RELAXED GRADIENT BASED ITERATIVE ALGORITHM FOR SOLVING THE GENERALIZED SYLVESTER MATRIX EQUATION

PUTTACHAT SONGSRI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE
(APPLIED MATHEMATICS)
DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
YEAR 2018

KMITL-2018-SC-M-001-029

หัวข้อวิทยานิพนธ์	ขั้นตอนวิธีทำซ้ำที่มีฐานมาจากเกรเดียนต์แบบผ่อนคลายสำหรับการหาผลเฉลยสมการเมทริกซ์ซิลเวสเตอร์ที่มีนัยทั่วไป
ชื่อนักศึกษา	นางสาวพุทธชาติ สองศรี
รหัสประจำตัว	59605016
ปริญญา	วิทยาศาสตรมหาบัณฑิต (คณิตศาสตร์ประยุกต์)
ภาควิชา	คณิตศาสตร์
พ.ศ.	2561
อาจารย์ที่ปรึกษา	ผศ.ดร.ภัทราวุธ จันทร์เสงี่ยม

บทคัดย่อ

เรานำเสนอวิธีทำซ้ำแบบใหม่สำหรับหาผลเฉลยสมการเมทริกซ์ซิลเวสเตอร์ที่มีนัยทั่วไป ซึ่งวิธีดังกล่าวมีฐานมาจากเกรเดียนต์และหลักการการระบุตามลำดับชั้น ในกรณีเฉพาะเราได้ขั้นตอนวิธีวนทำซ้ำสำหรับหาผลเฉลยของรูปแบบเฉพาะที่น่าสนใจของสมการเมทริกซ์ซิลเวสเตอร์ที่มีนัยทั่วไป และการจำลองเชิงตัวเลขแสดงให้เห็นถึงการใช้งานได้จริงและประสิทธิภาพของขั้นตอนวิธี

คำสำคัญ : ขั้นตอนวิธีทำซ้ำที่มีฐานมาจากเกรเดียนต์ ตัวดำเนินการเวกเตอร์ ผลคูณโครเนคเคอร์ สมการเมทริกซ์ซิลเวสเตอร์ที่มีนัยทั่วไป หลักการการระบุตามลำดับชั้น

Thesis Title	Relaxed gradient based iterative algorithm for solving the generalized Sylvester matrix equation
Student Name	Miss Puttachat Songsri
Student ID	59605016
Degree	Master of Science (Applied Mathematics)
Department	Mathematics
Year	2018
Thesis Advisor	Asst.Prof.Dr. Pattrawut Chansangiam

Abstract

We propose a new iterative method for solving the generalized Sylvester matrix equation. Our method is based on gradients and hierarchical identification principle. In particular, we obtain relaxed gradient based iterative algorithms for certain interesting special cases of the generalized Sylvester equation. Numerical simulations are also provided to show the applicability and the efficiency of the algorithm.

Keywords : Gradient based iterative algorithm, vector operator, Kronecker product, generalized Sylvester matrix equation, hierarchical identification principle.

Acknowledgements

I would like to express my sincere thanks to my thesis advisor, Asst.Prof.Dr. Pattrawut Chansangiam, for his invaluable help and constant encouragement throughout the course of this research. I am most grateful for his teaching and advice. I would not have achieved this far and this thesis would not have been completed without all the support that I have always received from him.

In addition, I am grateful for the rest of my thesis committee: Assoc.Prof.Dr. Wicharn Lewkeeratiyutkul, Asst.Prof.Dr.Nopparat Pochai, and Dr.Wannaporn Sanprasert, for their insightful comments.

I am sincerely grateful to financial supports from the Graduate Study of the Faculty of Science, King Mongkut's Institute of Technology Ladkrabang Fund.

Finally, I most gratefully acknowledge my parents and my friends for all their support throughout the period of my researh. Thank you.

Puttachat Songsri

Table of Contents

List of Tables

Table Page

Chapter 1

Introduction

1.1 Inception and importance

Consider the following linear dynamical system:

$$x'(t) = Ax(t) \tag{1.1}$$

where x is an unknown vector-valued function and A is a given square matrix. To analyse the stability of an equilibrium point of (1), it suffices to find a positive definite matrix P such that A^TP+PA is negative definite [6]. Hence, we will solve the so-called Lyapunov equation

$$AX - XA^T = Q \tag{1.2}$$

for some negative definite matrix Q. To discuss (2), we shall investigate a more general form, namely, the Sylvester equation:

$$AX + XB = C. \tag{1.3}$$

The generalized Sylvester equation takes the form

$$AXB + CXD = F. \tag{1.4}$$

Note that when $B = C = I$, (1.4) reduces to the Sylvester equation (1.3). (1.4) also includes the following equation as special cases :

$$AXB = C, \tag{1.5}$$

$$AXB + X = C. \tag{1.6}$$

By taking the vector operator, (1.4) is reduced to $Px = b$ where $P = B^T \otimes A + D^T \otimes C$, $x = \text{vec}\, X$, and $b = \text{vec}\, F$. Here, vec is the vector operator and the symbol $\otimes$ stands for the Kronecker product. Then, (1.4) has a unique solution if and only if P is nonsingular. This matrix equation has others profound application in linear system theory; see e.g. [6] and references therein.

However, if the dimension of matrices is high, e.g. if A, B, C, D are of dimension 100×100, the dimension of $B^T \otimes A + D^T \otimes C$ is 10000×10000. Such a dimension problem leads to computational difficulty in that excessive computer memory is required for computation and inversion of large matrices. Thus, this approach is only applicable for a small dimension of A.

On the other hand, there are many other methods for solving linear matrix equations. In particular, iterative methods have been received much attention. Gradient based iterative methods for (1.3), (1.5), (1.6) were investigated in [4, 2, 5, 9, 11,

12, 13, 3, 1]. For the generalized Sylvester equation (1.4), there are gradient based iterative method [5] as follows:

Algorithm 1.1.1. The gradient based iterative (GI) algorithm.

Step 1 Input matrices $A, B, C, D, F \in \mathbb{M}_n$ and an error $\varepsilon > 0$. Choose initial matrices $X_1(0)$ and $X_2(0)$. Set $k := 1$.

Step 2 If $\|AX(k)B + CX(k)D - F\|_F / \|F\|_F < \varepsilon$, stop; otherwise, go to Step 3.

Step 3 Update the sequences

$$\begin{aligned} X(k) &= \frac{X_1(k) + X_2(k)}{2}, \\ X_1(k) &= X(k-1) + \tau A^T[F - AX(k-1)B - CX(k-1)D]B^T, \\ X_2(k) &= X(k-1) + \tau C^T[F - AX(k-1)B - CX(k-1)D]D^T. \end{aligned}$$

Step 4 Set $k := k + 1$, return to Step 2.

If we choose the convergence factor τ such that

$$0 < \tau < \frac{2}{\|A\|_2^2\|B\|_2^2 + \|C\|_2^2\|D\|_2^2}, \tag{1.7}$$

then Algorithm 1.1.1 converges for any initial values $X_1(0)$ and $X_2(0)$.

In this paper, we shall propose a new iterative method for solving the generalized Sylvester equation (1.4). Our method is based on gradients and hierarchical identification principle. The convergence analysis shows that the proposed algorithm converges to the unique solution for any initial value matrix. In particular, we obtain relaxed gradient based iterative algorithms for (1.3), (1.5) and (1.6). Numerical simulations are also provided.

1.2 Objectives

1) Propose a new gradient based iterative method for solving the generalized Sylvester equation and deduce the convergence analysis of the proposed algorithm.

2) Propose a new gradient based iterative method for solving some special cases of the generalized Sylvester equation and deduce the convergence analysis of the proposed algorithm.

3) Give a numerical simulation for the proposed algorithm.

1.3 Scope of the study

All matrices considered here are real.

1.4 Benefits

To obtain a new algorithm for solving the generalized Sylvester equation.

1.5 Research methodology

1) Study preliminaries of matrix in Matrix theory.
2) Study research papers and textbooks concerning Kronecker products and the vector operator and norm of matrix.
3) Study definition and the properties of matrix differential equations.
4) Determine the objectives and scope of the research.
5) Study hierarchical identification principle from research papers.
6) Study iterative algorithms for the matrix equation from research papers.
7) Institute new gradient based iterative method for solving the generalized Sylvester matrix equation.
8) Provide numerical simulations.
9) Conclude the results, make suggestions for further works and write the thesis.

Table 1.1: The research schedule

Activity	Time frame							
	2016		2017				2018	
	Aug.-Sep.	Oct.-Dec.	Jan.-Mar.	Apr.-Jun.	Aug.-Sep.	Oct.-Dec.	Jan.-Apr.	May.-Jul.
Step 1	⟷							
Step 2		⟷	⟷					
Step 3			⟷	⟷				
Step 4					⟷			
Step 5						⟷		
Step 6							⟷	
Step 7							⟷	
Step 8								⟷
Step 9								⟷

Chapter 2

Preliminaries

The purpose of this chapter is to provide basic concepts and tools in matrix theory, linear algebra and functional analysis used in the research. The first section deals with preliminaries in matrix theory. We collect fundamental properties of the Kronecker product, vector operator, Kronecker sum of complex matrices. In the second section, we give definitions and some properties about norms of vector/matrices, sequences of vectors, and sequences and series of matrices. In the third section, we discuss iterative systems of linear equation. In the final section, we review literature results about iterative methods for solving certain linear matrix equations.

2.1 Kronecker product and the vector operator

In this section, we recall some basic definitions and results about of the Kronecker product and the vector operator of complex matrices. More detailed description can be found in [7, 8].

Denote by $\mathbb{M}_{m,n}$ the set of all $m \times n$ complex matrices, $\mathbb{M}_{n,n}$ is abbreviated to $\mathbb{M}_n$ and $\mathbb{M}_n(\mathbb{R})$ is the set of all $n \times n$ real matrices. We write A^T to indicate the transpose of $A \in \mathbb{M}_{m,n}$. The conjugate transpose A^* of $A \in \mathbb{M}_{m,n}$ is defined by $A^* = \bar{A}^T$ where $\bar{A}$ is the componentwise conjugate of A.

Definition 2.1.1. Let $A = [a_{ij}]$ and $B = [b_{kl}]$ be complex matrices of order $m \times n$ and $p \times q$, respectively. The **Kronecker product** of A and B (or **tensor product**), denoted as $A \otimes B$, is defined by

$$A \otimes B = [a_{ij}B]_{ij}, \tag{2.1}$$

where $a_{ij}B$ is the (i,j)th submatrix of order $p \times q$ and $A \otimes B$ of order $mp \times nq$.

Example 2.1.2. Consider

$$A = \begin{bmatrix} 0 & -2 \\ 3 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 1 & 5 & 0 \\ -4 & -2 & 6 & 3 \\ -3 & 2 & -1 & 4 \end{bmatrix}.$$

The Kronecker product of A and B is

$$A \otimes B = \begin{bmatrix} 0B & -2B \\ 3B & -1B \end{bmatrix}$$

$$= \begin{bmatrix} 0 \begin{bmatrix} 2 & 1 & 5 & 0 \\ -4 & -2 & 6 & 3 \\ -3 & 2 & -1 & 4 \end{bmatrix} & -2 \begin{bmatrix} 2 & 1 & 5 & 0 \\ -4 & -2 & 6 & 3 \\ -3 & 2 & -1 & 4 \end{bmatrix} \\ 3 \begin{bmatrix} 2 & 1 & 5 & 0 \\ -4 & -2 & 6 & 3 \\ -3 & 2 & -1 & 4 \end{bmatrix} & -1 \begin{bmatrix} 2 & 1 & 5 & 0 \\ -4 & -2 & 6 & 3 \\ -3 & 2 & -1 & 4 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & -4 & -2 & -10 & 0 \\ 0 & 0 & 0 & 0 & 8 & 4 & -12 & -6 \\ 0 & 0 & 0 & 0 & 6 & -4 & 2 & -8 \\ 6 & 3 & 15 & 0 & -2 & -1 & -5 & 0 \\ -12 & -6 & 18 & 9 & 4 & 2 & -6 & -3 \\ -9 & 6 & -3 & 12 & 3 & -2 & 1 & -4 \end{bmatrix}.$$

Definition 2.1.3. The vector operator is defined for each $A = [a_{ij}] \in \mathbb{M}_{m,n}$, by

$$\text{vec}(A) = [a_{11}, ..., a_{m1}, a_{12}, ..., a_{m2}, ..., a_{1n}, ..., a_{mn}]^T \in \mathbb{C}^{mn}.$$

The vector operator is clearly linear, bijective, and continuous.

Example 2.1.4.

$$\text{Consider } A = \begin{bmatrix} 2 & 3 \\ -5 & 4 \end{bmatrix}.$$

$$\text{Then, } \text{vec}(A) = \begin{bmatrix} 2 \\ -5 \\ 3 \\ 4 \end{bmatrix}$$

Proposition 2.1.5. *[8] Let $A, C \in \mathbb{M}_{m,n}$, $B, D \in \mathbb{M}_{p,q}$, $E \in \mathbb{M}_{r,s}$ and $\alpha \in \mathbb{C}$. Then*

(i) $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha(A \otimes B)$,

(ii) $(A + C) \otimes B = A \otimes B + C \otimes B$,

(iii) $A \otimes (B + D) = A \otimes B + A \otimes D$,

(iv) $A \otimes (B \otimes E) = (A \otimes B) \otimes E$,

(v) $(A \otimes B)^\star = A^\star \otimes B^\star$.

Theorem 2.1.6. *[8] Let $A \in \mathbb{M}_{m,n}$ and $B \in \mathbb{M}_{p,q}$ be given and let $X \in \mathbb{M}_{n,p}$ be unknown.* Then,

$$\text{vec}(AXB) = (B^T \otimes A)\,\text{vec}(X). \tag{2.2}$$

Definition 2.1.7. Let $A \in \mathbb{M}_n$ be Hermitian. Then A is called

- **negative definite** if $x^*Ax < 0$ for all $x \in \mathbb{C}^n - \{0\}$,
- **positive definite** if $x^*Ax > 0$ for all $x \in \mathbb{C}^n - \{0\}$,
- **positive semidefinite** if $x^*Ax \geqslant 0$ for all $x \in \mathbb{C}^n$.

2.2 Norms, sequences, and series of vectors and matrices

In this section, we review fundamental background in matrix analysis. This topic covers norms for vectors and matrices, sequences of vector/matrices, and series of matrices. They are important in the analysis of iterative processes and many other applications. See [7] for more information.

Definition 2.2.1. A **normed vector space** is a vector space $\mathcal{X}$ over a field $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$ together with a function $\|\cdot\| : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, called a **norm**, satisfying the following conditions:

(A1) $\|x\| \geqslant 0$ for all $x \in \mathcal{X}$.

(A2) $\|x\| = 0$ if and only if $x = 0$.

(A3) $\|\alpha x\| = |\alpha|\|x\|$ for all $x \in \mathcal{X}$ and $\alpha \in \mathbb{F}$.

(A4) $\|x + y\| \leqslant \|x\| + \|y\|$ for all $x, y \in \mathcal{X}$.

Example 2.2.2. The Euclidean norm ($l_2 - norm$) of a vector $x = [x_1 ... x_n]^T \in \mathbb{C}^n$,

$$\| x \|_2 = \left(| x_1 |^2 + ... + | x_n |^2 \right)^{1/2}$$

is perhaps the most familiar norm, since $\| x - y \|_2$ measures the standard Euclidean distance between two point $x, y \in \mathbb{C}^n$.

Definition 2.2.3. A function $|||\cdot|||: M_n \to R$ is a **matrix norm** if, for all $A, B \in \mathbb{M}_n$, it satisfies the following five axioms:

(S1) $||| A ||| \geqslant 0$ for all $A \in \mathbb{M}_n$.

(S2) $||| A ||| = 0$ if and only if $A = 0$.

(S3) $||| \alpha A ||| = |\alpha| \, ||| A |||$ for all $A \in \mathbb{M}_n$.and $\alpha \in \mathbb{F}$.

(S4) $||| A + B ||| \leqslant ||| A ||| + ||| B |||$ for all $A, B \in \mathbb{M}_n$.

(S5) $||| AB ||| \leqslant ||| A ||| \, ||| B |||$ for all $A, B \in \mathbb{M}_n$.

Example 2.2.4. The Frobenius norm of $A \in \mathbb{M}_{m,n}$ defined by

$$\|A\|_F = \mid trAA^* \mid^{1/2} = \left(\sum_{i,j=1}^{n} \mid a_{ij} \mid^2 \right)^{1/2}.$$

The Frobenius norm is a matrix norm because

$$\begin{aligned} \|A\|_F \|B\|_F &= \left(\sum_{i,j=1}^{n} \mid \sum_{k=1}^{n} (a_{ik} b_{kj} \mid^2 \right)^{1/2} \\ &\leq \left(\sum_{i,j=1}^{n} \left(\sum_{k=1}^{n} \mid a_{ik} \mid^2 \right) \left(\sum_{m=1}^{n} \mid b_{mj} \mid^2 \right) \right) \\ &= \left(\sum_{i,k=1}^{n} \mid a_{ik} \mid^2 \right)^{1/2} \left(\sum_{m,j=1}^{n} \mid b_{mj} \mid^2 \right)^{1/2} \\ &= \|A\|_F \|B\|_F. \end{aligned}$$

Lemma 2.2.5. *(e.g. [7]) For any matrices A and B with proper sizes, we have*

(i) $\|A^T A\|_2 = \|A\|_2^2$,

(ii) $\|AB\|_F \leqslant \|A\|_2 \|B\|_F$,

(iii) $\|A \otimes B\|_2 = \|A\|_2 \|B\|_2$,

(v) $\mathrm{tr}(PA^T A) \leqslant \lambda_{max}(P)\|A\|_F^2$, *where P is any positive definite matrix with proper size.*

Definition 2.2.6. Let $A \in \mathbb{M}_n$, The **singular values** of A are the square roots of the eigenvalues of AA^*.

Example 2.2.7. The spectral norm $\|.\|_2$ is defined on $\mathbb{M}_n$ by

$$\|A\|_2 = \sigma(A),$$

the largest singular value of A. Alternatively, we have the following formula

$$\|A\|_2 = \sqrt{\lambda_{max}(A^T A)}.$$

The spectral norm is a matrix norm.

Definition 2.2.8. Let $\{A_n\}_{n=1}^{\infty}$ be a sequence in $\mathbb{M}_n$ and $L \in \mathbb{M}_{m,n}$. We say that $\lim_{n\to\infty} A_n = L$ if for every $\epsilon > 0$ there is a natural number $N > 0$ so that whenever $n > N, \| A_n - L \| < \epsilon$. If $\lim_{n\to\infty} A_n = L$ we say that the sequence converges, otherwise it diverges.

Definition 2.2.9. A complex matrix A is called convergent if its powers converge to the zero matrix, $A^k \to 0$, meaning that the individual entries of A^k all go to 0 as $k \to \infty$.

Definition 2.2.10. The **spectral radius** of a matrix $A \in \mathbb{M}_n$ is defined as the maximal modulus of all of its eigenvalues : $\rho(A) = \max\{| \lambda_1 |, ..., | \lambda_k |\}$ where $\lambda_1, ..., \lambda_k$ are eigenvalue of A.

Theorem 2.2.11. [10] *The matrix A is convergent if and only if its spectral radius is strictly less than one:* $\rho(A) < 1$.

Theorem 2.2.12. [10] *The spectral radius of a matrix is bounded by its matrix norm:*

$$\rho(A) \leq \| A \| .$$

Definition 2.2.13. If $\{A_n\}_{n=1}^{\infty}$ is a sequence of complex square matrices of the same size, then the associated series is

$$\sum_{n=o}^{\infty} A_n = A_0 + A_1 + A_2 + ...$$

and the sequence of partial sums $\{S_k\}_{k=1}^{\infty}$ is defined by

$$S_k = \sum_{n=0}^{k} A_n$$

for each $k \in \mathbb{N}$. So

$$S_0 = A_0, \ S_1 = A_0 + A_1, \ S_2 = A_0 + A_1 + A_2, \ ...$$

A series converges if the sequence of partial sums converges, and otherwise the series diverges. In this case, we call $\lim_{k\to\infty} S_k$ the sum of the series $\sum_{n=0}^{\infty} A_n$, denoted by

$$\sum_{n=0}^{\infty} A_n = \lim_{k\to\infty} S_k.$$

2.3 Linear iterative systems

Iteration of linear systems is important in applied mathematics and numerical analysis. We discuss convergence criteria based on the spectral radius.

Definition 2.3.1. A linear iterative system takes the form

$$u^{(k+1)} = Au^{(k)}, \ u^{(0)} = a, \tag{2.3}$$

where $A \in \mathbb{M}n$, the iterates $u^{(k)}$ are vector in $\mathbb{C}^n$ for $k = 1, 2, 3, ...$ the and solution $u^{(k)}$ is uniquely determined by the initial condition $u^{(0)} = a$

Theorem 2.3.2. [10] *If the coefficient matrix A is diagonalizable, then the general solution to the linear iterative system $u^{(k+1)} = Au^{(k)}$ is given by*

$$u^{(k)} = c_1\lambda_1^k v_1 + c_2\lambda_2^k v_2 + ... + c_n\lambda_n^k v_n. \tag{2.4}$$

where $v_1, ..., v_n$ are the linearly independent eigenvalue and $\lambda_1, ..., \lambda_n$ the corresponding eigenvalue of A. The coefficients $c_1, ..., c_n$ are arbitrary scalars and are uniquely prescribed by initial conditions $u^{(0)} = a$.

If A is diagonalizable, then we can apply the triangle inequality to (2.6) estimate

$$\begin{aligned} \| u^{(k)} \| &= \| c_1\lambda_1^k v_1 + c_2\lambda_2^k v_2 + ... + c_n\lambda_n^k v_n \| \\ &\leq | \lambda_1 |^k \| c_1 v_1 \| + ... + | \lambda_n |^k \| c_n v_n \| \\ &\leq \rho(A)^k (| c_1 | \| v_1 \| + ... + | c_n | \| v_n \|) = C\rho(A)^k, \end{aligned}$$

for some constant $C > 0$ that depends only upon the initial conditions.In particular, if $\rho(A) < 1$, then

$$\| u^{(k)} \| = C\rho(A)^k \longrightarrow 0 \ as \ k \longrightarrow \infty,$$

in accordance with Theorem 2.3.2 Thus, the spectral radius prescribes the rate of convergence of the solution to equilibrium. The smaller the spectral radius, the faster the solution go to 0.

2.4 Gradients and hierarchical identification principle

Definition 2.4.1. Differential of scalar-by-matrix The derivative of a scalar y function of a $p \times q$ matrix X of independent variables, with respect to the matrix X, is given by

$$\frac{\partial y}{\partial X} = \begin{bmatrix} \frac{\partial y}{\partial x_{11}} & \frac{\partial y}{\partial x_{12}} & \cdots & \frac{\partial y}{\partial x_{1q}} \\ \frac{\partial y}{\partial x_{21}} & \frac{\partial y}{\partial x_{22}} & \cdots & \frac{\partial y}{\partial x_{2q}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{p1}} & \frac{\partial y}{\partial x_{p2}} & \cdots & \frac{\partial y}{\partial x_{pq}} \end{bmatrix}.$$

Proposition 2.4.2. *(e.g. [7]) **Derivatives of the Trace** For any matrices A, B and X with proper sizes, we have*

(i) $\frac{\partial tr(AX)}{\partial X} = \frac{\partial tr(XA)}{\partial X} = A^T,$

(ii) $\frac{\partial tr(X^TA)}{\partial X} = \frac{\partial tr(AX^T)}{\partial X} = A,$

(iii) $\frac{\partial tr(XAX^TB)}{\partial X} = B^TXA^T + BXA.$

Lemma 2.4.3. *(e.g. [5]) For any real number a, b with $b > a > 0$, we have*

$$\min_{0<x<\frac{2}{b}} \{\max\{| 1 - ax |, | 1 - bx |\}\} = \frac{b-a}{b+a}.$$

The minimality is reached at $x_{opt} = \frac{2}{a+b}$.

Chapter 3

Relaxed gradient based iterative algorithm and convergence analysis

3.1 Proposing the Relaxed Gradient Based Iterative Algorithm for Generalized Sylvester Equation

In this section, we propose an iterative algorithm for solving the equation (1.4), called the relaxed gradient based iterative algorithm.

Let $A, B, C, D, F \in \mathbb{M}_n(\mathbb{R})$. We recall the hierarchical identification principle in [12] and apply the principle to solve the generalized Sylvester matrix equation. Define two matrices

$$M := F - CXD \quad \text{and} \quad N := F - AXB. \tag{3.1}$$

From (1.4), we get two subsystems

$$AXB = M \quad \text{and} \quad CXD = N. \tag{3.2}$$

We would like to minimize the following quadratic functions:

$$L_1(X) := \|AXB - M\|_F^2, \tag{3.3}$$

$$L_2(X) := \|CXD - N\|_F^2. \tag{3.4}$$

Now, for (3.1) and (3.2), we will deduce their gradients as follows:

$$\begin{aligned}
\frac{\partial}{\partial X} L_1(X) &= \frac{\partial}{\partial X} \operatorname{tr}[(AXB - M)^T (AXB - M)] \\
&= \frac{\partial}{\partial X} \operatorname{tr}[(B^T X^T A^T - M^T)(AXB - M)] \\
&= \frac{\partial}{\partial X} \operatorname{tr}[B^T X^T A^T AXB - B^T X^T A^T M - M^T AXB + M^T M] \\
&= \frac{\partial}{\partial X} \operatorname{tr}(B^T X^T A^T AXB) - \frac{\partial}{\partial X} \operatorname{tr}(B^T X^T A^T M) - \frac{\partial}{\partial X} \operatorname{tr}(M^T AXB) + \frac{\partial}{\partial X} \operatorname{tr}(M^T M) \\
&= \frac{\partial}{\partial X} \operatorname{tr}(XBB^T X^T A^T A) - \frac{\partial}{\partial X} \operatorname{tr}(X^T A^T MB^T) - \frac{\partial}{\partial X} \operatorname{tr}(BM^T AX) \\
&= (A^T A)^T X(BB^T) + A^T AXBB^T - A^T MB^T - (BM^T A)^T \\
&= 2A^T (AXB - M)B^T.
\end{aligned} \tag{3.5}$$

And

$$\begin{aligned}
\frac{\partial}{\partial X}L_2(X) &= \frac{\partial}{\partial X}\mathrm{tr}[(CXD-N)^T(CXD-N)] \\
&= \frac{\partial}{\partial X}\mathrm{tr}[(D^TX^TC^T-N^T)(CXD-N)] \\
&= \frac{\partial}{\partial X}\mathrm{tr}[D^TX^TC^TCXD-D^TX^TC^TN-N^TCXD+N^TN] \\
&= \frac{\partial}{\partial X}\mathrm{tr}(D^TX^TC^TCXD)-\frac{\partial}{\partial X}\mathrm{tr}(D^TX^TC^TN)-\frac{\partial}{\partial X}\mathrm{tr}(N^TCXD)+\frac{\partial}{\partial X}\mathrm{tr}(N^TN) \\
&= \frac{\partial}{\partial X}\mathrm{tr}(XDD^TX^TC^TC)-\frac{\partial}{\partial X}\mathrm{tr}(X^TC^TND^T)-\frac{\partial}{\partial X}\mathrm{tr}(DN^TCX) \\
&= (C^TC)^TX(DD^T)+C^TCXDD^T-C^TND^T-(DN^TC)^T \\
&= 2C^T(CXD-N)D^T. \qquad (3.6)
\end{aligned}$$

Let $X_1(k), X_2(k)$ and $X(k)$ be the iterative solutions of the system (3.2) at iteration k. The recursive formulas of $X_1(k)$ and $X_2(k)$ come from the gradient formulas (3.5) and (3.6) as follows:

$$\begin{aligned}
X_1(k) &= X(k-1)+\tau(1-\omega)A^T[M-AX(k-1)B]B^T \\
&= X(k-1)+\tau(1-\omega)A^T[F-AX(k-1)B-CXD]B^T, \\
X_2(k) &= X(k-1)+\tau\omega C^T[N-CX(k-1)D]D^T \\
&= X(k-1)+\tau\omega C^T[F-AXB-CX(k-1)D]D^T.
\end{aligned}$$

Here, the convergence factor τ is introduced to control the convergence of the algorithm. Base on the hierarchical identification principle, the unknown variable X is replaced by its estimates at iteration $k-1$. The iterative solution $X(k)$ is now the weighted arithmetic mean $\omega X_1(k)+(1-\omega)X_2(k)$, here $\omega \in (0,1)$ is called the relaxation parameter. The above discussion leads to the following algorithm for solving the generalized Sylvester equation (1.4).

Algorithm 3.1.1. The relaxed gradient based iterative (RGI) algorithm.

Step 1 Input matrices $A, B, C, D, F \in \mathbb{M}_n(\mathbb{R})$, given any small positive number ε and appropriative positive number ω. Choose the initial matrices $X_1(0)$ and $X_2(0)$. Set $k := 1$.

Step 2 If $\delta_k := \|AX(k)B+CX(k)D-F\|_F/\|F\|_F < \varepsilon$, stop; otherwise, go to Step 3.

Step 3 Update the sequences

$$X_1(k) = X(k-1)+(1-\omega)\tau A^T[F-AX(k-1)B-CX(k-1)D]B^T, \qquad (3.7)$$

$$X_2(k) = X(k-1)+\omega\tau C^T[F-AX(k-1)B-CX(k-1)D]D^T, \qquad (3.8)$$

$$X(k) = \omega X_1(k)+(1-\omega)X_2(k). \qquad (3.9)$$

Step 4 Set $k := k+1$, return to Step 2.

3.2 Convergence Analysis of the Proposed Algorithm

In this section, we shall discuss a criterion for the applicability of the algorithm, and the optimal parameter for the fastest convergence.

The next theorem shows the applicability of Algorithm 3.1.1.

Theorem 3.2.1. *Assume that the matrix equation $AXB + CXD = F$ has a unique solution X. If the parameter τ is chosen so that*

$$0 < \tau < \frac{2}{\omega(1-\omega)\left(\|A\|_2\|B\|_2 + \|C\|_2\|D\|_2\right)^2} := \tau_0,$$

then the iterative sequence $X(k)$ generated by Algorithm 3.1.1 converges to X, i.e., $\lim_{k\to\infty} X(k) = X$ for any initial value $X_1(0)$ and $X_2(0)$.

Proof. Define the error matrices

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

and

$$\theta_1(k) := A\widehat{X}(k-1)B,$$
$$\theta_2(k) := C\widehat{X}(k-1)D.$$

We shall show that $\lim_{k\to\infty} \widehat{X} = 0$. From Algorithm 3.1.1, we get

$$\begin{aligned}\widehat{X}_1(k) &= \widehat{X}(k-1) + (1-\omega)\tau A^T(F - AX(k-1)B - CX(k-1)D)B^T,\\ &= \widehat{X}(k-1) + (1-\omega)\tau A^T(-\theta_1(k) - \theta_2(k))B^T,\\ &= \widehat{X}(k-1) - (1-\omega)\tau A^T(\theta_1(k) + \theta_2(k))B^T,\end{aligned}$$

and

$$\begin{aligned}\widehat{X}_2(k) &= \widehat{X}(k-1) + \omega\tau C^T(F - AX(k-1)B - CX(k-1)D)D^T,\\ &= \widehat{X}(k-1) + \omega\tau C^T(-\theta_1(k) - \theta_2(k))D^T,\\ &= \widehat{X}(k-1) - \omega\tau C^T(\theta_1(k) + \theta_2(k))D^T.\end{aligned}$$

Hence,

$$\begin{aligned}
\widehat{X}(k) &= X(k) - X \\
&= \omega X_1(k) + (1-\omega)X_2(k) - X \\
&= \omega(\widehat{X}_1(k) + X) + (1-\omega)(\widehat{X}_2(k) + X) - X \\
&= \omega\widehat{X}_1(k) + (1-\omega)\widehat{X}_2(k) \\
&= \widehat{X}(k-1) - \omega(1-\omega)\tau[A^T(\theta_1(k) + \theta_2(k))B^T + C^T(\theta_1(k) + \theta_2(k))D^T].
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|\widehat{X}(k)\|_F^2 &= \mathrm{tr}\left(\widehat{X}(k)\widehat{X}(k)^T\right) \\
&= \mathrm{tr}\{\left[\widehat{X}(k-1) - \omega(1-\omega)\tau[A^T(\theta_1(k) + \theta_2(k))B^T + C^T(\theta_1(k) + \theta_2(k))D^T]\right] \times \\
&\quad \left[\widehat{X}(k-1) - \omega(1-\omega)\tau[A^T(\theta_1(k) + \theta_2(k))B^T + C^T(\theta_1(k) + \theta_2(k))D^T]\right]^T\} \\
&= \mathrm{tr}\{\left[\widehat{X}(k-1) - \omega(1-\omega)\tau[A^T(\theta_1(k) + \theta_2(k))B^T + C^T(\theta_1(k) + \theta_2(k))D^T]\right] \times \\
&\quad \left[\widehat{X}(k-1)^T - \omega(1-\omega)\tau[B(\theta_1(k)^T + \theta_2(k)^T)A + D(\theta_1(k)^T + \theta_2(k)^T)C]\right]^T\} \\
&= \mathrm{tr}\left(\widehat{X}(k-1)\widehat{X}(k-1)^T\right) \\
&\quad - \omega(1-\omega)\tau\,\mathrm{tr}\left(\widehat{X}(k-1)[B(\theta_1(k)^T + \theta_2(k)^T)A + D(\theta_1(k)^T + \theta_2(k)^T)C]\right) \\
&\quad - \omega(1-\omega)\tau\,\mathrm{tr}\left(\widehat{X}(k-1)^T[A^T(\theta_1(k) + \theta_2(k))B^T + C^T(\theta_1(k) + \theta_2(k))D^T]\right) \\
&\quad + \omega^2(1-\omega)^2\tau^2\,\mathrm{tr}([A^T(\theta_1(k) + \theta_2(k))B^T + C^T(\theta_1(k) + \theta_2(k))D^T] \times \\
&\quad [B(\theta_1(k)^T + \theta_2(k)^T)A + D(\theta_1(k)^T + \theta_2(k)^T)C]) \\
&= \|\widehat{X}(k-1)\|_F^2 - 2\omega(1-\omega)\tau\|\theta_1(k) + \theta_2(k)\|_F^2 \\
&\quad + \omega^2(1-\omega)^2\tau^2\|A^T(\theta_1(k) + \theta_2(k))B^T + C^T(\theta_1(k) + \theta_2(k))D^T\|_F^2 \\
&= \|\widehat{X}(k-1)\|_F^2 - 2\omega(1-\omega)\tau\|\theta_1(k) + \theta_2(k)\|_F^2 + \omega^2(1-\omega)^2\tau^2 \times \\
&\quad \mathrm{tr}[(A^TB^T + C^TD^T)(BA + DC)(\theta_1(k)^T + \theta_2(k)^T)(\theta_1(k) + \theta_2(k))] \\
&\leqslant \|\widehat{X}(k-1)\|_F^2 - 2\omega(1-\omega)\tau\|\theta_1(k) + \theta_2(k)\|_F^2 \\
&\quad + \omega^2(1-\omega)^2\tau^2\lambda_{max}[(A^TB^T + C^TD^T)(BA + DC)]\|\theta_1(k) + \theta_2(k)\|_F^2 \\
&\leqslant \|\widehat{X}(k-1)\|_F^2 - 2\omega(1-\omega)\tau\|\theta_1(k) + \theta_2(k)\|_F^2 \\
&\quad + \omega^2(1-\omega)^2\tau^2\|A^TB^T + C^TD^T\|_2^2\|\theta_1(k) + \theta_2(k)\|_F^2 \\
&\leqslant \|\widehat{X}(k-1)\|_F^2 - 2\omega(1-\omega)\tau\|\theta_1(k) + \theta_2(k)\|_F^2 \\
&\quad + \omega^2(1-\omega)^2\tau^2\left(\|AB\|_2 + \|CD\|_2\right)^2\|\theta_1(k) + \theta_2(k)\|_F^2
\end{aligned}$$

$$
\begin{aligned}
&\leqslant \|\widehat{X}(k-1)\|_F^2 - 2\omega(1-\omega)\tau\|\theta_1(k)+\theta_2(k)\|_F^2 \\
&\quad + \omega^2(1-\omega)^2\tau^2\left(\|A\|_2\|B\|_2 + \|C\|_2\|D\|_2\right)^2\|\theta_1(k)+\theta_2(k)\|_F^2 \\
&= \|\widehat{X}(k-1)\|_F^2 - \Big(2\omega(1-\omega)\tau - \omega^2(1-\omega)^2\tau^2\left[\|A\|_2\|B\|_2 + \|C\|_2\|D\|_2\right]^2\Big)\times \\
&\quad \|\theta_1(k)+\theta_2(k)\|_F^2
\end{aligned}
$$

By induction, we have

$$\|\widehat{X}(k)\|_F^2 \leqslant \|\widehat{X}(0)\|_F^2 - \gamma\sum_{i=1}^{k}\|\theta_1(i)+\theta_2(i)\|_F^2,$$

where $\gamma = 2\omega(1-\omega)\tau - \omega^2(1-\omega)^2\tau^2(\|A\|_2\|B\|_2 + \|C\|_2\|D\|_2)^2$. Since $0 < \tau < \tau_0$, we have $\gamma > 0$. Now, we have

$$\sum_{i=1}^{k}\|\theta_1(i)+\theta_2(i)\|_F^2 \leqslant \frac{1}{\gamma}\Big(\|\widehat{X}(0)\|_F^2 - \|\widehat{X}(k)\|_F^2\Big) \leqslant \frac{1}{\gamma}\|\widehat{X}(0)\|_F^2,$$

and thus

$$\sum_{i=1}^{\infty}\|\,\theta_1(i)+\theta_2(i)\,\|_F^2 < \infty.$$

It implies that $\theta_1(k)+\theta_2(k) \to 0$ as $k \to \infty$, that is,

$$A\widehat{X}(k-1)B + C\widehat{X}(k-1)D \to 0 \; as \; k \to \infty.$$

Note that the uniqueness of the solution for (1.4) implies the invertibility of the matrix $B^T \otimes A + D^T \otimes C$. Now, $(B^T \otimes A + D^T \otimes C)\,\mathrm{vec}\,\widehat{X}(k-1) \to 0$ as $k \to \infty$ and, hence, $\widehat{X}(k-1) \to 0$ as $k \to \infty$. Therefore, $\lim_{k\to\infty}\widehat{X}(k) = 0$. □

Remark 3.2.2. When $\omega = \frac{1}{2}$, the range of parameters τ in Theorem 3.2.1 is wider than that of given in (1.7).

Theorem 3.2.3. *Suppose the matrix $\Psi := B^T \otimes A + D^T \otimes C$ is invertible. Then, the optimal factor for which Algorithm 3.1.1 converges for any initial value is given by*

$$\tau_{opt} = \frac{2}{\omega(1-\omega)(\lambda_{min}(\Psi^T\Psi) + \lambda_{max}(\Psi^T\Psi))}. \tag{3.10}$$

Proof. Denote the error matrix $\widehat{X}(k) = X(k) - X$. From the proof of Theorem 3.2.1 , we have

$$
\begin{aligned}
\widehat{X}(k) &= \omega X_1(k) + (1-\omega)X_2(k), \\
&= \widehat{X}(k-1) - \omega(1-\omega)\tau[A^T(A\widehat{X}(k-1)B + C\widehat{X}(k-1)D]B^T \\
&\quad - \omega(1-\omega)\tau[C^T(A\widehat{X}(k-1) + C\widehat{X}(k-1)D]D^T.
\end{aligned}
$$

By applying the vector operator, we now obtain

$$\begin{aligned}
&\operatorname{vec}\widehat{X}(k)\\
&\quad= \operatorname{vec}\widehat{X}(k-1)-\tau\omega(1-\omega)[\operatorname{vec}(A^TA\widehat{X}(k-1)BB^T)+\operatorname{vec}(A^TC\widehat{X}(k-1)DB^T)\\
&\qquad+\operatorname{vec}(C^TA\widehat{X}(k-1)BD^T)+\operatorname{vec}(C^TC\widehat{X}(k-1)DD^T)]\\
&\quad= \operatorname{vec}\widehat{X}(k-1)-\tau\omega(1-\omega)([BB^T\otimes A^TA]\operatorname{vec}\widehat{X}+[BD^T\otimes A^TC]\operatorname{vec}\widehat{X}\\
&\qquad+[DB^T\otimes C^TA]\operatorname{vec}\widehat{X}+[DD^T\otimes C^TC]\operatorname{vec}\widehat{X})\\
&\quad= \{I_{n^2}-\tau\omega(1-\omega)[(B\otimes A^T)(B^T\otimes A)+(B\otimes A^T)(D^T\otimes C)\\
&\qquad+(D\otimes C^T)(B^T\otimes A)+(D\otimes C^T)(D^T\otimes C)]\}\operatorname{vec}\widehat{X}(k-1)\\
&\quad= \{I_{n^2}-\tau\omega(1-\omega)[(B^T\otimes A+D^T\otimes C)^T(B^T\otimes A+D^T\otimes C)]\}\operatorname{vec}\widehat{X}(k-1)\\
&\quad= \{I_{n^2}-\tau\omega(1-\omega)\Psi^T\Psi\}\operatorname{vec}\widehat{X}(k-1). \qquad (3.11)
\end{aligned}$$

Since $x^T\Psi^T\Psi x=(\Psi x)^T(\Psi x)\geqslant 0$ for each $x\in\mathbb{R}^{n^2}$, the matrix $\Psi^T\Psi$ is positive semidefinite. Since Ψ is invertible, we have that $\Psi^T\Psi$ is positive definite. The spectral radius of the iteration matrix is given by

$$\begin{aligned}
&\rho(I_{n^2}-\tau\omega(1-\omega)\Psi^T\Psi)\\
&\quad=\max\{|\,1-\tau\omega(1-\omega)\lambda_{min}(\Psi^T\Psi)\,|,|\,1-\tau\omega(1-\omega)\lambda_{\max}(\Psi^T\Psi)\,|\}. \qquad (3.12)
\end{aligned}$$

Thus, the minimality of this spectral radius subject to the condition $0<\tau<\tau_0$ in the view of Lemma 2.4.3 is determined by (3.10). □

Chapter 4

Iterative methods for special cases of the generalized Sylvester equation

In this section, we discuss iterative algorithms for solving interesting special cases of (1.4)

4.1 The matrix equation $AXB = C$

The following one is for solving $AXB = C$.

Algorithm 4.1.1. The relaxed gradient based iterative (RGI) algorithm for solving (1.5).

Step 1 Input matrices $A, B, C \in \mathbb{M}_n(\mathbb{R})$, given any small positive number ε and appropriative positive number ω. Choose the initial matrix $X(0)$. Set $k := 1$.

Step 2 If $\delta_k := \|AX(k)B - C\|_F / \|C\|_F < \varepsilon$, stop; otherwise, go to Step 3.

Step 3 Update the sequences

$$X(k) = \omega[X(k-1) + (1-\omega)\tau A^T(C - AX(k-1) - X(k-1)B)B^T] \tag{4.1}$$

Step 4 Set $k := k + 1$, return to Step 2.

Corollary 4.1.2. *Assume that (1.5) has a unique solution X. If the parameter τ is chosen so that*

$$0 < \tau < \frac{2}{\omega(1-\omega)(\|A\|_2^2\|B\|_2^2)},$$

then the iterative sequence $X(k)$ generated by Algorithm 4.1.1 converges to X, for any initial value $X(0)$.

Proof. From Theorem 3.2.1, put $C = 0$ and $F = C$. □

4.2 The Sylvester matrix equation

The algorithm is proposed for solving the Sylvester matrix equation (1.3).

Algorithm 4.2.1. The relaxed gradient based iterative (RGI) algorithm for solving (1.3).

Step 1 Input matrices $A, B, C \in \mathbb{M}_n(\mathbb{R})$, given any small positive number ε and appropriative positive number ω. Choose the initial matrices $X_1(0)$ and $X_2(0)$. Set $k := 1$.

Step 2 If $\delta_k := \|AX(k) + X(k)B - C\|_F / \|C\|_F < \varepsilon$, stop; otherwise, go to Step 3.

Step 3 Update the sequences

$$X_1(k) = X(k-1) + (1-\omega)\tau A^T[C - AX(k-1) - X(k-1)B], \tag{4.2}$$

$$X_2(k) = X(k-1) + \omega\tau[C - AX(k-1) - X(k-1)B]B^T, \tag{4.3}$$

$$X(k) = \omega X_1(k) + (1-\omega)X_2(k). \tag{4.4}$$

Step 4 Set $k := k+1$, return to Step 2.

Corollary 4.2.2. *Assume that* (1.3) *has a unique solution* X. *If the parameter* τ *is chosen so that*

$$0 < \tau < \frac{2}{\omega(1-\omega)(\|A\|_2 + \|B\|_2)^2},$$

then the iterative sequence $X(k)$ *generated by Algorithm 4.2.1 converges to* X, *for any initial value* $X_1(0)$ *and* $X_2(0)$.

Proof. It follows from Theorem 3.2.1 by setting $B = C = I, D = B$ and $F = C$. □

4.3 The discrete-time Sylvester equation matrix equation

The algorithm is proposed for solving the discrete-time Sylvester equation (1.6).

Algorithm 4.3.1. The relaxed gradient based iterative (RGI) algorithm for solving (1.6).

Step 1 Input matrices $A, B, C \in \mathbb{M}_n(\mathbb{R})$, given any small positive number ε and appropriative positive number ω. Choose the initial matrices $X_1(0)$ and $X_2(0)$. Set $k := 1$.

Step 2 If $\delta_k := \|AX(k)B + X(k) - C\|_F / \|C\|_F < \varepsilon$, stop; otherwise, go to Step 3.

Step 3 Update the sequences

$$X_1(k) = X(k-1) + (1-\omega)\tau A^T[C - AX(k-1)B - X(k-1)]B^T, \tag{4.5}$$

$$X_2(k) = X(k-1) + \omega\tau[C - AX(k-1)B - X(k-1)]. \tag{4.6}$$

$$X(k) = \omega X_1(k) + (1-\omega)X_2(k). \tag{4.7}$$

Step 4 Set $k := k+1$, return to Step 2.

Corollary 4.3.2. *Assume that* (1.6) *has a unique solution* X. *If the parameter* τ *is chosen so that*

$$0 < \tau < \frac{2}{\omega(1-\omega)(\|A\|_2\|B\|_2 + 1)^2},$$

then the iterative sequence $X(k)$ *generated by Algorithm 4.3.1 converges to* X, *for any initial value* $X_1(0)$ *and* $X_2(0)$.

Proof. It follows from Theorem 3.2.1 by setting $C = D = I$ and $F = C$. □

Chapter 5

Numerical Simulation

In this section, we report numerical results to illustrate the effectiveness of Algorithm 3.1.1. All iterations have been carried out by MATLAB 724504-R2013a, Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz 3.19 GHz, RAM 8.00 GB. PC environment.

Example 5.0.3. We consider the equation (1.4) with

$$A = \begin{bmatrix} 1 & 2 & 6 & 1 & 2 & -1 \\ 3 & 5 & 7 & 3 & 5 & 0 \\ 4 & 8 & 9 & 4 & 8 & 1 \\ 1 & 2 & 6 & 3 & 5 & 7 \\ 1 & 1 & 7 & 2 & 7 & -1 \\ 3 & 6 & -5 & 0 & 6 & -5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 6 & -1 & 0 & 3 \\ 1 & 6 & -1 & 2 & 0 & 4 \\ 1 & 2 & -7 & 0 & -2 & 5 \\ -3 & 2 & 7 & -5 & 5 & -7 \\ 1 & -1 & -5 & 6 & 0 & 1 \\ 2 & -1 & 0 & 1 & 2 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 2 & 9 & 1 & -5 & 0 \\ 6 & 2 & -6 & 2 & -1 & 1 \\ 1 & 1 & 1 & 3 & 2 & -2 \\ 0 & -3 & -1 & 0 & -1 & -5 \\ 1 & 3 & 1 & 0 & 1 & 2 \\ 0 & 0 & 7 & 5 & 2 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 4 & 2 & 5 & 7 & 3 \\ 9 & 1 & 6 & 4 & 2 & 8 \\ 6 & 1 & 2 & 5 & 3 & 7 \\ 1 & 1 & 1 & -1 & 2 & 7 \\ 3 & -1 & 1 & 4 & 2 & 3 \\ -11 & -8 & -3 & 10 & 1 & 0 \end{bmatrix}$$

and $F = \begin{bmatrix} 4720 & 1899 & -364 & 3721 & 1904 & 7326 \\ -1684 & 1258 & -2938 & -1339 & -1135 & -242 \\ 1095 & 1919 & -2273 & 876 & 40 & 3607 \\ 213 & 1277 & -1347 & -1043 & -546 & 367 \\ 878 & 1358 & -2550 & 1251 & -182 & 3370 \\ 2875 & -452 & 3236 & 3177 & 2185 & 3400 \end{bmatrix}$.

Choose $\omega = 0.2$ and initial matrices

$$X_1(0) = 10^{-6} \begin{bmatrix} 1 & 0 & -1 & 1 & 0 & 2 \\ 2 & 4 & 3 & 1 & 5 & 7 \\ 0 & 8 & 10 & 5 & 2 & 0 \\ 4 & 9 & 11 & 7 & -2 & -3 \\ -1 & 1 & 2 & 4 & 5 & 6 \\ 2 & 1 & -1 & 2 & 1 & 1 \end{bmatrix} \text{ and } X_2(0) = 10^{-6} \begin{bmatrix} 4 & 2 & 1 & -1 & 1 & 0 \\ 8 & 7 & 3 & 0 & 4 & 5 \\ -3 & -4 & 0 & -1 & 1 & 2 \\ 6 & 5 & 3 & 0 & -1 & -2 \\ 7 & 5 & -1 & 0 & 9 & 10 \\ -12 & 11 & 4 & -5 & -6 & 7 \end{bmatrix}.$$

Then, the convergence factor is given by

$$0 < \tau < \frac{2}{\omega(1-\omega)\left(\|A\|_2\|B\|_2 + \|C\|_2\|D\|_2\right)^2} \approx 6.0026 \times 10^{-5}.$$

Running Algorithm 3.1.1, we obtain that the approximated solutions converge to the exact solution

$$X^* = \begin{bmatrix} 6 & 5 & 3 & -2 & 1 & 0 \\ 4 & 2 & 1 & 1 & -1 & 8 \\ 3 & 10 & 58 & 6 & 4 & 2 \\ 0 & 1 & -1 & 9 & 7 & 4 \\ 2 & 3 & -7 & -2 & -3 & 5 \\ 6 & -4 & 0 & 10 & 4 & 1 \end{bmatrix},$$

although the initial matrices are very closed to the zero matrix. Clearly, as k increases, the term $\|X(k) - X\|_F / \|X\|_F$ becomes smaller and goes to zero. Fig 5.1 shows the relative errors for the algorithm with different convergence factors τ.

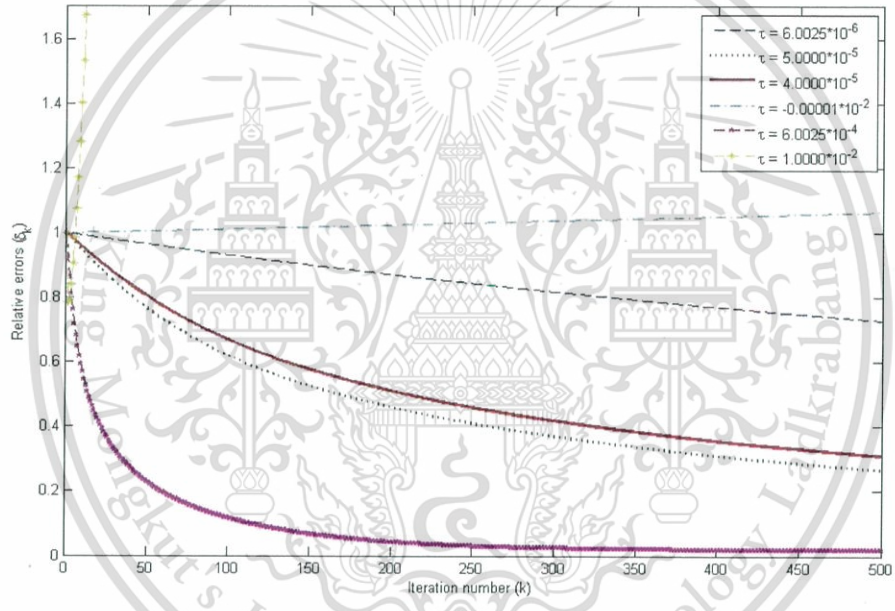


Figure 5.1: Relative errors for Example 5.0.3

Example 5.0.4. We consider the equation (1.4) with

$$A = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}, \; B = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; D = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \text{ and } F = \begin{bmatrix} 65 & 26 \\ 26 & 65 \end{bmatrix}.$$

Then, the matrix $\Psi^T \Psi$ is given by

$$\Psi^T \Psi = \begin{bmatrix} 14 & 3 & -3 & -1 \\ 3 & 14 & -1 & -3 \\ -3 & -1 & 14 & 3 \\ -1 & -3 & 3 & 614 \end{bmatrix}.$$

Choose $\omega = 0.7$ and initial matrices

$$X_1(0) = 10^{-6}\begin{bmatrix} 5 & 1 \\ 4 & 0 \end{bmatrix} \text{ and } X_2(0) = 10^{-6}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Then, the optimal convergence factor is given by

$$\tau_{opt} = \frac{2}{\omega(1-\omega)(\lambda_{min}(\Psi^T\Psi) + \lambda_{max}(\Psi^T\Psi))} \approx 0.0182.$$

Running Algorithm 3.1.1, we see in Table 5.1 that the approximated solutions converge to the exact solution

$$X^* = \begin{bmatrix} 5 & 2 \\ 2 & 5 \end{bmatrix},$$

although the initial matrices are very closed to the zero matrix. Clearly, as k increases, the term $\|X(k) - X\|_F / \|X\|_F$ becomes smaller and goes to zero. Fig 5.2 shows the relative errors for the algorithm with different convergence factors τ. We see that as k is large enough, the relative error for $\tau = 0.0182$ goes faster to 0 than those for another convergence factors.

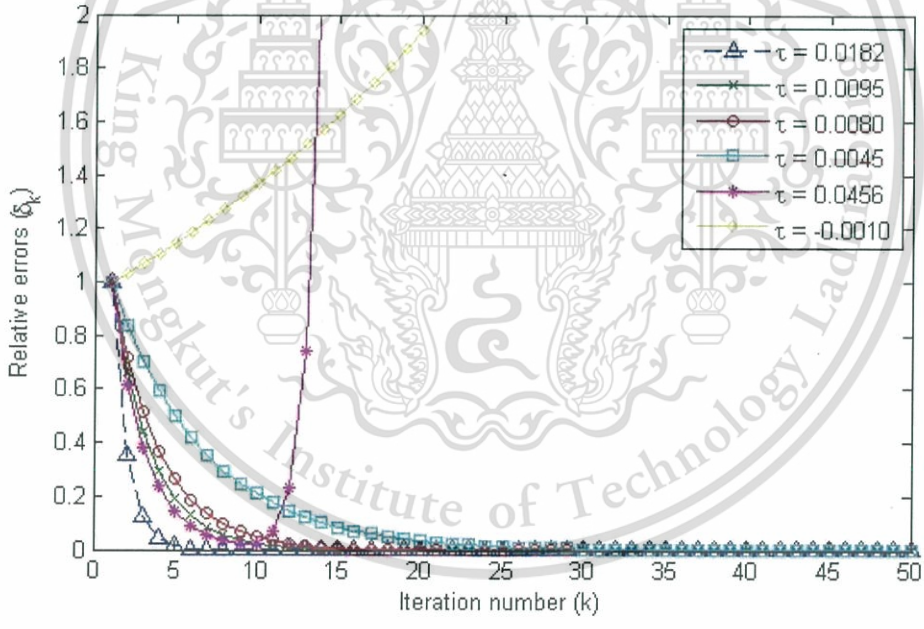


Figure 5.2: Relative errors for Example 5.0.4

Example 5.0.5. We consider the equation $AXB + CXD = F$ with

$$A = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix}, \; B = \begin{bmatrix} 4 & 2 & -2 & -2 \\ 8 & 0 & 2 & -2 \\ -2 & 2 & 2 & -4 \\ 6 & -2 & 4 & 2 \end{bmatrix}, \; C = \begin{bmatrix} -2 & 6 & 2 & 4 \\ 0 & 4 & 6 & 2 \\ 2 & -2 & 4 & -4 \\ 8 & 2 & 4 & -2 \end{bmatrix},$$

Table 5.1: Iterative solution ($\tau = 0.0182$)

k	x_{11}	x_{12}	x_{21}	x_{22}	$\|X(k) - X\|_F / \|X\|_F$
2	3.2296	1.2918	1.2918	3.2296	0.3541
4	4.7780	1.9112	1.9112	4.7780	0.0444
6	4.9722	1.9889	1.9889	4.9722	0.0056
8	4.9965	1.9986	1.9986	4.9965	$7.0000e-04$
10	4.9996	1.9998	1.9998	4.9996	$8.3045e-05$
Solution	5	2	2	5	

$$D = \begin{bmatrix} 4 & 8 & 2 & 2 \\ -2 & 10 & 0 & 2 \\ 2 & 2 & -2 & -2 \\ 4 & 4 & 6 & 2 \end{bmatrix} \text{ and } F = \begin{bmatrix} 456 & 1220 & -20 & 20 \\ 576 & 996 & 184 & 32 \\ 396 & -336 & 216 & -172 \\ 804 & 304 & 376 & -64 \end{bmatrix}.$$

We compare the efficiency Algorithm 3.1.1 with GI Algorithm. In both algorithms, we use the same convergence factor $\tau = 2.1323 \times 10^{-4}$ and initial matrices

$$X_1(0) = 10^{-6} \begin{bmatrix} -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix} \text{ and } X_2(0) = 10^{-6} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Table 5.2: Relative errors for Example 5.0.5

k	GI	RGI, $\omega = 0.55$	RGI, $\omega = 0.6$	RGI, $\omega = 0.65$	RGI, $\omega = 0.7$
20	0.7972	0.5441	0.1706	0.0550	0.0539
40	0.7633	0.3481	0.0451	0.0354	0.0372
60	0.7311	0.2230	0.0258	0.0264	0.0281
80	0.7005	0.1430	0.0195	0.0205	0.0221
100	0.6712	0.0919	0.0156	0.0165	0.0179

Fig 5.3 and Table 5.2 show the relative errors when running Algorithm 3.1.1 with different ralaxation parameters, comparing with GI Algorithm. We notice that the RGI Algorithm with $\omega = 0.55, 0.6, 0.65$ and 0.7 converge faster than the GI Algorithm.

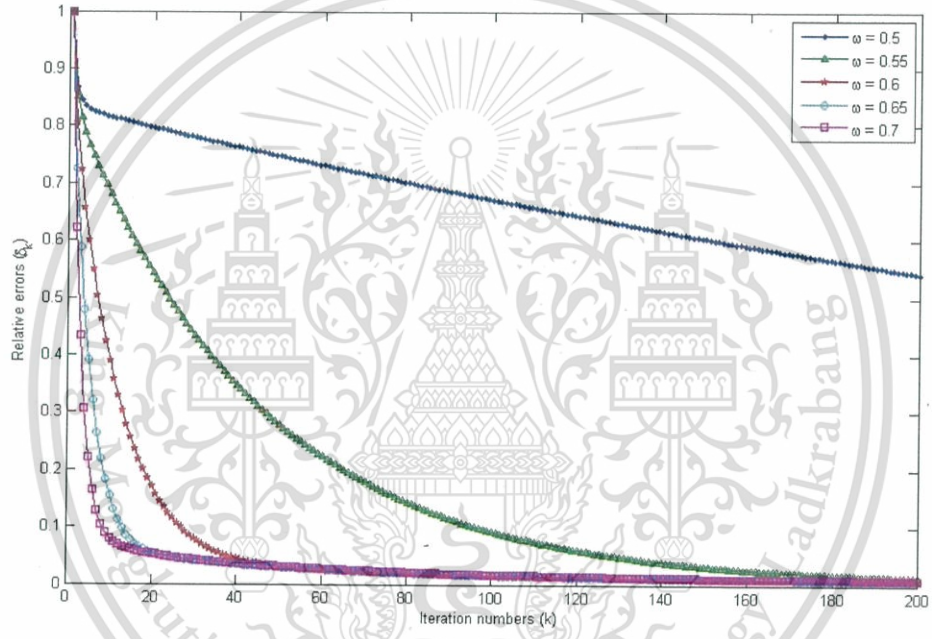
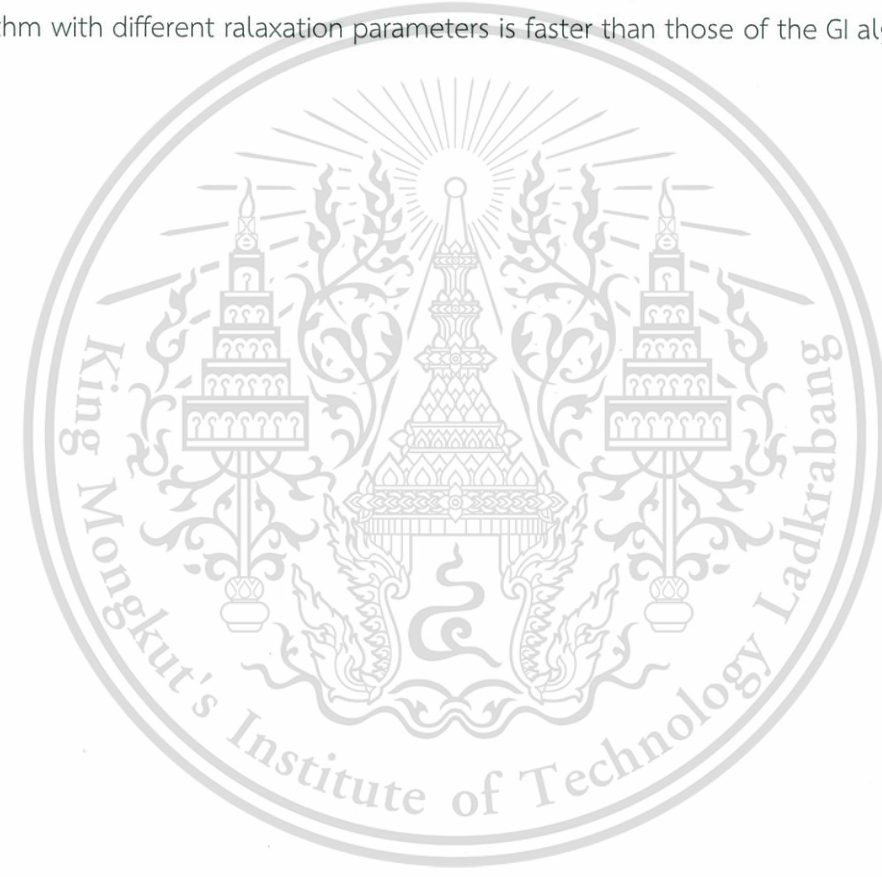


Figure 5.3: Relative errors for Example 5.0.5

Chapter 6

Conclusions

We propose a relaxed gradient based iterative algorithm for solving the generalized Sylvester equation (1.4). This algorithm guarantees that the approximated solutions converge to the exact solution for any given initial value. The convergence criteria of the algorithm depends on the convergence factor and the relaxation parameter. The best choice of the convergence factor is revealed, in order to maximize the convergence rate or equivalently to minimize the spectral radius of the associated iteration matrix. Some numerical simulations show that the convergence rate when running this algorithm with different ralaxation parameters is faster than those of the GI algorithm 1.1.1.



References

[1] F. Ding, T. Chen, Hierarchical least squares identification methods for multivariable systems, IEEE Transactions on Automatic Control, 42 (3) (1997) 408 -411.

[2] F. Ding, T. Chen, Gradient based iterative algorithms for solving a class of matrix equations, IEEE Transactions on Automatic Control, 50 (8) (2005) 1216-1221.

[3] F. Ding, T. Chen, Hierarchical gradient-based identification of multivariable discrete-time systems, Automatica, 41 (2) (2005) 315-325.

[4] F. Ding, T. Chen, Iterative least squares solutions of coupled Sylvester matrix equations, Systems and Control Letters 54 (2) (2005) 95-107.

[5] F. Ding, P.X. Liu, J. Ding, Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle, Applied Mathematics and Computation 197 (1) (2008) 41-50.

[6] G.E. Dullerud, F. Paganini, A Course in Robust Control Theory: A Convex Approach, Publisher: Springer, New York, (1999)

[7] R.A. Horn and C.R. Johnson, Topics in Matrix Analysis, Cambridge University Press, 1999.

[8] L. Jódar and H. Abou-Kandil, Kronecker products and coupled matrix Riccati differential systems, Linear Algebra Applied 121 (1989), 39-51.

[9] Q. Niu, X. Wang, L-Z. Lu, A relaxed gradient based algorithm for solving Sylvester equation, Asian j. Control 13 (3) (2011) 461-464.

[10] P.J. Olver and C. Shakiban, Applied Linear Algebra, Prentice-Hall, Upper Saddle River, NJ 07458, 2006

[11] X. Wang, L. Dai, D. Liao, A modified gradient based algorithm for solving Sylvester equation, Applied Mathematics and Computation. 218 (2012) 5620-5628.

[12] Y.J. Xie and C.F. Ma, The accelerated gradient based iterative algorithm for solving a class of generalized Sylvester-transpose matrix equation, Applied Mathematics and Computation, 218 (9) (2012) 5620-5628.

[13] X. Zhang and X. Sheng, The Relaxed Gradient Based Iterative Algorithm for the Symmetric (Skew Symmetric) Solution of the Sylvester Equation AX+XB = C, Mathematical Problem in Engineering (2017) 1624969, 8.

Appendices

Appendix A

The research paper

Annual Pure and Applied Mathematics Conference 2018 (APAM 2018) 149
Chulalongkorn University, 30th-31st May and 1st June 2018

Relaxed Gradient Based Iterative Algorithm for Solving the Generalized Sylvester Matrix Equation

Puttachat Songsri [1] and Pattrawut Chansangiam [*2]

[1,2]Department of Mathematics, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang

[1]bbnini0400gmail.com and [2]pattrawut.ch@kmitl.ac.th

Abstract

We propose a new iterative method for solving the generalized Sylvester matrix equation. Our method is based on gradients and hierarchical identification principle. In particular, we obtain relaxed gradient based iterative algorithms for certain interesting special cases of the generalized Sylvester equation. Numerical simulations are also provided to show the applicability and the efficiency of the algorithm.

Mathematics Subject Classification: 65F10, 65F30, 65L20, 15A69

Keywords: Generalized Sylvester matrix equation, Gradient based iterative algorithm, Hierarchical identification principle, Vector operator, Kronecker product.

1 Introduction

Consider the following linear dynamical system:

$$x'(t) = Ax(t) \tag{1}$$

where x is an unknown vector-valued function and A is a given square matrix. To analyse the stability of an equilibrium point of (1), it suffices to find a positive definite matrix P such that $A^TP + PA$ is negative definite [5]. Hence, we will solve the so-called Lyapunov equation

$$AX - XA^T = Q \tag{2}$$

for some negative definite matrix Q. To discuss (2), we shall investigate a more general form, namely, the Sylvester equation:

$$AX + XB = C. \tag{3}$$

*Corresponding author

The generalized Sylvester equation takes the form

$$AXB + CXD = F. \tag{4}$$

Note that when $B = C = I$, (4) reduces to the Sylvester equation (3). (4) also includes the following equation as special cases :

$$AXB = C, \tag{5}$$

$$AXB + X = C. \tag{6}$$

By taking the vector operator, (4) is reduced to $Px = b$ where $P = B^T \otimes A + D^T \otimes C$, $x = \text{vec}\, F$, and $b = \text{vec}\, F$. Here, vec is the vector operator and the symbol $\otimes$ stands for the Kronecker product. Then, (4) has a unique solution if and only if P is nonsingular. This matrix equation has others profound application in linear system theory, see e.g. [5] and references therein.

However, if the dimension of matrices is high, e.g. if A, B, C, D are of dimension 100×100, the dimension of $B^T \otimes A + D^T \otimes C$ is 10000×10000. Such a dimension problem leads to computational difficulty in that excessive computer memory is required for computation and inversion of large matrices. Thus, this approach is only applicable for a small dimension of A.

On the other hand, there are many other methods for solving linear matrix equations. In particular, iterative methods have been received much attention. Gradient based iterative methods for (3), (5), (6) were investigated in [1–4, 6–10]. For the generalized Sylvester equation (4), there are gradient based iterative method [1] as follows:

Algorithm 1.1. The gradient based iterative (GI) algorithm.

Step 1 Input matrices $A, B, C, D, F \in \mathbb{M}_n$ and an error $\varepsilon > 0$. Choose initial matrices $X_1(0)$ and $X_2(0)$. Set $k := 1$.

Step 2 If $\|AX(k)B + CX(k)D - F\|_F / \|F\|_F < \varepsilon$, stop; otherwise, go to Step 3.

Step 3 Update the sequences

$$X(k) = \frac{X_1(k) + X_2(k)}{2},$$
$$X_1(k) = X(k-1) + \tau A^T[F - AX(k-1)B - CX(k-1)D]B^T,$$
$$X_2(k) = X(k-1) + \tau C^T[F - AX(k-1)B - CX(k-1)D]D^T.$$

Step 4 Set $k := k + 1$, return to Step 2.

If we choose the convergence factor τ such that

$$0 < \tau < \frac{2}{\|A\|_2^2\|B\|_2^2 + \|C\|_2^2\|D\|_2^2}, \tag{7}$$

then Algorithm 1.1 converges for any initial values $X_1(0)$ and $X_2(0)$.

In this paper, we shall propose a new iterative method for solving the generalized Sylvester equation (4). Our method is based on gradients and hierarchical identification principle. The convergence analysis shows that the proposed algorithm converges to the unique solution for any initial value matrix. In particular, we obtain relaxed gradient based iterative algorithms for (3) and (5). Numerical simulations are also provided.

2 Proposing the Relaxed Gradient Based Iterative Algorithm for Generalized Sylvester Equation

In this section, we propose an iterative algorithm for solving the equation (4), called the relaxed gradient based iterative algorithm.

In order to make convergence analysis, the following notations are used. For any square matrix X, denote its spectral radius, its maximum eigenvalue and its minimum eigenvalue by $\rho(X)$, $\lambda_{max}(X)$ and $\lambda_{min}(X)$, respectively. The spectral norm and the Frobenius norm of $A \in \mathbb{M}_{m,n}$ are respectively defined by

$$\|A\|_2 = \sqrt{\lambda_{max}(A^TA)} \quad \text{and} \quad \|A\|_F = \sqrt{\text{tr}(A^TA)}.$$

The Frobenius norm for vectors is just the Euclidean norm.

Firstly, we recall the hierarchical identification principle in [1] and apply the principle to solve the generalized Sylvester matrix equation. Define two matrices

$$M := F - CXD \text{ and } N := F - AXB. \tag{8}$$

From (4), we get two subsystems

$$AXB = M \text{ and } CXD = N. \tag{9}$$

We would like to minimize the following quadratic functions:

$$L_1(X) := \|AXB - M\|_F^2, \tag{10}$$

$$L_2(X) := \|CXD - N\|_F^2. \tag{11}$$

Now, for (8) and (9), we will deduce their gradients as follows:

$$\begin{aligned}\frac{\partial}{\partial X}L_1(X) &= \frac{\partial}{\partial X}\mathrm{tr}[(AXB-M)^T(AXB-M)] \\ &= \frac{\partial}{\partial X}\mathrm{tr}(XBB^TX^TA^TA) - \frac{\partial}{\partial X}\mathrm{tr}(X^TA^TMB^T) - \frac{\partial}{\partial X}\mathrm{tr}(BM^TAX) \\ &= (A^TA)^TX(BB^T) + A^TAXBB^T - A^TMB^T - (BM^TA)^T \\ &= 2A^T(AXB-M)B^T. \end{aligned} \tag{12}$$

Similarly, we have

$$\frac{\partial}{\partial X}L_2(X) = 2C^T(CXD-N)D^T. \tag{13}$$

Let $X_1(k)$, $X_2(k)$ and $X(k)$ be the iterative solutions of the system (9) at iteration k. The recursive formulas of $X_1(k)$ and $X_2(k)$ come from the gradient formulas (12) and (13) as follows:

$$\begin{aligned} X_1(k) &= X(k-1) + \tau(1-\omega)A^T[M - AX(k-1)B]B^T \\ &= X(k-1) + \tau(1-\omega)A^T[F - AX(k-1)B - CXD]B^T, \\ X_2(k) &= X(k-1) + \tau\omega C^T[N - CX(k-1)D]D^T \\ &= X(k-1) + \tau\omega C^T[F - AXB - CX(k-1)D]D^T. \end{aligned}$$

Here, the convergence factor τ is introduced to control the convergence of the algorithm. Base on the hierarchical identification principle, the unknown variable X is replaced by its estimates at iteration $k-1$. The iterative solution $X(k)$ is now the weighted arithmetic mean $\omega X_1(k) + (1-\omega)X_2(k)$, here $\omega \in (0,1)$ is called the relaxation parameter. The above discussion leads to the following algorithm for solving the generalized Sylvester equation (4).

Algorithm 2.1. The relaxed gradient based iterative (RGI) algorithm.

Step 1 Input matrices $A, B, C, D, F \in \mathbb{M}_n(\mathbb{R})$, given any small positive number ε and appropriative positive number ω. Choose the initial matrices $X_1(0)$ and $X_2(0)$. Set $k := 1$.

Step 2 If $\delta_k := \|AX(k)B + CX(k)D - F\|_F / \|F\|_F < \varepsilon$, stop; otherwise, go to Step 3.

Step 3 Update the sequences

$$X_1(k) = X(k-1) + (1-\omega)\tau A^T[F - AX(k-1)B - CX(k-1)D]B^T, \quad (14)$$

$$X_2(k) = X(k-1) + \omega\tau C^T[F - AX(k-1)B - CX(k-1)D]D^T, \quad (15)$$

$$X(k) = \omega X_1(k) + (1-\omega)X_2(k). \quad (16)$$

Step 4 Set $k := k+1$, return to Step 2.

3 Convergence Analysis of the Proposed Algorithm

In this section, we shall discuss a criterion for the applicability of the algorithm, and the optimal parameter for the fastest convergence.

The next theorem shows the applicability of Algorithm 2.1.

Theorem 3.1. *Assume that the matrix equation $AXB + CXD = F$ has a unique solution X. If the parameter τ is chosen so that*

$$0 < \tau < \frac{2}{\omega(1-\omega)(\|A\|_2\|B\|_2 + \|C\|_2\|D\|_2)^2} := \tau_0,$$

then the iterative sequence $X(k)$ generated by Algorithm 2.1 converges to X, i.e., $\lim_{k\to\infty} X(k) = X$ for any initial value $X_1(0)$ and $X_2(0)$.

Proof. Define the error matrices

$$\widehat{X}_1(k) := X_1(k) - X,\ \widehat{X}_2(k) := X_2(k) - X,\ \widehat{X}(k) := X(k) - X$$
$$\text{and } \theta_1(k) := A\widehat{X}(k-1)B,\ \theta_2(k) := C\widehat{X}(k-1)D.$$

We shall show that $\lim_{k\to\infty} X(k) = 0$. From Algorithm 2.1, we get

$$\widehat{X}_1(k) = \widehat{X}(k-1) - (1-\omega)\tau A^T(\theta_1(k) + \theta_2(k))B^T,$$
$$\widehat{X}_2(k) = \widehat{X}(k-1) - \omega\tau C^T(\theta_1(k) + \theta_2(k))D^T.$$

Hence,

$$\begin{aligned}\widehat{X}(k) &= X(k) - X \\ &= \omega\widehat{X}_1(k) + (1-\omega)\widehat{X}_2(k) \\ &= \widehat{X}(k-1) - \omega(1-\omega)\tau[A^T(\theta_1(k)+\theta_2(k))B^T + C^T(\theta_1(k)+\theta_2(k))D^T].\end{aligned}$$

Therefore,

$$\begin{aligned}\|\widehat{X}(k)\|_F^2 &= \|\widehat{X}(k-1)\|_F^2 - 2\omega(1-\omega)\tau\|\theta_1(k)+\theta_2(k)\|_F^2 \\ &\quad + \omega^2(1-\omega)^2\tau^2[A^T(\theta_1(k)+\theta_2(k))B^T + C^T(\theta_1(k)+\theta_2(k))D^T]^2 \\ &\leq \|\widehat{X}(k-1)\|_F^2 - 2\omega(1-\omega)\tau\|\theta_1(k)+\theta_2(k)\|_F^2 \\ &\quad + \omega^2(1-\omega)^2\tau^2\,\mathrm{tr}[(A^TB^T + C^TD^T)(BA+DC)(\theta_1(k)^T+\theta_2(k)^T)(\theta_1(k)+\theta_2(k))] \\ &\leq \|\widehat{X}(k-1)\|_F^2 - 2\omega(1-\omega)\tau\|\theta_1(k)+\theta_2(k)\|_F^2 \\ &\quad + \omega^2(1-\omega)^2\tau^2\lambda_{max}[(A^TB^T + C^TD^T)(BA+DC)]\|\theta_1(k)+\theta_2(k)\|_F^2 \\ &\leq \|\widehat{X}(k-1)\|_F^2 - 2\omega(1-\omega)\tau\|\theta_1(k)+\theta_2(k)\|_F^2 \\ &\quad + \omega^2(1-\omega)^2\tau^2\|A^TB^T + C^TD^T\|_2^2\|\theta_1(k)+\theta_2(k)\|_F^2.\end{aligned}$$

By induction, we have

$$\|\widehat{X}(k)\|_F^2 \leq \|\widehat{X}(0)\|_F^2 - \gamma\sum_{i=1}^{k}\|\theta_1(i)+\theta_2(i)\|_F^2,$$

where $\gamma = 2\omega(1-\omega)\tau - \omega^2(1-\omega)^2\tau^2(\|A\|_2\|B\|_2 + \|C\|_2\|D\|_2)^2$. Since $0 < \tau < \tau_0$, we have $\gamma > 0$. Now, we have

$$\sum_{i=1}^{k}\|\theta_1(i)+\theta_2(i)\|_F^2 \leq \frac{1}{\gamma}\left(\|\widehat{X}(0)\|_F^2 - \|\widehat{X}(k)\|_F^2\right) \leq \frac{1}{\gamma}\|\widehat{X}(0)\|_F^2,$$

and thus

$$\sum_{i=1}^{\infty}\|\theta_1(i)+\theta_2(i)\|_F^2 < \infty.$$

It implies that $\theta_1(k)+\theta_2(k) \to 0$ as $k \to \infty$, that is,

$$A\widehat{X}(k-1)B + C\widehat{X}(k-1)D \to 0 \; as \; k \to \infty.$$

Note that the uniqueness of the solution for (4) implies the invertibility of the matrix $B^T \otimes A + D^T \otimes C$. Now, $(B^T \otimes A + D^T \otimes C)\,\mathrm{vec}\,\widehat{X}(k-1) \to 0$ as $k \to \infty$ and, hence, $\widehat{X}(k-1) \to 0$ as $k \to \infty$. Therefore, $\lim_{k\to\infty} \widehat{X}(k) = 0$. □

Remark 3.2. When $\omega = \frac{1}{2}$, the range of parameters τ in Theorem 3.1 is wider than that of given in (7).

Now, we discuss the optimality for the convergence factor τ by using the next lemma.

Lemma 3.3. *(e.g. [9]) For any real number a, b with $b > a$, and $x > 0$. If $0 < a < b$, then*

$$\min_{0<x\leq\frac{2}{b}} \{\max\{|\,1-ax\,|, |\,1-bx\,|\}\} = \frac{b-a}{b+a} < 1.$$

The optimality is reached at $\tau_{opt} = \dfrac{2}{a+b}$.

Theorem 3.4. *Suppose the matrix $\Psi := B^T \otimes A + D^T \otimes C$ is invertible. Then, the optimal factor for which Algorithm 2.1 converges for any initial value is given by*

$$\tau_{opt} = \frac{2}{\omega(1-\omega)(\lambda_{min}(\Psi^T\Psi) + \lambda_{max}(\Psi^T\Psi))}. \tag{17}$$

Proof. Denote the error matrix $\widehat{X}(k) = X(k) - X$. From the proof of Theorem (3.1) , we have

$$\begin{aligned}\widehat{X}(k) &= \omega X_1(k) + (1-\omega)X_2(k),\\ &= \widehat{X}(k-1) - \omega(1-\omega)\tau[A^T(A\widehat{X}(k-1)B + C\widehat{X}(k-1)D]B^T\\ &\quad - \omega(1-\omega)\tau[C^T(A\widehat{X}(k-1) + C\widehat{X}(k-1)D]D^T.\end{aligned}$$

By applying the vector operator, we now obtain

$$\begin{aligned}&\mathrm{vec}\,\widehat{X}(k)\\ &= \mathrm{vec}\,\widehat{X}(k-1) - \tau\omega(1-\omega)[\mathrm{vec}\,(A^TA\widehat{X}(k-1)BB^T) + \mathrm{vec}\,(A^TC\widehat{X}(k-1)DB^T)\\ &\quad + \mathrm{vec}\,(C^TA\widehat{X}(k-1)BD^T) + \mathrm{vec}\,(C^TC\widehat{X}(k-1)DD^T)]\\ &= \{I_{n^2} - \tau\omega(1-\omega)[(B\otimes A^T)(B^T\otimes A) + (B\otimes A^T)(D^T\otimes C)\\ &\quad + (D\otimes C^T)(B^T\otimes A) + (D\otimes C^T)(D^T\otimes C)]\}\,\mathrm{vec}\,\widehat{X}(k-1)\\ &= \{I_{n^2} - \tau\omega(1-\omega)\Psi^T\Psi\}\,\mathrm{vec}\,\widehat{X}(k-1).\end{aligned} \tag{18}$$

Since $x^T\Psi^T\Psi x = (\Psi x)^T(\Psi x) \geqslant 0$ for each $x \in \mathbb{R}^{n^2}$, the matrix $\Psi^T\Psi$ is positive semidefinite. Since Ψ is invertible, we have that $\Psi^T\Psi$ is positive definite. The spectral radius of the iteration matrix is given by

$$\begin{aligned}&\rho(I_{n^2} - \tau\omega(1-\omega)\Psi^T\Psi)\\ &= \max\{|\,1 - \tau\omega(1-\omega)\lambda_{min}(\Psi^T\Psi)\,|, |\,1 - \tau\omega(1-\omega)\lambda_{\max}(\Psi^T\Psi)\,|\}. \end{aligned} \quad (19)$$

Thus, the minimality of this spectral radius subject to the condition $0 < \tau < \tau_0$ in the view of Lemma 3.3 is determined by (17). □

4 Iterative Methods for Special Cases of The Generalized Sylvester Equation

In this section, we discuss iterative algorithms for solving interesting special cases of (4). The following one is for solving (5).

Algorithm 4.1. The relaxed gradient based iterative (RGI) algorithm for solving (5).

Step 1 Input matrices $A, B, C \in \mathbb{M}_n(\mathbb{R})$, given any small positive number ε and appropriative positive number ω. Choose the initial matrix $X(0)$. Set $k := 1$.

Step 2 If $\delta_k := \|AX(k)B - C\|_F / \|C\|_F < \varepsilon$, stop; otherwise, go to Step 3.

Step 3 Update the sequences

$$X(k) = \omega[X(k-1) + (1-\omega)\tau A^T(C - AX(k-1) - X(k-1)B)B^T] \quad (20)$$

Step 4 Set $k := k + 1$, return to Step 2.

Corollary 4.2. *Assume that (5) has a unique solution X. If the parameter τ is chosen so that*

$$0 < \tau < \frac{2}{\omega(1-\omega)(\|A\|_2^2\|B\|_2^2)},$$

then the iterative sequence $X(k)$ generated by Algorithm 4.1 converges to X, for any initial value $X(0)$.

Proof. From Theorem 3.1, put $C = 0$ and $F = C$. □

The next algorithm is proposed for solving the Sylvester equation (3).

Algorithm 4.3. The relaxed gradient based iterative (RGI) algorithm for solving (3).

Step 1 Input matrices $A, B, C \in \mathbb{M}_n(\mathbb{R})$, given any small positive number ε and appropriative positive number ω. Choose the initial matrices $X_1(0)$ and $X_2(0)$. Set $k := 1$.

Step 2 If $\delta_k := \|AX(k) + X(k)B - C\|_F / \|C\|_F < \varepsilon$, stop; otherwise, go to Step 3.

Step 3 Update the sequences

$$X_1(k) = X(k-1) + (1-\omega)\tau A^T[C - AX(k-1) - X(k-1)]B, \tag{21}$$
$$X_2(k) = X(k-1) + \omega\tau[C - AX(k-1) - X(k-1)B]B^T, \tag{22}$$
$$X(k) = \omega X_1(k) + (1-\omega)X_2(k). \tag{23}$$

Step 4 Set $k := k + 1$, return to Step 2.

Corollary 4.4. *Assume that* (3) *has a unique solution* X. *If the parameter* τ *is chosen so that*

$$0 < \tau < \frac{2}{\omega(1-\omega)(\|A\|_2 + \|B\|_2)^2},$$

then the iterative sequence $X(k)$ *generated by Algorithm 4.3 converges to* X, *for any initial value* $X_1(0)$ *and* $X_2(0)$.

Proof. It follows from Theorem 3.1 by setting $B = C = I$, $D = B$ and $F = C$. □

The next algorithm is proposed for solving the discrete-time Sylvester equation (6).

Algorithm 4.5. The relaxed gradient based iterative (RGI) algorithm for solving (6).

Step 1 Input matrices $A, B, C \in \mathbb{M}_n(\mathbb{R})$, given any small positive number ε and appropriative positive number ω. Choose the initial matrices $X_1(0)$ and $X_2(0)$. Set $k := 1$.

Step 2 If $\delta_k := \|AX(k)B + X(k) - C\|_F / \|C\|_F < \varepsilon$, stop; otherwise, go to Step 3.

Step 3 Update the sequences

$$X_1(k) = X(k-1) + (1-\omega)\tau A^T[C - AX(k-1)B - X(k-1)]B^T, \tag{24}$$
$$X_2(k) = X(k-1) + \omega\tau[C - AX(k-1)B - X(k-1)]. \tag{25}$$
$$X(k) = \omega X_1(k) + (1-\omega)X_2(k). \tag{26}$$

Step 4 Set $k := k+1$, return to Step 2.

Corollary 4.6. *Assume that (6) has a unique solution X. If the parameter τ is chosen so that*

$$0 < \tau < \frac{2}{\omega(1-\omega)(\|A\|_2\|B\|_2+1)^2},$$

then the iterative sequence $X(k)$ generated by Algorithm 4.5 converges to X, for any initial value $X_1(0)$ and $X_2(0)$.

Proof. It follows from Theorem 3.1 by setting $C = D = I$ and $F = C$. □

5 Numerical Simulation

In this section, we report numerical results to illustrate the effectiveness of Algorithm 2.1. All iterations have been carried out by MATLAB 724504-R2013a, Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz 3.19 GHz, RAM 8.00 GB. PC environment.

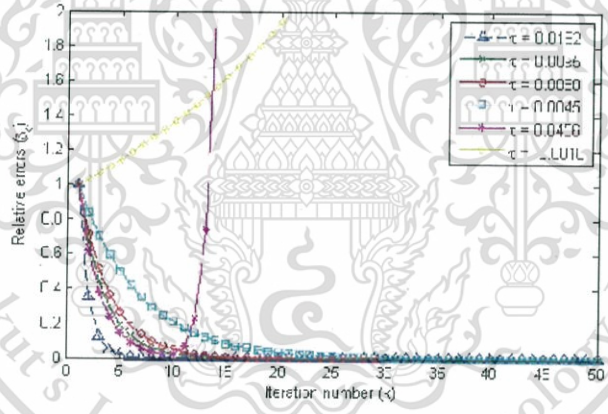


Figure 1: Relative errors for Example 5.1

Example 5.1. We consider the equation (4) with

$$A = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}, B = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \text{ and } F = \begin{bmatrix} 65 & 26 \\ 26 & 65 \end{bmatrix}.$$

k	x_{11}	x_{12}	x_{21}	x_{22}	$\|X(k)-X\|_F/\|X\|_F$
2	3.2296	1.2918	1.2918	3.2296	0.3541
4	4.7780	1.9112	1.9112	4.7780	0.0444
6	4.9722	1.9889	1.9889	4.9722	0.0056
8	4.9965	1.9986	1.9986	4.9965	$7.0000e-04$
10	4.9996	1.9998	1.9998	4.9996	$8.3045e-05$
Solution	5	2	2	5	

Table 1. Iterative solution ($\tau = 0.0182$)

Then, the matrix $\Psi^T\Psi$ is given by

$$\Psi^T\Psi = \begin{bmatrix} 14 & 3 & -3 & -1 \\ 3 & 14 & -1 & -3 \\ -3 & -1 & 14 & 3 \\ -1 & -3 & 3 & 614 \end{bmatrix}.$$

Choose $\omega = 0.7$ and initial matrices

$$X_1(0) = 10^{-6}\begin{bmatrix} 5 & 1 \\ 4 & 0 \end{bmatrix} \text{ and } X_2(0) = 10^{-9}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Then, the optimal convergence factor is given by

$$\tau_{opt} = \frac{2}{\omega(1-\omega)(\lambda_{min}(\Psi^T\Psi) + \lambda_{max}(\Psi^T\Psi))} \approx 0.0182.$$

Running Algorithm 2.1, we see in Table 1 that the approximated solutions converge to the exact solution

$$X^* = \begin{bmatrix} 5 & 2 \\ 2 & 5 \end{bmatrix},$$

although the initial matrices are very closed to the zero matrix. Clearly, as k increases, the term $\|X(k)-X\|_F/\|X\|_F$ becomes smaller and goes to zero. Fig 1 shows the relative errors for the algorithm with different convergence factors τ. We see that as k is large enough, the relative error for $\tau = 0.0182$ goes faster to 0 than those for another convergence factors.

Example 5.2. We consider the equation $AXB + CXD = F$ with

$$A = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 2 & -2 & -2 \\ 8 & 0 & 2 & -2 \\ -2 & 2 & 2 & -4 \\ 6 & -2 & 4 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} -2 & 6 & 2 & 4 \\ 0 & 4 & 6 & 2 \\ 2 & -2 & 4 & -4 \\ 8 & 2 & 4 & -2 \end{bmatrix},$$

$$D = \begin{bmatrix} 4 & 8 & 2 & 2 \\ -2 & 10 & 0 & 2 \\ 2 & 2 & -2 & -2 \\ 4 & 4 & 6 & 2 \end{bmatrix} \quad \text{and} \quad F = \begin{bmatrix} 456 & 1220 & -20 & 20 \\ 576 & 996 & 184 & 32 \\ 396 & -336 & 216 & -172 \\ 804 & 304 & 376 & -64 \end{bmatrix}.$$

We compare the efficiency Algorithm 2.1 with GI Algorithm. In both algorithms, we use the same convergence factor $\tau = 2.1323 \times 10^{-4}$ and initial matrices

$$X_1(0) = 10^{-6} \begin{bmatrix} -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix} \text{ and } X_2(0) = 10^{-6} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

k	GI	RGI, $\omega = 0.55$	RGI, $\omega = 0.6$	RGI, $\omega = 0.65$	RGI, $\omega = 0.7$
20	0.7972	0.5441	0.1706	0.0550	0.0539
40	0.7633	0.3481	0.0451	0.0354	0.0372
60	0.7311	0.2230	0.0258	0.0264	0.0281
80	0.7005	0.1430	0.0195	0.0205	0.0221
100	0.6712	0.0919	0.0156	0.0165	0.0179

Table 2. Relative errors for Example 5.2

Fig 2 and Table 2 show the relative errors when running Algorithm 2.1 with different ralaxation parameters, comparing with GI Algorithm. We notice that the RGI Algorithm with $\omega = 0.55, 0.6, 0.65$ and 0.7 converge faster than the GI Algorithm.

6 Conclusions

We propose a relaxed gradient based iterative algorithm for solving the generalized Sylvester equation (4). This algorithm guarantees that the approximated solutions converge to the exact solution for any given initial value. The convergence criteria of the algorithm depends on the convergence factor and the relaxation parameter. The best choice of the conver-

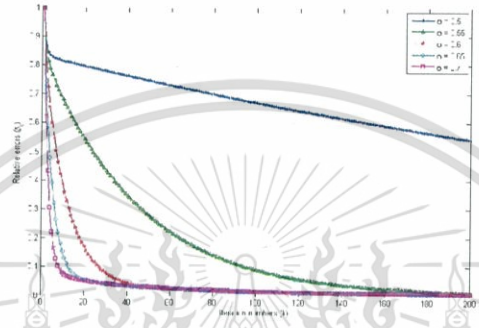


Figure 2: Relative errors for Example 5.2

gence factor is revealed, in order to maximize the convergence rate or equivalently to minimize the spectral radius of the associated iteration matrix. Some numerical simulations show that the convergence rate when running this algorithm with different ralaxation parameters is faster than those of the GI algorithm 1.1.

References

[1] 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 **197** (2008), no. 1, 41–50.

[2] F. Ding and T. Chen, *Gradient based iterative algorithms for solving a class of matrix equations*, IEEE Transactions on Automatic Control **50** (2005), no. 8, 1216–1221.

[3] ———, *Hierarchical gradient-based identification of multivariable discrete-time systems*, Automatica **41** (2005), no. 2, 315–325.

[4] ———, *Iterative least squares solutions of coupled Sylvester matrix equations*, Systems and Control Letters **54** (2005), no. 2, 95–107.

[5] G. E. Dullerud and F. Paganini, *A course in robust control theory: A convex approach*, Springer, New York, 1999.

[6] F. Ding and T. Chen, *Hierarchical least squares identification methods for multivariable systems*, IEEE Trans. Auton. Control **42** (1997), no. 3, 408–411.

[7] Q. Niu, X. Wang, and L-Z. Lu, *A relaxed gradient based algorithm for solving Sylvester equation*, Asian j. Control **13** (2011), no. 3, 461–464.

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

[9] Y. J. Xie and C.F. Ma, *The accelerated gradient based iterative algorithm for solving a class of generalized Sylvester-transpose matrix equation*, Applied Mathematics and Computation **218** (2012), no. 9, 5620–5628.

[10] X. Zhang and X. Sheng, *The relaxed gradient based iterative algorithm for the symmetric (skew symmetric) solution of the Sylvester equation AX+XB = C*, Mathematical Problem in Engineering **2017** (2017), 1624969, 8.

Author Biography

Name	Miss Puttachat Songsri
Date of Birth	30 March 1994
Address	4 Moo 2, Lao, Thung Khao Luang, Roi Et, 45170
Education	2012-2015 Bachelor of Science in Applied Mathematics. GPA 3.39 (Second-class honors) King Mongkut's Institute of Technology Ladkrabang
Education	2016-2017 Master of Science in Applied Mathematics. GPA 3.54 King Mongkut's Institute of Technology Ladkrabang
Scholarship	2016-2017 Faculty of science, King Mongkut's Institute of Technology Ladkrabang Graduate Scholarship

Academic Publication(s)

1. Songsri, P. and Chansangiam, P. 2018,"Relaxed gradient based iterative algorithm for solving the generalized Sylvester matrix equation", Proceedings of Annual Pure and Applied Mathematics Conference, Chulalongkorn University, Bangkok, Thailand, 30-31 May and 1 June 2018. pp.149-162.