Journal of Mathematical Modeling
Vol. 2, No. 2, 2015, pp. 115-131

JMM

Residual norm steepest descent based iterative algorithms for Sylvester tensor equations

Fatemeh Panjeh Ali Beik* **and Salman Ahmadi-Asl**

Department of Mathematics, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran
Emails: f.beik@vru.ac.ir, s.ahmadiasl@stu.vru.ac.ir

Abstract. Consider the following consistent Sylvester tensor equation

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

where the matrices A, B, C and the tensor $\mathcal{D}$ are given and $\mathcal{X}$ is the unknown tensor. The current paper concerns with examining a simple and neat framework for accelerating the speed of convergence of the gradient-based iterative algorithm and its modified version for solving the mentioned Sylvester tensor equation without setting the restriction of the existence of a unique solution. Numerical experiments are reported which confirm the validity of the presented results.

Keywords: Sylvester tensor equation, iterative algorithm, convergence.
AMS Subject Classification: 15A24, 65F10.

1 Introduction and preliminaries

A tensor is a multidimensional array, for an extensive survey on the subject of higher-order tensors and their decomposition one may refer to [10]. The order of a tensor is the number of dimensions which is called by modes or ways. Throughout this paper, matrices (tensors of order two) are signified

*Corresponding author.
Received: 6 June 2014 / Revised: 11 July 2014 / Accepted: 22 July 2014.

by capital letters. Higher-order tensors (here order three) are indicated by Euler script letters, e.g., $\mathcal{X}$.

A fundamental operation for a tensor is the tensor-matrix multiplication. The 1-mode tensor product of tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ by a matrix $A \in \mathbb{R}^{P\times I}$ is denoted by $\mathcal{X} \times_1 A$ which is a $P \times J \times K$ tensor and its entries are given by

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

Similarly, the elements of the 2-mode multiplication of $\mathcal{X}$ by a matrix $B \in \mathbb{R}^{Q\times J}$ are expounded by

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

In an analogous manner the 3-mode multiplication can be determined.

Definition 1. *The inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I\times J\times K}$ is defined by*

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

and the corresponding induced norm is given by

$$\|\mathcal{X}\|^2 = \langle \mathcal{X}, \mathcal{X} \rangle .$$

In this paper we consider the Sylvester tensor equation

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

where the matrices $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{l\times l}$ and the tensor $\mathcal{D} \in \mathbb{R}^{m\times n\times l}$ are given. Note that (1) may appear from finite difference discretization of a linear partial equation in three dimension; for further details see [1, 3]. In addition, in the case that $\mathcal{X}$ is a simple 2-mode tensor ($\mathcal{X}$ is matrix), the tensor equation (1) reduces to the Sylvester matrix equation $AX + XB^T = D$ which has wide application in control theory, signal processing and system identification; see [5, 6, 9] and the references therein.

It is not difficult to verify that (1) is equivalent to the following linear system of equations

$$\mathcal{W}x = b,$$

with $x = \text{vec}(\mathcal{X})$, $b = \text{vec}(\mathcal{D})$ and

$$\mathcal{W} = I_l \otimes I_n \otimes A + I_l \otimes B \otimes I_m + C \otimes I_n \otimes I_m,$$

where $\otimes$ denotes the Kronecker product [4], I_n stands for the identity matrix of order n and the "vec" operator stacks the column of a matrix (or a tensor) to form a vector. For a square matrix X, its spectrum is denoted by $\sigma(X)$. It is not difficult to see that the set of all eigenvalues of $\mathcal{W}$ is given by

$$\begin{aligned}\sigma(\mathcal{W}) \quad = \quad & \{\lambda_i + \mu_j + \nu_k \mid \lambda_i \in \sigma(A),\ \mu_j \in \sigma(B),\ \sigma(C) \in \nu_k \\ & i = 1, \ldots, m, \quad j = 1, \ldots, n, \quad k = 1, \ldots, l\}\,.\end{aligned}$$

Evidently, the linear system $\mathcal{W}x = b$ is consistent if b belongs to the range of $\mathcal{W}$ and it has a unique solution if and only if $0 \notin \sigma(\mathcal{W})$. It is well-known that the linear system $\mathcal{W}x = b$ is consistent if and only if (1) is consistent. Evidently, the size of the linear system $\mathcal{W}x = b$ would be huge even for moderate values of l, m and n. Consequently, it is more desirable to apply an iterative method based on tensor format for solving (1).

In the case that (1) has a unique solution, Chen and Lu [3] have developed a projection method on the tensor format for solving (1). In fact the authors have examined the tensor form of the well-known GMRES method. In [2], the gradient-based (GB) and modified gradient-based (MGB) methods have been proposed to determine the solution of (1) under the restriction that (1) has a unique solution. We would like to comment here that the GB and MGB algorithms rely on the fixed parameters μ and κ. It has been established that if μ (κ) satisfies a sufficient condition, then the proposed GB (MGB) algorithm converges to the unique solution of (1) for an arbitrary given initial value tensor. In the current paper we aim to relax the referred limitation on the hypophysis and ameliorate the speed of convergence of the algorithms proposed by Chen and Lu [2]. More precisely, we suppose that (1) is consistent and then improve the convergence rate of the GB and MGB algorithms for solving (1) by exploiting the idea of an oblique projection method; for further details in the topic of the projection methods see Chapter 5 of [11].

The remainder of this paper is organized as follows. In Section 2, we momentarily review the GB and MGB algorithms for solving (1) and mention the disadvantages of the algorithms which motivate us to improve these algorithms. In Section 3, it reveals that how the speed of convergence of the GB and MGB algorithms can be accelerated without setting the restriction of the existence of a unique solution. Some numerical experiments are presented in Section 4 to illustrate the applicability and feasibly of the proposed methods. Finally a brief conclusion is a subject of Section 5.

2 The GB and MGB methods for the Sylvester tensor equation

In the literature, the gradient-based (GB) iterative algorithm is a common approach for solving matrix equations. For instance, Ding and Chen [7, 8] have presented various iterative methods based on the hierarchical identification principle to solve different kinds of matrix equations. Lately, Zhou et al. [12, 13, 14, 15] have offered the gradient-based algorithms to solve some kinds of (coupled) matrix equations. The examined algorithms depend on a fixed parameter denoted by μ. In each of these works, Zhou et al. have assumed that their considered problem has a unique solution and established a necessary and sufficient condition for the parameter μ under which the proposed algorithm is convergent. Furthermore, the optimum value for the fixed parameter μ has been derived which is not easy to compute in practice.

More recently, Chen and Lu [2] have developed the next algorithm to determine the unique solution of (1).

Algorithm 1: The gradient-based (GB) algorithm [2] for solving (1).

Data: $A, B, C, \mathcal{D}$ and μ.
Result: $\mathcal{X}$.
Initialize $\mathcal{X}^0$.
begin
  $\mathcal{R}^0 = \mathcal{D} - \mathcal{X}^0 \times_1 A - \mathcal{X}^0 \times_2 B - \mathcal{X}^0 \times_3 C$;
  for $k = 1, 2, \ldots, k_{max}$ **do**
    $\mathcal{X}_1^k = \mathcal{X}^{k-1} + \mu \mathcal{R}^{k-1} \times_1 A^T$;
    $\mathcal{X}_2^k = \mathcal{X}^{k-1} + \mu \mathcal{R}^{k-1} \times_2 B^T$;
    $\mathcal{X}_3^k = \mathcal{X}^{k-1} + \mu \mathcal{R}^{k-1} \times_3 C^T$;
    $\mathcal{X}^k = \left(\mathcal{X}_1^k + \mathcal{X}_2^k + \mathcal{X}_3^k\right)/3$;
    $\mathcal{R}^k = \mathcal{D} - \mathcal{X}^k \times_1 A - \mathcal{X}^k \times_2 B - \mathcal{X}^k \times_3 C$;
  end
end

It has been proved that if

$$0 < \mu < 2/(\|A\|_2^2 + \|B\|_2^2 + \|C\|_2^2), \tag{2}$$

then Algorithm 1 is convergent to the unique solution of (1) for any initial value $\mathcal{X}^0$. Nevertheless, checking the condition (2) is difficult in general

circumstances. Meanwhile, the inequality (2) is a sufficient condition for the convergence of Algorithm 1 and the optimum value for μ has not been derived. In order to improve the convergence speed of the GB algorithm, a modified algorithm (Algorithm 2) has been handled which depends on a fixed parameter κ.

Algorithm 2: The modified gradient based (MGB) algorithm [2] for solving (1).

Data: $A, B, C, \mathcal{D}$ and κ.
Result: $\mathcal{X}$.
Initialize $\mathcal{X}^0$.
begin
  for $k = 1, 2, \ldots, k_{max}$ **do**

$$\mathcal{X}_1^k = \mathcal{X}^{k-1} + \kappa(\mathcal{D} - \mathcal{X}^{k-1} \times_1 A - \mathcal{X}^{k-1} \times_2 B - \mathcal{X}^{k-1} \times_3 C) \times_1 A^T;$$
$$\mathcal{X}^{k-1} = \left(\mathcal{X}_1^k + \mathcal{X}_2^{k-1} + \mathcal{X}_3^{k-1}\right)/3;$$

$$\mathcal{X}_2^k = \mathcal{X}^{k-1} + \kappa(\mathcal{D} - \mathcal{X}^{k-1} \times_1 A - \mathcal{X}^{k-1} \times_2 B - \mathcal{X}^{k-1} \times_3 C) \times_2 B^T;$$
$$\mathcal{X}^{k-1} = \left(\mathcal{X}_1^k + \mathcal{X}_2^k + \mathcal{X}_3^{k-1}\right)/3,$$

$$\mathcal{X}_3^k = \mathcal{X}^{k-1} + \kappa(\mathcal{D} - \mathcal{X}^{k-1} \times_1 A - \mathcal{X}^{k-1} \times_2 B - \mathcal{X}^{k-1} \times_3 C) \times_3 C^T;$$
$$\mathcal{X}^k = \left(\mathcal{X}_1^k + \mathcal{X}_2^k + \mathcal{X}_3^k\right)/3;$$

  end
end

Analogous to the GB method, the convergence of Algorithm 2 has been proved under the restriction that (1) has unique solution where the parameter κ satisfies the following condition

$$0 < \kappa < \min\{1/\|A\|_2^2, 1/\|B\|_2^2, 1/\|C\|_2^2\}. \tag{3}$$

We would like to comment that condition (3) for the parameter κ in Algorithm 2 is a sufficient condition which guarantees the convergence of the algorithm for arbitrary initial value $\mathcal{X}^0$. Whereas its optimum value has not been obtained.

Remark 1. *Now let us summery the drawback regarding applying the GB and MGB algorithms for solving* (1) *as follows:*

- *The convergence of the algorithms have not been studied without setting the restriction of the existence of the unique solution.*

- *Each of the proposed algorithms relies on a fixed parameter which satisfies a sufficient condition which is not easy to check out in general situations.*
- *The optimum values for the fixed parameters exploited in Algorithms 1 and 2 have not been derived.*

In order to overcome the disadvantages which are mentioned in Remark 1, we offer using an oblique projection processes. To this end, we propose two new algorithms by choosing the fixed parameters in a progressive manner. As a matter of fact, we use the idea of a projection technique to select these parameters such that at each step the residual tensor corresponding to the new approximation satisfies an optimality property at each step of the algorithms. As seen the convergence of the algorithms can be studied without assuming the curtailment in hypophysis that (1) has a unique solution. That is we only suppose that (1) is consistent.

3 Main results

In this section we exploit the idea of a projection technique to improve the speed of converges of Algorithms 1 and 2. In fact, at each iterate say kth iterate, we derive the parameter such that the norm of the residual tensor

$$\mathcal{R}^{k+1} = \mathcal{D} - \mathcal{X}^{k+1} \times_1 A - \mathcal{X}^{k+1} \times_2 B - \mathcal{X}^{k+1} \times_3 C,$$

corresponding to the new approximation $\mathcal{X}^{k+1}$ is minimized over

$$S_k = \{\overline{\mathcal{X}} \mid \overline{\mathcal{X}} = \mathcal{X}^k + \alpha \mathcal{P}^k \text{ for } \alpha > 0\},$$

where

$$\mathcal{P}^k = \mathcal{R}^k \times_1 A^T + \mathcal{R}^k \times_2 B^T + \mathcal{R}^k \times_3 C^T.$$

For the basic concepts of the projection techniques for solving the linear system of equations, one can refer to Chapter 5 of [11]. In fact our goal is to find α^* such that the residual tensor $\mathcal{R}^{k+1}$ associated with $\mathcal{X}^{k+1} = \mathcal{X}^k + \alpha^* \mathcal{P}^k$ satisfies

$$\|\mathcal{R}^{k+1}\| = \min_{\overline{\mathcal{X}} \in S_k} \|\mathcal{D} - \overline{\mathcal{X}} \times_1 A - \overline{\mathcal{X}} \times_2 B - \overline{\mathcal{X}} \times_3 C\|.$$

To this end, it is sufficient to determine α^* such that

$$\left\langle \mathcal{R}^{k+1}, \mathcal{A}(\mathcal{P}^k) \right\rangle = 0,$$

where

$$\mathcal{A}(\mathcal{P}^k) = \mathcal{P}^k \times_1 A + \mathcal{P}^k \times_2 B + \mathcal{P}^k \times_3 C.$$

Or equivalently, we may set

$$\alpha^* = \frac{\langle \mathcal{R}^k, \mathcal{A}(\mathcal{P}^k) \rangle}{\langle \mathcal{A}(\mathcal{P}^k), \mathcal{A}(\mathcal{P}^k) \rangle},$$

where

$$\left\langle \mathcal{A}(\mathcal{P}^k), \mathcal{A}(\mathcal{P}^k) \right\rangle \neq 0. \tag{4}$$

Evidently S_k incorporates the $(k+1)$th approximate solution obtained by Algorithm 1 for $\alpha = \frac{\mu}{3}$. In the proof Theorem 2, we demonstrate that if the left-hand side of (4) become zero, i.e. $\left\langle \mathcal{A}(\mathcal{P}^k), \mathcal{A}(\mathcal{P}^k) \right\rangle = 0$, then $\mathcal{X}^k$ is an exact solution of (1). Now we present the following useful theorem which turns out that $\mathcal{P}^k = 0$ implies that $\mathcal{R}^k = 0$, i.e., $\mathcal{X}^k$ satisfies (1).

Theorem 1. *Suppose that* (1) *is consistent. Presume that* $\mathcal{X}_*$ *is an arbitrary solution of* (1)*. Then,*

$$\left\langle \mathcal{P}^k, \mathcal{X}_* - \mathcal{X}^k \right\rangle = \left\langle \mathcal{R}^k, \mathcal{R}^k \right\rangle.$$

Proof. It is not difficult to see that i-mode $(i = 1, 2, 3)$ multiplication commutes with respect to the inner product. That is for an arbitrary tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I \times J \times K}$ and a given matrix A with suitable dimension, we have

$$\langle \mathcal{X}, \mathcal{Y} \times_i A \rangle = \left\langle \mathcal{X} \times_i A^T, \mathcal{Y} \right\rangle.$$

Invoking the above equality, it can be easily seen that

$$\begin{aligned} \left\langle \mathcal{P}^k, \mathcal{X}_* - \mathcal{X}^k \right\rangle &= \left\langle \mathcal{R}^k \times_1 A^T + \mathcal{R}^k \times_2 B^T + \mathcal{R}^k \times_3 C^T, \mathcal{X}_* - \mathcal{X}^k \right\rangle \\ &= \left\langle \mathcal{R}^k, \mathcal{D} - \left(\mathcal{X}^k \times_1 A + \mathcal{X}^k \times_2 B + \mathcal{X}^k \times_3 C \right) \right\rangle \\ &= \left\langle \mathcal{R}^k, \mathcal{R}^k \right\rangle, \end{aligned}$$

which completes the proof. □

Now we may present our offered algorithm, Algorithm 3, which outperforms the GB algorithm. As seen in our proposed manner the fixed parameter μ is chosen in a progressive way. The algorithm an the extended form of the Residual Norm Steepest Descent (RNSD) [11] method.

The following theorem shows that Algorithm 3 converges to a solution of the (1).

Theorem 2. *Presume that the Sylvester tensor equation (1) is consistent. Then for $k = 1, 2, \ldots$, the next statements hold.*
a. *If $\langle \mathcal{A}(\mathcal{P}^k), \mathcal{A}(\mathcal{P}^k) \rangle \neq 0$ then $\|\mathcal{R}^k\| < \|\mathcal{R}^{k-1}\|$.*
b. *If $\langle \mathcal{A}(\mathcal{P}^k), \mathcal{A}(\mathcal{P}^k) \rangle = 0$ then $\mathcal{X}^k$ is a solution of (1).*

Proof. Straightforward computations demonstrate that

$$\left\langle \mathcal{R}^{k+1}, \mathcal{R}^{k+1} \right\rangle = \left\langle \mathcal{R}^k, \mathcal{R}^k \right\rangle \left[1 - \frac{\left\langle \mathcal{R}^k, \mathcal{A}(\mathcal{P}^k) \right\rangle^2}{\left\langle \mathcal{A}(\mathcal{P}^k), \mathcal{A}(\mathcal{P}^k) \right\rangle \left\langle \mathcal{R}^k, \mathcal{R}^k \right\rangle} \right].$$

From the Cauchy–Schwarz inequality, the above relation implies that $\|\mathcal{R}^{k+1}\| \leq \|\mathcal{R}^k\|$. It can be easily seen that

$$\left\langle \mathcal{R}^k, \mathcal{A}(\mathcal{P}^k) \right\rangle = \left\langle \mathcal{P}^k, \mathcal{P}^k \right\rangle. \tag{5}$$

From (5), we may conclude that $\langle \mathcal{R}^k, \mathcal{A}(\mathcal{P}^k) \rangle = 0$ if and only if $\langle \mathcal{P}^k, \mathcal{P}^k \rangle = 0$. Therefore $\langle \mathcal{R}^k, \mathcal{A}(\mathcal{P}^k) \rangle = 0$ iff $\mathcal{P}^k = 0$. Hence, the inequality holds strictly in (5) if $\mathcal{P}^k \neq 0$. Consequently without loss of generality we may assume that

$$\left\langle \mathcal{R}^k, \mathcal{A}(\mathcal{P}^k) \right\rangle \neq 0,$$

because if $\mathcal{P}^k = 0$ then Theorem 1 implies that $\mathcal{X}^k$ is an exact solution of (1) and there would be no requirement to compute the new approximate solution. For proving the second statement, first note that $\langle \mathcal{A}(\mathcal{P}^k), \mathcal{A}(\mathcal{P}^k) \rangle = 0$ implies that $\mathcal{A}(\mathcal{P}^k) = 0$. Now the validity of the second assertion can be deduced from (5) immediately. □

Here we would like to comment that in the case that (1) has a unique solution, analogous to the above manner, the convergence of Algorithm 2 (MGB) can be improved by utilizing the projection technique which is given in Algorithm 4 (called by MRNSD(1)). The convergence analysis of the algorithm is similar to Algorithm 3 hence we omit the details. Consider Lines 7, 13 and 19 of Algorithm 4 where the directions for updating the new approximations are constructed. Note that the directions are not dependent in general and this may lead to loosing the convergence in the case that (1) has infinity number of solutions. In this case the modification should be employed in an alternative scheme. For instance we suggest to split the matrices A, B and C into the following forms

$$\begin{aligned} A &= A_1 + A_2 + A_3, \\ B &= B_1 + B_2 + B_3, \\ C &= C_1 + C_2 + C_3, \end{aligned} \tag{6}$$

such that $A_i \circ A_j = 0$, $B_i \circ B_j = 0$ and $C_i \circ C_j = 0$ for $i \neq j$. Here $\circ$ stands for the well-known Hadamard product. Note that there would be different manners for choosing the splittings and further works can be focused on the way of choosing these splittings. Nevertheless for an arbitrary choices of theses splittings, the resulting approach surpasses both RNSD and GB methods in the case that (1) is consistent and has infinity number of solutions. The proposed algorithm is named MNRSD(2). In order to derive MNRSD(2), we only need to change three lines of Algorithm 4, i.e., Lines 7, 13 and 19 are respectively replaced by

$$\begin{aligned}
\text{Line } 7 \quad &\longleftarrow \quad \mathcal{P} = \mathcal{R} \times_1 A_1^T + \mathcal{R} \times_2 B_1^T + \mathcal{R} \times_3 C_1^T, \\
\text{Line } 13 \quad &\longleftarrow \quad \mathcal{P} = \mathcal{R} \times_1 A_2^T + \mathcal{R} \times_2 B_2^T + \mathcal{R} \times_3 C_2^T, \\
\text{Line } 19 \quad &\longleftarrow \quad \mathcal{P} = \mathcal{R} \times_1 A_3^T + \mathcal{R} \times_2 B_3^T + \mathcal{R} \times_3 C_3^T.
\end{aligned}$$

Algorithm 3: The RNSD method for solving (1).

Data: Input $A, B, C, \mathcal{D}$ and initialize $\mathcal{X}^0$.
Result: $\mathcal{X}$.
begin
  Set $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{X}_3 = \mathcal{X}^0$;
  Choose the tolerance $\epsilon > 0$;
  $\mathcal{R} = \mathcal{D} - \mathcal{X}^0 \times_1 A - \mathcal{X}^0 \times_2 B - \mathcal{X}^0 \times_3 C$;
  $\mathcal{X} = \mathcal{X}^0$;
  while $\|\mathcal{R}\| > \epsilon$ **do**
    $\mathcal{P} = \mathcal{R} \times_1 A^T + \mathcal{R} \times_2 B^T + \mathcal{R} \times_3 C^T$;
    $\mathcal{A}(\mathcal{P}) = \mathcal{P} \times_1 A + \mathcal{P} \times_2 B + \mathcal{P} \times_3 C$;
    $\alpha = \frac{\langle \mathcal{R}, \mathcal{A}(\mathcal{P}) \rangle}{\langle \mathcal{A}(\mathcal{P}), \mathcal{A}(\mathcal{P}) \rangle}$,
    $\mathcal{X}_1 = \mathcal{X} + \alpha \mathcal{P}$;
    $\mathcal{X}_2 = \mathcal{X} + \alpha \mathcal{P}$;
    $\mathcal{X}_3 = \mathcal{X} + \alpha \mathcal{P}$;
    $\mathcal{X} = (\mathcal{X}_1 + \mathcal{X}_2 + \mathcal{X}_3)/3$;
    $\mathcal{R} = \mathcal{D} - \mathcal{X} \times_1 A - \mathcal{X} \times_2 B - \mathcal{X} \times_3 C$;
  end
end

4 Numerical experiments

In this section we examine some numerical examples to illustrate the superiority of our proposed manners in comparison with Algorithms 1 and 2.

Algorithm 4: The modified RNSD (MRNSD(1)) method for solving (1) with unique solution.

Data: Input $A, B, C, \mathcal{D}$ and initialize $\mathcal{X}^0$.
Result: $\mathcal{X}$.

begin
  Set $\mathcal{X}_1^0 = \mathcal{X}_2 = \mathcal{X}_3 = \mathcal{X}_0$;
  Choose the tolerance $\epsilon > 0$;
  $\mathcal{R} = \mathcal{D} - \mathcal{X}^0 \times_1 A - \mathcal{X}^0 \times_2 B - \mathcal{X}^0 \times_3 C$;
  $\mathcal{X} = \mathcal{X}^0$;
  while $\|\mathcal{R}\| > \epsilon$ **do**
    $\mathcal{P} = \mathcal{R} \times_1 A^T$;
    $\mathcal{A}(\mathcal{P}) = \mathcal{P} \times_1 A + \mathcal{P} \times_2 B + \mathcal{P} \times_3 C$;
    $\alpha = \frac{\langle \mathcal{R}, \mathcal{A}(\mathcal{P}) \rangle}{\langle \mathcal{A}(\mathcal{P}), \mathcal{A}(\mathcal{P}) \rangle}$;
    $\mathcal{X}_1 = \mathcal{X} + \alpha \mathcal{P}$;
    $\mathcal{X} = (\mathcal{X}_1 + \mathcal{X}_2 + \mathcal{X}_3)/3$;
    $\mathcal{R} = \mathcal{D} - \mathcal{X} \times_1 A - \mathcal{X} \times_2 B - \mathcal{X} \times_3 C$;
    $\mathcal{P} = \mathcal{R} \times_2 B^T$;
    $\mathcal{A}(\mathcal{P}) = \mathcal{P} \times_1 A + \mathcal{P} \times_2 B + \mathcal{P} \times_3 C$;
    $\alpha = \frac{\langle \mathcal{R}, \mathcal{A}(\mathcal{P}) \rangle}{\langle \mathcal{A}(\mathcal{P}), \mathcal{A}(\mathcal{P}) \rangle}$;
    $\mathcal{X}_2 = \mathcal{X} + \alpha \mathcal{P}$;
    $\mathcal{X} = (\mathcal{X}_1 + \mathcal{X}_2 + \mathcal{X}_3)/3$;
    $\mathcal{R} = \mathcal{D} - \mathcal{X} \times_1 A - \mathcal{X} \times_2 B - \mathcal{X} \times_3 C$;
    $\mathcal{P} = \mathcal{R} \times_3 C^T$;
    $\mathcal{A}(\mathcal{P}) = \mathcal{P} \times_1 A + \mathcal{P} \times_2 B + \mathcal{P} \times_3 C$;
    $\alpha = \frac{\langle \mathcal{R}, \mathcal{A}(\mathcal{P}) \rangle}{\langle \mathcal{A}(\mathcal{P}), \mathcal{A}(\mathcal{P}) \rangle}$;
    $\mathcal{X}_3 = \mathcal{X} + \alpha \mathcal{P}$;
    $\mathcal{X} = (\mathcal{X}_1 + \mathcal{X}_2 + \mathcal{X}_3)/3$;
    $\mathcal{R} = \mathcal{D} - \mathcal{X} \times_1 A - \mathcal{X} \times_2 B - \mathcal{X} \times_3 C$;
  end
end

Table 1: Numerical results for Example 1.

	GB (μ=0.2)	MGB (κ=0.2747)	RNSD	MRNSD
Iteration	155	49	105	20
CPU time	0.4372	0.2781	0.3243	0.1658
$\|\mathcal{X}-\mathcal{X}^*\|_2$	9.9241e-006	8.9563e-006	8.9716e-006	9.4045e-006

All of the reported experiments were performed on a Pentium 4 PC with a 2.99 GHz CPU and 4.00GB of RAM.

Example 1. ([2]) Consider the Sylvester tensor equation (1) where

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

The exact solution of (1) is

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

In this example we use $\mathcal{X}^0 = 0$ as the initial guess and $\|\mathcal{X}-\mathcal{X}^*\| < 10^{-5}$, as the stopping criterion. In this case the optimal value of parameters μ and κ are chosen which have been derived experimentally and they are out of the range established in [2].

As seen Table 1 demonstrates that Algorithms 3 and 4 outperform Algorithms 1 and 2 for solving (1), respectively. For more clarification, we exhibit the convergence curves of the methods in Figure 1.

Example 2. In this instance, we mention the Sylvester tensor equation (1) such that

$$\begin{aligned} A &= \text{ones}(30) + \text{hilb}(30) + \text{eye}(30), \\ B &= \text{ones}(30) + \text{diag}(3 + \text{diag}(\text{hilb}(30))) + \text{eye}(30), \\ C &= \text{ones}(30) + \text{diag}(50 + \text{diag}(\text{hilb}(30))) + 4\text{eye}(30). \end{aligned}$$

The right hand sides $\mathcal{D}$ is chosen such that $\mathcal{X}^*$ is exact solutions of (1),

$$\mathcal{X}^*(:,:,i) = \text{ones}(30), \quad i = 1, 2, \ldots, 30.$$

We have applied the GB, MGB, RNSD and MRSND methods with $\mathcal{X}^0 = 0$ for solving the second example and as the exact solution is available the

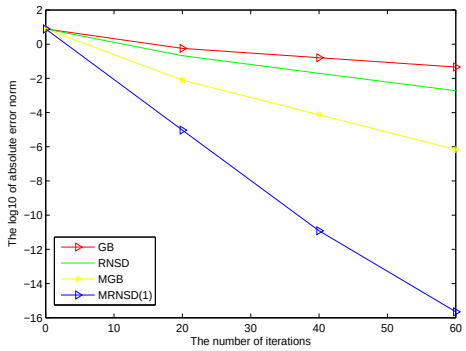

Figure 1: Convergence history for Example 1.

Table 2: Numerical results for Example 2.

	GB	MGB	RNSD	MRNSD
Iters	67	38	35	24
CPU time	0.3781	0.4399	0.2346	0.4274
$\|\mathcal{X}-\mathcal{X}^*\|_2$	8.3037e-007	8.5157e-007	8.7649e-007	6.3880e-007

stopping criterion $\|\mathcal{X}-\mathcal{X}^*\| < 10^{-6}$, is selected. In this case, the parameters μ and κ are chosen in the following manner (see [2])

$$\mu = 2.2/\left(\|A\|^2+\|B\|^2+\|C\|^2\right),$$
$$\kappa = 1.2\min\left(1/\|A\|^2, 1/\|B\|^2, 1/\|C\|^2\right).$$

The numerical results are reported in Table 2 which reveal that the RNSD and MRSND methods work better than the GB and MGB methods.

For more elucidation, the convergence history of the methods are depicted in Figure 2.

All the presented theorems in [2] such as the sufficient conditions to guarantee the convergence of GB and MGB methods are proved in the case that the Sylvester tensor equation (1) has a unique solution. Hence the application the GB and MGB methods for the Sylvester tensor equation (1) without setting the assumption of the existence of a unique solution has not been studied. In the two later examples, we consider the situations that the consistent Sylvester tensor equation (1) has not a unique solution.

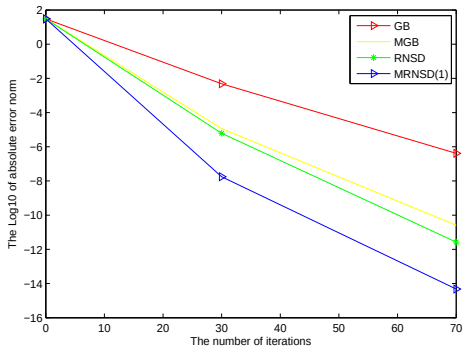

Figure 2: Convergence history for Example 2.

Example 3. Consider the Sylvester tensor equation (1) with

$$A = \begin{pmatrix} 1 & 0 \\ 10 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 10 & 0 \\ 11 & 31 \end{pmatrix}, \quad C = \begin{pmatrix} -11 & 0 \\ 1 & -34 \end{pmatrix},$$
$$\mathcal{D}(:,:,1) = \begin{pmatrix} 0 & 53 \\ 16 & 145 \end{pmatrix}, \quad \mathcal{D}(:,:,2) = \begin{pmatrix} -91 & 42 \\ -20 & 57 \end{pmatrix}.$$

Evidently $\sigma(A) = \{1, 3\}$, $\sigma(B) = \{10, 31\}$ and $\sigma(C) = \{-11, -34\}$ and one solution of the Sylvester tensor equation (1) is

$$\mathcal{X}^*(:,:,1) = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \quad \mathcal{X}^*(:,:,2) = \begin{pmatrix} 4 & 2 \\ 3 & 1 \end{pmatrix}. \tag{7}$$

where $\text{norm}(\mathcal{X}^*) = 7.7460$. Hence the mentioned problem has not a unique solution. In this case the parameters κ and μ are chosen in a similar manner applied in the previous example. The convergence curves of the methods are depicted in Figure 3 which reveal the slow convergence of GB and MGB methods and the efficiency of our proposed algorithm. By choosing $\mathcal{X}^{(0)}(:,:,1) = 0$, $\mathcal{X}^{(0)}(:,:,2) = 0$, as an initial guess and using RNSD algorithm after 100 iteration we obtain the following approximate solution

$$\mathcal{X}^{(100)}(:,:,1) = \begin{pmatrix} 1.1535 & 1.9196 \\ 2.2323 & 4.4021 \end{pmatrix},$$
$$\mathcal{X}^{(100)}(:,:,2) = \begin{pmatrix} 4.0067 & 1.9965 \\ 2.9666 & -1.9752\text{e}-018 \end{pmatrix},$$

where $\text{norm}(\mathcal{X}^{(100)}) = 7.6557$ and

$$\text{norm}\left(\mathcal{D} - \mathcal{X}^{(100)} \times_1 A - \mathcal{X}^{(100)} \times_2 B - \mathcal{X}^{(100)} \times_3 C\right) = 1.2413\text{e}-007.$$

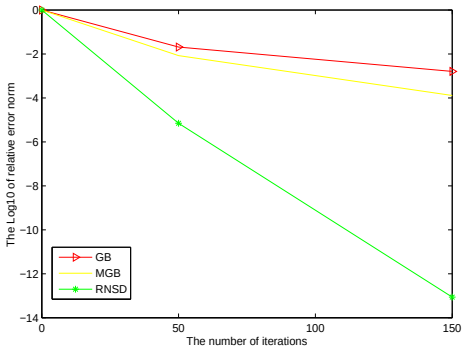

Figure 3: Convergence history for Example 3.

Also by choosing

$$\mathcal{X}^{(0)}(:,:,1) = \begin{pmatrix} 1 & -1 \\ 1 & 0 \end{pmatrix}, \quad \mathcal{X}^{(0)}(:,:,2) = \begin{pmatrix} 2 & 0 \\ 1 & 0 \end{pmatrix},$$

as an initial guess and using RNSD algorithm after 100 iteration we obtain the following approximate solution

$$\mathcal{X}^{(100)}(:,:,1) = \begin{pmatrix} 1.0449 & 1.9765 \\ 2.7757 & 4.1175 \end{pmatrix},$$

$$\mathcal{X}^{(100)}(:,:,2) = \begin{pmatrix} 4.0020 & 1.9990 \\ 2.9902 & -8.0760\text{e}-017 \end{pmatrix},$$

where $\text{norm}(\mathcal{X}^{(100)}) = 7.6557$ and

$$\text{norm}\left(\mathcal{D} - \mathcal{X}^{(100)} \times_1 A - \mathcal{X}^{(100)} \times_2 B - \mathcal{X}^{(100)} \times_3 C\right) = 1.2413\text{e}-007.$$

Example 4. Let us consider the Sylvester tensor equation (1) with

$$A = \begin{pmatrix} 31 & 0 \\ -1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 11 & 0 \\ -1 & -1 \end{pmatrix}, \quad C = \begin{pmatrix} 2 & 0 \\ 1 & -42 \end{pmatrix},$$
$$\mathcal{D}(:,:,1) = \begin{pmatrix} 44 & 63 \\ 44 & 7 \end{pmatrix}, \mathcal{D}(:,:,2) = \begin{pmatrix} 1 & -26 \\ -88 & -42 \end{pmatrix}.$$

As seen $\sigma(A) = \{31, 2\}$, $\sigma(B) = \{11, -1\}$ and $\sigma(C) = \{2, -42\}$ and $\mathcal{X}^*(:,:,1)$ and $\mathcal{X}^*(:,:,2)$, given in (7), satisfies the Sylvester tensor equation (1)

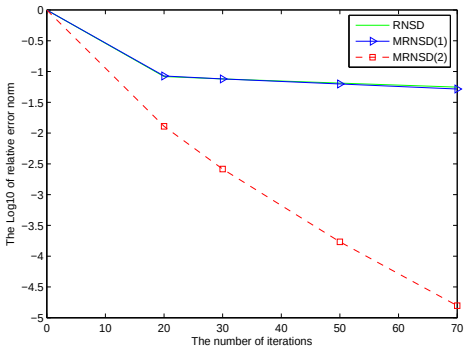

Figure 4: Convergence history for Example 4.

which implies that the above problem has not a unique solution. In this example we aim to illustrate the applicability of MRNSD(2). To this end we consider specific splittings for the matrices A, B and C as defined by (6) where

$$A_1 = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 31 & 0 \\ 0 & 0 \end{pmatrix},$$
$$B_1 = \begin{pmatrix} 11 & 0 \\ 0 & 0 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}, \quad B_3 = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix},$$
$$C_1 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad C_2 = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}, \quad C_3 = \begin{pmatrix} 0 & 0 \\ 0 & -42 \end{pmatrix}.$$

The convergence history of the methods are depicted in Figure 4.

5 Conclusion

We have concerned with developing projection based approaches for solving the Sylvester tensor equation without setting the assumption that the mentioned problem has a unique solution. It has been both theoretically and numerically demonstrated that the propounded algorithms outperform the gradient-based iterative algorithm.

Acknowledgements

The authors are grateful to anonymous referee for her/his valuable recommendations and comments which have improved the quality of the paper.

References

[1] Z.Z. Bai, G.H. Golub and M.K. Ng, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl. **24** (2003) 603–626.

[2] Z. Chen and L.Z. Lu, *A gradient based iterative solutions for Sylvester tensor equations*, Math. Probl. Eng. (2013), DOI: 10.1155/2013/819479.

[3] Z. Chen and L. Z. Lu, *A projection method and Kronecker product precondetioner for solving Sylvester tensor equations*, Science China, **55** (2012) 1281–1292.

[4] D.S. Bernstein, Matrix Mathematics: theory, facts, and formulas, Second edition, Princeton University Press, 2009.

[5] F. Ding and T. Chen, *Gradient based iterative algorithms for solving a class of matrix equations*, IEEE Trans. Authomat. Control, **50** (2005) 1216–1221.

[6] F. Ding and T. Chen, *Iterative least squares solutions of coupled Sylvester matrix equations*, Systems & Control Lett. **54** (2005) 95–107.

[7] F. Ding and T. Chen, *On iterative solutions of general coupled matrix equations*, SIAM J. Control Optim. **44** (2006) 2269–2284.

[8] F. Ding, P. X. Liu and J. Ding, Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle, Appl. Math. Comput. **197** (2008) 41–50.

[9] G.H. Golub, S. Nash and C.F. Van-Loan, *A Hessenberg-Schur method for the problem* $AX + XB = C$, IEEE Trans. Authomat. Control **24** (1979) 909–913.

[10] T.G. Kolda and B.W. Bader, *Tensor decompositions and applications*, SIAM Rev. **51** (2009) 455–500.

[11] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS press, New York, 1995.

[12] B. Zhou and G.R. Duan, *On the generalized Sylvester mapping and matrix equation*, Systems & Control Lett. **57** (2008) 200–208.

[13] B. Zhou, G.R. Duan and Z.Y. Li, *Gradient based iterative algorithm for solving coupled matrix equations*, Systems & Control Lett. **58** (2009) 227–333.

[14] B. Zhou, G.R. Duan and Y. Wang, *Weighted least squares solutions to general coupled Sylvester matrix equations*, J. Comput. Appl. Math. **224** (2009) 759–776.

[15] B. Zhou, J. Lam J and G.R. Duan, *Gradient-based maximal convergence rate iterative method for solving linear matrix equations*, Int. J. Comput. Math. **87** (2010) 515–527.