize  $\mathcal{X}^0$ .  
For  $k = 1, 2, \dots, k_{\max}$   
   $\mathcal{X}^k = \mathcal{X}^{k-1} + \kappa(\mathcal{D} - \mathcal{X}^{k-1} \times_1 A - \mathcal{X}^{k-1} \times_2 B - \mathcal{X}^{k-1} \times_3 C) \times_1 A^T$ ,  
   $\mathcal{X}^{k-1} = (\mathcal{X}_1^k + \mathcal{X}_2^{k-1} + \mathcal{X}_3^{k-1})/3$ ,  
   $\mathcal{X}^k = \mathcal{X}^{k-1} + \kappa(\mathcal{D} - \mathcal{X}^{k-1} \times_1 A - \mathcal{X}^{k-1} \times_2 B - \mathcal{X}^{k-1} \times_3 C) \times_2 B^T$ ,  
   $\mathcal{X}^{k-1} = (\mathcal{X}_1^k + \mathcal{X}_2^k + \mathcal{X}_3^{k-1})/3$ ,  
   $\mathcal{X}^k = \mathcal{X}^{k-1} + \kappa(\mathcal{D} - \mathcal{X}^{k-1} \times_1 A - \mathcal{X}^{k-1} \times_2 B - \mathcal{X}^{k-1} \times_3 C) \times_3 C^T$ ,  
   $\mathcal{X}^k = (\mathcal{X}_1^k + \mathcal{X}_2^k + \mathcal{X}_3^k)/3$ .  
End

ALGORITHM 2: The modified gradient based iterative algorithm (MGI).

**Theorem 4.** *If the Sylvester tensor equation (1) has a unique solution  $\mathcal{X}^*$  and  $0 < \kappa < \min\{1/\|A\|_2^2, 1/\|B\|_2^2, 1/\|C\|_2^2\}$ , then the iterative solution  $\mathcal{X}^k$  given by the Algorithm 2 converges to  $\mathcal{X}^*$  for any initial value  $\mathcal{X}^0$ .*

*Proof.* The proof is similar to Theorem 3 and hence omitted.  $\square$

Now we consider the convergence rate of the Algorithm 1. From (19) and (21), it is not difficult to get the following residual error equation:

$$\text{vec}(\widetilde{\mathcal{X}}^k) = \left[ I_{mnl} - \frac{\mu}{3} P \right]^k \text{vec}(\widetilde{\mathcal{X}}^0), \quad (27)$$

where  $P = I_1 \otimes I_n \otimes A^T A + I_1 \otimes B \otimes A^T + C \otimes I_n \otimes A^T + I_1 \otimes B^T \otimes A + I_1 \otimes B^T B \otimes I_m + C^T \otimes I_n \otimes A + C^T \otimes B \otimes I_m + C \otimes B^T \otimes I_m + C^T C \otimes I_n \otimes I_m$ . We observe that the closer the eigenvalues of  $\mu P$  are to 3, the closer the eigenvalues of  $I_{mnl} - (\mu/3)P$  tend to zero, and hence, the faster the residual error  $\widetilde{\mathcal{X}}^k$  converges to zero. So the convergence rate of Algorithm 1 depends on the condition number of the associated system [22]; that is, Algorithm 1 has a fast convergence rate for a small condition number of  $P$ . The analysis of Algorithm 2 is similar; hence, it is omitted here.

## 5. Numerical Experiments

This section gives some examples to illustrate the performance of the proposed algorithms. All the implementations are based on the codes from the MATLAB Tensor Toolbox developed by Bader and Kolda [31, 32].

*Example 5.* Let

$$A = \begin{pmatrix} 3 & 1 \\ -1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix},$$

$$C = \begin{pmatrix} 1 & 0 \\ 1 & -2 \end{pmatrix}, \quad (28)$$

$$\mathcal{D}(:, :, 1) = \begin{pmatrix} 10 & 13 \\ 15 & 11 \end{pmatrix}, \quad \mathcal{D}(:, :, 2) = \begin{pmatrix} 14 & 3 \\ 3 & 0 \end{pmatrix}.$$

Then, the solution of  $\mathcal{X}$  is

$$\mathcal{X}(:, :, 1) = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \quad \mathcal{X}(:, :, 2) = \begin{pmatrix} 4 & 2 \\ 3 & 1 \end{pmatrix}. \quad (29)$$

Taking  $\mathcal{X}^0 = \mathbf{0}$ ,  $\mu = 2.8 * 1/(\|A\|_2^2 + \|B\|_2^2 + \|C\|_2^2) \approx 0.1067$ ,  $\kappa = 2.8 * \min\{1/\|A\|_2^2, 1/\|B\|_2^2, 1/\|C\|_2^2\} \approx 0.2747$ , we apply Algorithms 1 and 2 to compute approximate solution in (1). Since the exact solution is known in this example, we plot the values  $\|\mathcal{X}^k - \mathcal{X}\|/\|\mathcal{X}\|$  to indicate the convergence of the

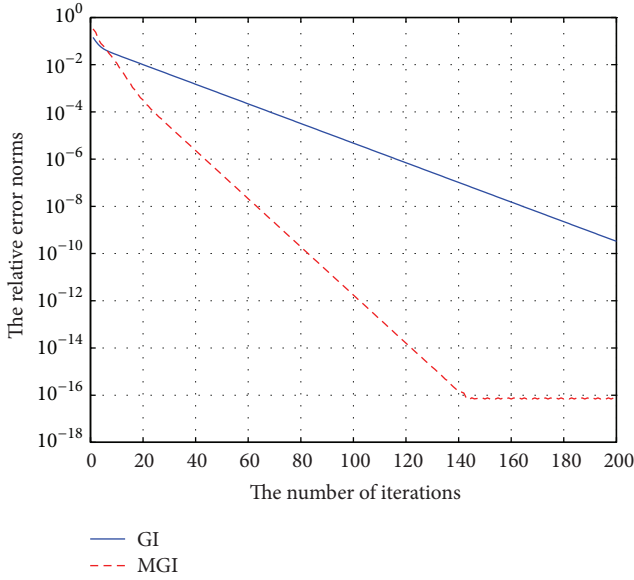


FIGURE 1: Comparison of convergence curves using GI and MGI algorithms.

sequence  $\{\mathcal{X}^k\}$ . These values are plotted on logarithmic scales, if the curve is approximate decreasing line, then it is indicated that the convergence of the sequence is at least linear.

From Figure 1, we can see that two algorithms are convergent linearly and MGI algorithm converges much faster than GI algorithm.

*Remark 6.* The effect of changing the step length  $\mu$  and  $\kappa$  is illustrated in Figure 2. We observe that the larger the step length is, the faster the convergence of the algorithm. However, if the step length is too large, the algorithm may diverge. How to choose a best step length is still a issue to be studied [22].

*Example 7.* This example comes from [22], except for  $C$  and  $\mathcal{D}$  which will be set to a random matrix (and tensor). The MATLAB codes which generate these matrices and compute the step lengths are as follow:

```

m = 30; n = 30; l = 30; rand('state',0);
alpha = 3;
A = triu(rand(m,m),1) + diag(alpha +
diag(rand(m)));
B = triu(rand(n,n),1) + diag(alpha + diag
(rand(n)));
C = triu(rand(1,1),1) + diag(alpha
+ diag(rand(1)));
D = tenrand([m,n,1]);
mu = 2.2/(norm(A)^2 + norm(B)^2 + norm
(C)^2); %Compute mu

```

TABLE 1: The effect of the condition number of matrix  $P$  on two algorithms.

$\alpha$	$\kappa(P)$	The number of iteration (GI)	The number of iteration (MGI)
2	73.1190	461	377
3	22.1410	134	105
5	7.9640	44	33
8	4.1285	40	14

```

kappa = 1.2 * min([1/norm(A)^2, 1/norm
(B)^2, 1/norm(C)^2]); %Compute kappa.

```

Since the exact solution in this example is unknown, we plot the relative residual errors

$$\frac{\|\mathcal{X}^k \times_1 A + \mathcal{X}^k \times_2 B + \mathcal{X}^k \times_3 C - \mathcal{D}\|}{\|\mathcal{D}\|}. \quad (30)$$

From Figure 3, we can see that the convergence of GI algorithm tends to slow down after some iterative steps. The MGI algorithm converges linearly, and much faster than GI method.

*Example 8.* In this example, we consider the effect of the condition number of matrix  $P$  on two algorithms. The stopping criterion is set to be  $10^{-6}$ . The MATLAB codes which generate these matrices are as follows:

```

m = 20; n = 10; l = 5; rand('state',0);
A = rand(m,m) + diag(ones(m,1)) * alpha;
B = rand(n,n) + diag(ones(n,1)) * alpha;
C = rand(1,1) + diag(ones(1,1)) * alpha;
D = tenrand([m,n,1]).

```

For different  $\alpha$ , the number of iterations and the corresponding condition number ( $\kappa(P)$ ) of the matrix  $P$  are outlined in Table 1.

The results of Table 1 affirm the analysis at the end of Section 4. In particular, we see that the convergence rate becomes faster as the condition number of  $P$  decreases.

## 6. Conclusions

We have presented the gradient based iterative algorithms for solving Sylvester tensor equation, Sylvester matrix equation as a special case included. They are based on the hierarchical identification principle and tensor arithmetic concepts. The convergence analysis indicates that the iterative solutions given by the proposed algorithms converge to exact solution for any initial values. Such statement is also confirmed by the given numerical examples. In addition, the idea of this paper can be easily extended to study iterative solutions of high order ( $\mathcal{X}$  is  $N$ -mode tensor) Sylvester tensor equation.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 11201092, no. 11261012) and

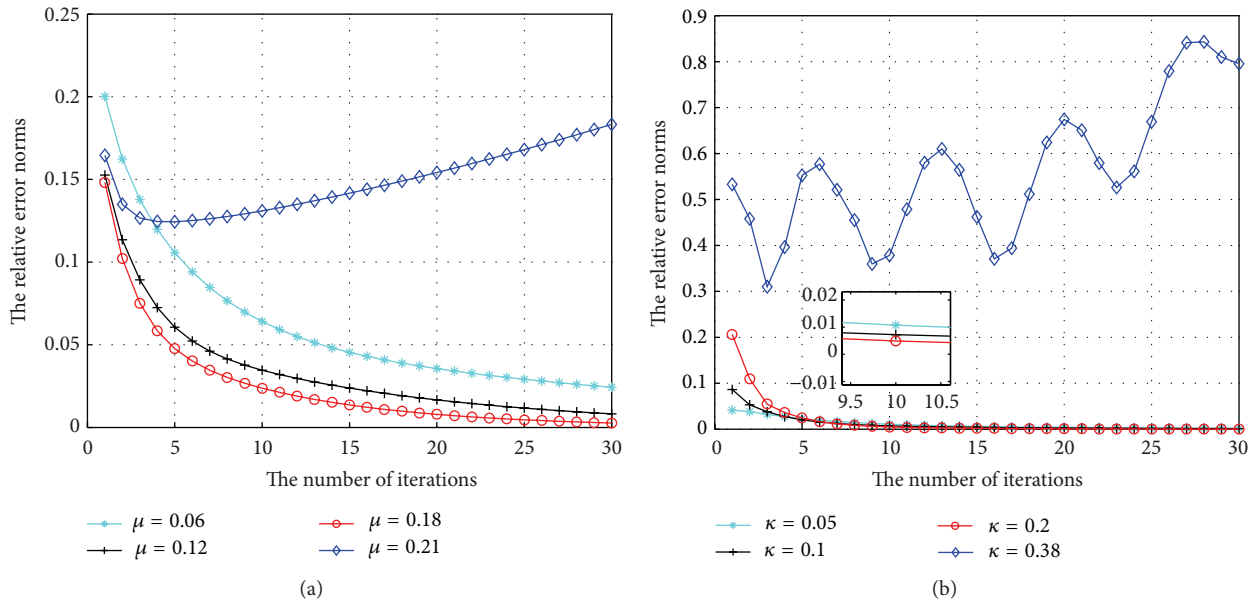


FIGURE 2: The effect of changing the step length.

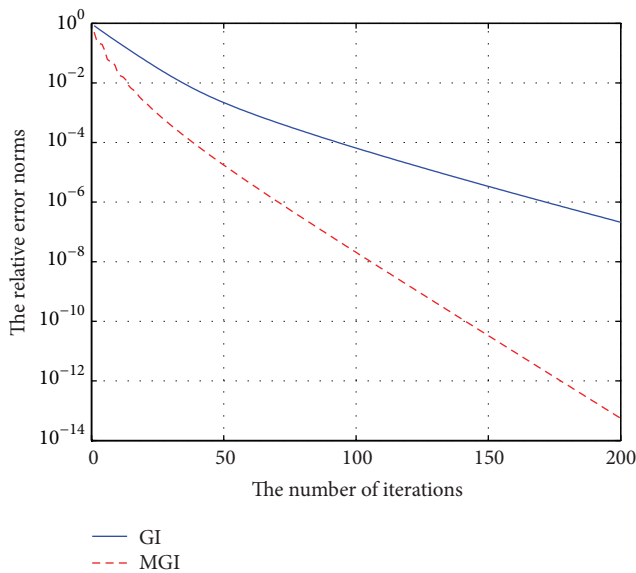


FIGURE 3: Comparison of convergence curves using GI and MGI algorithms.

the Scientific Research Starting Foundation for Doctors of Guizhou Normal University, China.

## References

- [1] G. H. Golub, S. Nash, and C. Van Loan, "A Hessenberg-Schur method for the problem  $AX+XB=C$ ," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 909–913, 1979.
- [2] U. Baur and P. Benner, "Cross-Gramian based model reduction for data-sparse systems," *Electronic Transactions on Numerical Analysis*, vol. 31, pp. 256–270, 2008.
- [3] D. Calvetti and L. Reichel, "Application of ADI iterative methods to the restoration of noisy images," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 165–186, 1996.
- [4] J. Li, F. Ding, and G. Yang, "Maximum likelihood least squares identification method for input nonlinear finite impulse response moving average systems," *Mathematical and Computer Modelling*, vol. 55, no. 3-4, pp. 442–450, 2012.
- [5] F. Ding, "Hierarchical multi-innovation stochastic gradient algorithm for Hammerstein nonlinear system modeling," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 1694–1704, 2013.
- [6] W. Wang, F. Ding, and J. Dai, "Maximum likelihood least squares identification for systems with autoregressive moving average noise," *Applied Mathematical Modelling*, vol. 36, no. 5, pp. 1842–1853, 2012.
- [7] F. Ding, Y. Liu, and B. Bao, "Gradient based and least squares based iterative estimation algorithms for multi-input multi-output systems," *Proceedings of the Institution of Mechanical Engineers I*, vol. 226, no. 1, pp. 43–55, 2012.
- [8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.
- [9] J. A. Heinen, "Technique for solving the extended discrete Lyapunov matrix equation," *IEEE Transactions on Automatic Control*, vol. 17, no. 1, pp. 156–157, 1972.
- [10] D. C. Sorensen and Y. Zhou, "Direct methods for matrix Sylvester and Lyapunov equations," *Journal of Applied Mathematics*, no. 6, pp. 277–303, 2003.
- [11] H. S. Najafi and A. H. Refahi Sheikhan, "Refinement methods for state estimation via Sylvester-observer equation," *Advances in Numerical Analysis*, vol. 2011, Article ID 184314, 15 pages, 2011.
- [12] H. Li and R. Li, "A note on the inversion of Sylvester matrices in control systems," *Mathematical Problems in Engineering*, vol. 2011, Article ID 609863, 10 pages, 2011.
- [13] B.-W. Li, S. Tian, Y.-S. Sun, and Z.-M. Hu, "Schur-decomposition for 3D matrix equations and its application

- in solving radiative discrete ordinates equations discretized by Chebyshev collocation spectral method,” *Journal of Computational Physics*, vol. 229, no. 4, pp. 1198–1212, 2010.
- [14] G.-S. Wang, Q. Lv, and G.-R. Duan, “On the parametric solution to the second-order Sylvester matrix equation  $MVF^2 + DVF + KV = BW$ ,” *Mathematical Problems in Engineering*, vol. 2007, Article ID 21078, 16 pages, 2007.
- [15] J. Zhou, R. Wang, and Q. Niu, “A preconditioned iteration method for solving Sylvester equations,” *Journal of Applied Mathematics*, vol. 2012, Article ID 401059, 12 pages, 2012.
- [16] L. Xie, Y. Liu, and H. Yang, “Gradient based and least squares based iterative algorithms for matrix equations  $AXB + CX^T D = F$ ,” *Applied Mathematics and Computation*, vol. 217, no. 5, pp. 2191–2199, 2010.
- [17] J. Ding, Y. Liu, and F. Ding, “Iterative solutions to matrix equations of the form  $AixBi=Fi$ ,” *Computers & Mathematics with Applications*, vol. 59, no. 11, pp. 3500–3507, 2010.
- [18] P. Benner, R.-C. Li, and N. Truhar, “On the ADI method for Sylvester equations,” *Journal of Computational and Applied Mathematics*, vol. 233, no. 4, pp. 1035–1045, 2009.
- [19] L. Grasedyck and W. Hackbusch, “A multigrid method to solve large scale Sylvester equations,” *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 3, pp. 870–894, 2007.
- [20] D. Y. Hu and L. Reichel, “Krylov-subspace methods for the Sylvester equation,” *Linear Algebra and its Applications*, vol. 172, pp. 283–313, 1992.
- [21] J.-R. Li and J. White, “Low-rank solution of Lyapunov equations,” *SIAM Review*, vol. 46, no. 4, pp. 693–713, 2004.
- [22] F. Ding and T. Chen, “Gradient based iterative algorithms for solving a class of matrix equations,” *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216–1221, 2005.
- [23] F. Ding and T. Chen, “Iterative least-squares solutions of coupled Sylvester matrix equations,” *Systems & Control Letters*, vol. 54, no. 2, pp. 95–107, 2005.
- [24] F. Ding and T. Chen, “On iterative solutions of general coupled matrix equations,” *SIAM Journal on Control and Optimization*, vol. 44, no. 6, pp. 2269–2284, 2006.
- [25] L. Xie, J. Ding, and F. Ding, “Gradient based iterative solutions for general linear matrix equations,” *Computers & Mathematics with Applications*, vol. 58, no. 7, pp. 1441–1448, 2009.
- [26] Q. Niu, X. Wang, and L.-Z. Lu, “A relaxed gradient based algorithm for solving Sylvester equations,” *Asian Journal of Control C*, vol. 13, no. 3, pp. 461–464, 2011.
- [27] F. Ding, P. X. Liu, and J. Ding, “Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle,” *Applied Mathematics and Computation*, vol. 197, no. 1, pp. 41–50, 2008.
- [28] L. Eldén, *Matrix Methods in Data Mining and Pattern Recognition*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 2007.
- [29] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [30] T. G. Kolda, “Multilinear Operators for Higher-Order Decompositions,” Tech. Rep. SAND2006-2081, Sandia National Laboratories, Livermore, Calif, USA, 2006.
- [31] B. W. Bader and T. G. Kolda, “Efficient MATLAB computations with sparse and factored tensors,” *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 205–231, 2007/08.
- [32] B. W. Bader and T. G. Kolda, “MATLAB Tensor Toolbox, Version 2.4,” 2010, <http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/>.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

