CONVERGENCE ANALYSIS OF GRADIENT BASED ITERATIVE ALGORITHMS FOR SYLVESTER-TYPE MATRIX EQUATIONS

NUNTHAKARN BOONRUANGKAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE DEGREE OF DOCTOR OF SCIENCE (APPLIED MATHEMATICS)
DEPARTMENT OF MATHEMATICS SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2021

KMITL-2021-SC-D-001-016

Thesis Title	Convergence analysis of gradient based iterative algorithms for Sylvester-type matrix equations
Student Name	Miss Nunthakarn Boonruangkan
Student ID	60605009
Degree	Doctor of Philosophy (Applied Mathematics)
Department	Mathematics
Year	2021
Thesis Advisor	Assoc.Prof.Dr. Pattrawut Chansangiam

Abstract

In this research, we introduce gradient based iterative algorithms for solving a generalized Sylvester matrix equation and a generalized Sylvester-transpose matrix equation. The iterative algorithms are derived from the gradients of the squared norm-errors of the associated subsystems for the equations. The convergence analysis reveals that the sequence of approximated solutions converges to the exact solution for any initial values if and only if the convergence factor is chosen properly in terms of the spectral radius of the associated iteration matrix. We also discuss the convergence rate and error estimations. Moreover, we determine the fastest convergence factor so that the associated iteration matrix has the smallest spectral radius. Finally, we provide numerical examples to illustrate the capability and the efficiency of the proposed algorithms.

Keywords : gradient-based iterative algorithm, linear iterative process, matrix norm, generalized Sylvester matrix equation, generalized Sylvester-transpose matrix equation

Acknowledgements

I feel very grateful to my advisor, Assoc.Prof.Dr. Pattrawut Chansangiam, who gives me a lot of knowledge, understanding, encouragement, constant guidance and support, without which this my research would not have been possible. Special thank to Assoc.Prof.Dr. Wicharn Lewkeeratiyutkul for his precious advice and suggesting this problem. I would like to thank Asst.Prof.Dr. Nopparat Pochai, Asst.Prof.Dr. Kanchana Kumnungkit and Assoc.Prof.Dr. Atid Kangtunyakarn for their helpful advices and reading my reports and commenting on countless revisions of this manuscript.

I also thank my friends i.e. Mr. Arnon Ploymukda and Mr. Jirapat Limthanakul whose share their support, knowledge and advice.

I would like to express my gratitude to my family for their care, support and encouragement during my study. Include thank for providing financial support.

Finally this research was support by Science Achievement Scholarship of Thailand(SAST).

Miss. Nunthakarn Boonruangkan

Table of Contents

Page

Abstract in English

Abstract in English i
Acknowledgements ii
Table of Contents iii
List of Tables iv
List of Figures v
Chapter 1. Introduction 1
1.1 Research Motivation 1
1.2 Objectives of the study 4
1.3 Scopes of the study 4
1.4 Benefits of the Study 5
1.5 Research methodology 5
Chapter 2. Preliminaries 6
2.1 Elements in matrix and functional analysis 6
2.2 Banach contraction principle 8
2.3 Linear Iterative Systems 9
2.4 Elements in finite difference method 9
Chapter 3. Gradient-based iterative algorithm with optimal convergence factor for a generalized Sylvester matrix equation 11
3.1 A direct method for a generalized Sylvester matrix equation 11
3.2 Introducing a Gradient-based iterative algorithm for the matrix equation a generalized Sylvester matrix equation 11
3.3 Convergence Analysis 13
3.4 The GIO algorithm for the Sylvester equation 15
3.5 Numerical simulations with discussion 16
3.6 An Application to discretization of the convection-diffusion equation 20
Chapter 4. Gradient-based iterative algorithm with optimal convergence factor for a generalized Sylvester-transpose matrix equation 25
4.1 A direct method for the generalized Sylvester-transpose matrix equation 25
4.2 Introducing a gradient-based iterative algorithm for a generalized Sylvester-transpose matrix equation 25
4.3 Convergence analysis 27
4.3.1 Convergence criteria 28

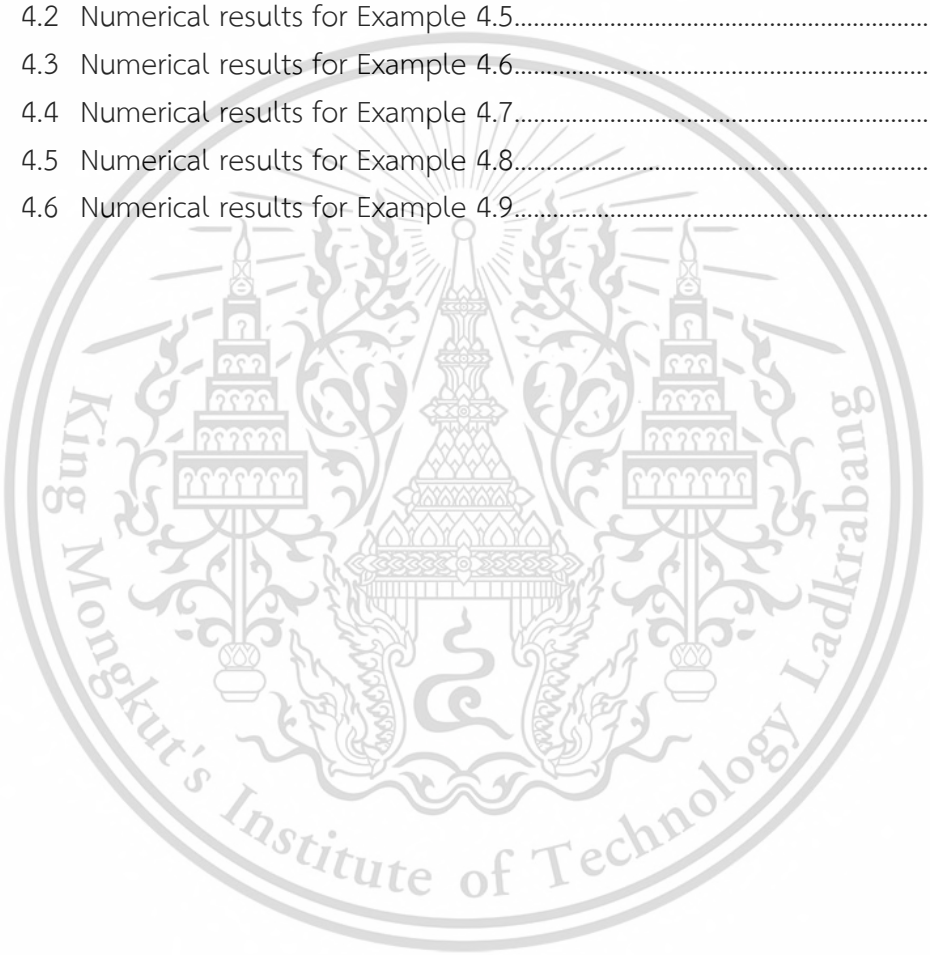
4.3.2 Performance of the algorithm 29
4.3.3 Optimal convergence factor 30
4.4 The GIO algorithm for the Sylvester-transpose equation 31
4.5 Numerical simulations with discussion 32
Chapter 5. Conclusions and Suggestions 41
5.1 Conclusions 41
5.2 Suggestions 42

List of Tables

Table **Page**

3.1 Iteration numbers and computational times for Example 3.6 18
3.2 Iteration numbers and computational times for Example 3.7 19
3.3 Iteration numbers and computational times for Example 3.8 19
3.4 Iteration numbers, computational times and errors for Example 3.9..... 23

4.1 Numerical results for Example 4.4..... 34
4.2 Numerical results for Example 4.5..... 35
4.3 Numerical results for Example 4.6..... 37
4.4 Numerical results for Example 4.7..... 37
4.5 Numerical results for Example 4.8..... 39
4.6 Numerical results for Example 4.9..... 40



List of Figures

Figure **Page**

3.1 Relative error for Example 3.6 18
3.2 Relative errors for Example 3.7 19
3.3 Relative errors for Example 3.8 20
3.4 Relative errors for Example 3.9 22
3.5 Relative errors for Example 3.10 23
3.6 The exact (left) and the iterative (right) solutions for Example 3.10 24

4.1 Relative error for Example 4.4 34
4.2 Relative errors for Example 4.5 35
4.3 Relative error for Example 4.6 36
4.4 Relative error for Example 4.7 38
4.5 Relative error for Example 4.8 39
4.6 Relative error for Example 4.9 40

Chapter 1

Introduction

1.1 Research Motivation

Differential equations play a prominent role in many disciplines including engineering, physics, economics, and biology. Mainly the study of differential equations consists of the study of the following linear system

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

where $x(t)$ is an unknown vector-valued function and A is a given square matrix. To analyse the stability of an equilibrium point of the system (1.1), it suffices to find a positive definite matrix L such that $A^TL + LA$ is negative definite; see e.g. [1]. So, we need to solve the so-called Lyapunov equation

$$AX - X^TA = R \tag{1.2}$$

for some negative definite matrix R. To discuss the equation (1.2), we investigate more general form, namely, the generalized Sylvester matrix equation and the generalized Sylvester-transpose matrix equation. Those are respectively show as:

$$\sum_{i=1}^{p} A_iXB_i = F, \tag{1.3}$$

$$\sum_{i=1}^{p} A_iXB_i + \sum_{j=1}^{q} C_jX^TD_j = F, \tag{1.4}$$

where A_i, B_i, C_j, D_j and F are given matrices of conforming dimensions and X is an unknown matrix to be determined. The equation (1.3) and (1.4) include important practical problems that are written as the matrix equations

$$AX + XB = C, \tag{1.5}$$

$$AX + X^TB = C, \tag{1.6}$$

$$AXB + CXD = E, \tag{1.7}$$

$$AXB + X = C, \tag{1.8}$$

known as Sylvester, Sylvester-transpose, generalized Sylvester, Kalman-Yakubovich matrix equations, respectively. The equations (1.2)-(1.8) have important applications in stability analysis, optimal control, observer design, output regulation problem, and so on; see e.g. [2, 3, 4, 5, 6, 7]. In direct method, a vectorization and the Kronecker product are used to find the exact solution. Here, recall that the vector operator vec[.] turns each matrix into a column vector by stacking its columns consecutively.

The Kronecker product of two matrices $A = [a_{ij}]$ and B is defined to be the block matrix $A \otimes B = [a_{ij}B]$. In fact, the equation (1.3) can be reduced to the linear system

$$Px \;=\; b,$$

where $P = \sum_{i=1}^{p}(B_i^T \otimes A_i)$, $b = \text{vec}[F]$ and $x = \text{vec}[X]$. Similarly, for the commutation matrix K, the equation (1.4) can be reduced to

$$Qx \;=\; b,$$

where $Q = \sum_{i=1}^{p}(B_i^T \otimes A_i) + \sum_{j=1}^{q}(D_j^T \otimes C_j)K$, $b = \text{vec}[F]$ and $x = \text{vec}[X]$. Thus, (1.3) and (1.4) have a unique solution if and only if P and Q are nonsingular. The exact solution $x = P^{-1}b$ and $x = Q^{-1}b$ are, in fact, computationally difficult due to the large size of the Kronecker multiplication. This inspires us to investigate certain iterative algorithms to generate a sequence of approximate solutions which are arbitrarily close to the exact solution. Efficient iterative algorithms produce a satisfaction approximated solution in a small iteration number. Many researchers have been developed such iterative algorithms for solving a class of matrix equations (1.2), (1.5)-(1.8) using many ideas, e.g. matrix sign function [8, 9], recursive blocked algorithms [10, 11] and Hermitian and skew-Hermitian splitting algorithms [12, 13, 14, 15]. In 2005, gradient-based iterative algorithms were firstly introduced by F. Ding and T. Chen for solving (1.5), (1.7) and (1.8); see e.g. [16, 17, 18]. After that, there are many iterative algorithms for solving (1.2)-(1.8) based on gradients and hierarchical identification principle, e.g. relaxed gradient based iterative method (RGI) [19], Jacobi-gradient based iterative method [20, 21], modified gradient based iterative method (MGI) [22], accelerated Jacobi-gradient based iterative method (AJGI)[23] and accelerated gradient based iterative algorithm (AGBI) [22]. See more information in [24, 25, 26, 28]. Convergence analysis of such algorithms are often relied on the Frobenious norm $\|\cdot\|_F$ and the spectral norm $\|\cdot\|_2$, defined for each matrix A by

$$\|A\|_F \;=\; (\text{tr}\, A^T A)^{\frac{1}{2}} \quad \text{and} \quad \|A\|_2 \;=\; (\lambda_{\max}(A^T A))^{\frac{1}{2}}.$$

A gradient-based iterative algorithm for solving (1.3) and (1.4) were introduced as follows:

Theorem 1.1 ([16])**.** Assume that the Sylvester matrix equation (1.3) has a unique solution X. For each $i = 1, 2, \ldots, p$, construct,

$$X_i(k) \;=\; X(k-1) + \tau A_i^T[F - \sum_{j=1}^{p} A_i X(k-1) B_i]B_i^T,$$

$$X(k) \;=\; \frac{1}{p}\left(\sum_{i=1}^{p} X_i(k)\right).$$

If we choose $\tau = [\|A_i\|_2^2\|B_i\|_2^2]^{-1}$, then the sequence $\{X(k)\}_{k=0}^{\infty}$ converges to the exact solution X for any given initial matrices $X_1(0), X_2(0), ..., X_p(0)$.

Theorem 1.2 ([28]). Assume that the matrix equation (1.4) has a unique solution X. For each $i = 1, 2, \ldots, p$ and $j = p+1, \ldots, q$, construct,

$$\begin{aligned} X_i(k) &= X(k-1) + \tau A_i^T [F - \sum_{s=1}^{p} A_s X(k-1) B_s - \sum_{t=1}^{q} C_t X^T(k-1) D_t] B_i^T, \\ X_j(k) &= X(k-1) + \tau D_j [F - \sum_{s=1}^{p} A_s X(k-1) B_s - \sum_{t=1}^{q} C_t X^T(k-1) D_t]^T C_j, \\ X(k) &= \frac{1}{p+q} \left(\sum_{i=1}^{p} X_i(k) + \sum_{j=p+1}^{q} X_j(k) \right). \end{aligned}$$

If $0 < \tau < [\sum_{i=1}^{p} \lambda_{\max}(A_i A_i^T) \lambda_{\max}(B_i B_i^T) + \sum_{j=p+1}^{q} \lambda_{\max}(C_j C_j^T) \lambda_{\max}(D_j D_j^T)]$, then sequence $\{X(k)\}_{k=0}^{\infty}$ converges to the exact solution X for any given initial matrices $X_1(0), \ldots, X_{p+q}(0)$.

A least-squares based iterative algorithm for solving (1.3) and (1.4) were introduced as follows:

Theorem 1.3. ([16]). Assume that the Sylvester matrix equation (1.3) has a unique solution X. For each $i = 1, 2, \ldots, p$, construct,

$$\begin{aligned} X_i(k) &= X(k-1) + \tau \sum_{i=1}^{p} (A_i^T A_i)^{-1} A_i^T [F - \sum_{j=1}^{p} A_j X(k-1) B_j] B_i^T (B_i B_i^T)^{-1}, \\ X(k) &= \frac{1}{p} \left(\sum_{i=1}^{p} X_i(k) \right). \end{aligned}$$

If we choose $0 < \tau < 2p$, then the sequence $\{X(k)\}_{k=0}^{\infty}$ converges to the exact solution X for any given initial matrices $X_1(0), X_2(0), ..., X_p(0)$.

Theorem 1.4 ([28]). Assume that the matrix equation (1.4) has a unique solution X. For each $i = 1, 2, \ldots, p$ and $j = p+1, \ldots, q$, construct,

$$\begin{aligned} R(k) &= E - \sum_{s=1}^{p} A_s X(k-1) B_s - \sum_{t=1}^{q} C_t X^T(k-1) D_t, \\ X_i(k) &= X(k-1) + \mu (A_i^T A_i)^{-1} A_i^T R(k) B_i^T (B_i B_i^T)^{-1}, \\ X_j(k) &= X(k-1) + \mu (D_j D_j^T)^{-1} D_j R(k) C_j (C_j^T C_j)^{-1}, \\ X(k) &= \frac{1}{p+q} \left(\sum_{i=1}^{p} X_i(k) + \sum_{j=p+1}^{q} X_j(k) \right). \end{aligned}$$

If we choose $0 < \tau < 2(p+q)$, then the sequence $\{X(k)\}_{k=0}^{\infty}$ converges to the exact solution X for any given initial matrices $X_1(0), X_2(0), ..., X_{p+q}(0)$.

In this research, we propose the gradient based iterative algorithms with an optimal convergence factor for solving the generalized Sylvester matrix equation (1.3) and Sylvester-transpose matrix equation (1.4). These algorithms are derived from gradients and least-squares optimization principle. Convergence analysis reveals that the sequence of approximated solutions converges to the exact solution for any initial

value if and only if the convergence factor is chosen properly. Then we discuss the convergence rate and error estimates for the proposed algorithms. Moreover, the convergence factor will be determined so that the asymptotic convergence rate is fastest, or equivalently, the spectral radius of associated iteration matrix is minimized. We provide numerical experiments to illustrate the efficiency of the proposed algorithms. Furthermore, we apply our algorithm to the convection-diffusion and the diffusion equations. Finally, we conclude the overall work.

1.2 Objectives of the study

1) Introduce Gradient based iterative algorithm for solving the generalized Sylvester and Sylvester-transpose matrix equations.
2) Make convergence analysis for the proposed algorithms as follows:
   (a) convergence criteria
   (b) convergence rate
   (c) error estimates
   (d) optimal convergence factor.
3) Provide numerical simulations to illustrate the efficiency for the proposed algorithm based on
   (a) iteration number
   (b) computational time
   (c) relative errors.
4) Apply the proposed algorithms to discretization of convection-diffusion and diffusion equations.

1.3 Scopes of the study

We consider the generalized Sylvester matrix equation

$$\sum_{i=1}^{p} A_i X B_i = F$$

and the generalized Sylvester-transpose matrix equation

$$\sum_{i=1}^{p} A_i X B_i + \sum_{j=1}^{q} C_j X^T D_j = F.$$

All matrices considered are real.

1.4 Benefits of the Study

1) Attain the gradient-based iterative algorithm for solving a generalized Sylvester and a generalized Sylvester-transpose matrix equations.

2) Obtain the convergence criteria, convergence rate, error estimates and the optimal convergence factor for the proposed algorithm.

1.5 Research methodology

1) Study advanced topics in matrix theory and functional analysis.

2) Study background in applied linear algebra.

3) Study topic of iterative algorithm for solving linear system from research paper.

4) Collect and study research papers and textbooks concerning iterative algorithm based on gradient for solving Sylvester and Sylvester-transpose matrix equations.

5) Propose a gradient based iterative algorithm together optimal convergence factor.

6) Analysis a convergence of algorithm.

7) Provide numerical simulations.

8) Conclude the results, make suggestions for further works and write the thesis.

Activity	Time frame										
	2017	2018			2019			2020			2021
	9-12	1-4	5-8	9-12	1-4	5-8	9-12	1-4	5-8	9-12	1-6
Step 1	⟷										
Step 2		⟷									
Step 3			⟷								
Step 4				⟷							
Step 5					⟷	⟷					
Step 6							⟷	⟷			
Step 7								⟷	⟷		
Step 8										⟷	⟷

Chapter 2

Preliminaries

In this chapter, we provide sufficiently some standard results that will be referred in the next chapter. First, we study some theorem of matrix and functional analysis. Then we discuss a Banach contraction principle. Next we consider the linear iterative systems relative to many important practical matrix equations. Finally we discuss a discretization of diffusion and convection-diffusion equations. We denote the set of m-by-n real matrices by $M_{m,n}(\mathbb{R})$ and M_n for square matrices instead of $M_{n,n}$.

2.1 Elements in matrix and functional analysis

Definition 2.1. Let $A = [a_{ij}] \in M_{m,n}(\mathbb{R})$ and $B \in M_{p,q}(\mathbb{R})$, then the Kronecker product $A \otimes B$ is the $mp \times nq$ block matrix,

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}. \tag{2.1}$$

Definition 2.2. For an $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \in M_{m,n}(\mathbb{R})$, vec$[X]$ is an mn-dimensional vector formed by the columns of X,

$$\text{vec}[X] = \begin{bmatrix} x_1^T & x_2^T & \cdots & x_n^T \end{bmatrix}^T.$$

Theorem 2.3. (see e.g. [27]) Let $A \in M_{m,n}(\mathbb{R}), X \in M_{n,p}(\mathbb{R})$ and $B \in M_{p,s}(\mathbb{R})$,

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

Definition 2.4. Let $A \in M_n(\mathbb{R})$. The trace of A, denoted tr(A), is defined to be the sum of elements on the main diagonal, i.e.,

$$\text{tr}(A) = \sum_{i=1}^{n} a_{ii}.$$

Lemma 2.5. (see e.g. [27]) Let A, B, C, D be compatibly constant matrices, X be a variable matrix and $c \in \mathbb{R}$. The properties and derivatives of trace of matrix are as follows:

1. $\frac{d}{dX}\,\text{tr}(AX) = \frac{d}{dX}\,\text{tr}(X^T A^T) = A^T$,
2. $\frac{d}{dX}\,\text{tr}(XAX^T B) = BXA + B^T XA^T$.

The commutation matrix K_{mn} is defined by

$$K_{mn} = \begin{bmatrix} I_m \otimes e_1^{n^T} I_m \otimes e_2^{n^T} \cdots I_m \otimes e_n^{n^T} \end{bmatrix} \in \mathbb{R}^{mn \times mn}$$

where e_i^n is the ith column of the $n \times n$ identity matrix I_n. Essential properties of the commutation matrices are given in the following lemma:

Lemma 2.6. (see e.g. [27, Ch. 4]) For any $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, we have

$$\mathsf{vec}(A^T) = K_{mn}\,\mathsf{vec}(A), \tag{2.2}$$

$$K_{mn}^{-1} = K_{mn}^T = K_{nm}, \tag{2.3}$$

$$(B \otimes A) = K_{pm}(A \otimes B)K_{nq}. \tag{2.4}$$

Definition 2.7. Let V be a vector space over the field $\mathbb{F}(\mathbb{F} = \mathbb{R}$ or $\mathbb{C})$. A function $\|\cdot\| : V \to \mathbb{R}$ is a norm if, for all $x, y \in V$ and all $c \in \mathbb{F}$,

1. $\|x\| \geqslant 0$,
2. $\|x\| = 0$ if and only if $x = 0$,
3. $\|cx\| = |c|\|x\|$,
4. $\|x + y\| \leqslant \|x\| + \|y\|$.

Definition 2.8. Let X and Y be vector spaces. A linear operator or a linear map T from X into Y is a function $T : X \to Y$ such that

1. $T(x + y) = T(x) + T(y)$ for any $x, y \in X$, and
2. $T(cx) = cT(x)$ for each $x \in X$ and $c \in \mathbb{F}$.

Definition 2.9. The Frobenius norm is a norm of an $m \times n$ matrix A defined as the square root of the matrix trace of $A^T A$:

$$\|A\|_F = \sqrt{\mathsf{tr}(A^T A)}.$$

Definition 2.10. The Spectral norm is a norm of an $m \times n$ matrix A defined as the square root of the largest eigenvalue of $A^T A$:

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

Lemma 2.11. (see e.g. [27]) For any compatible matrices A and B, we have

1. $\|A^T A\|_2 = \|A\|_2^2$,
2. $\|A^T\|_2 = \|A\|_2$,
3. $\|AB\|_F \leq \|A\|_2\|B\|_F$.

Definition 2.12. Let $\lambda_1, \ldots, \lambda_n$ be the (real or complex) eigenvalues of a matrix $A \in M_n(\mathbb{R})$. Then its spectral radius $\rho(A)$ is defined as:

$$\rho(A) = \mathsf{max}\{|\lambda_1|, \ldots, |\lambda_n|\}.$$

Definition 2.13. Let $A \in M_n(\mathbb{R})$ be symmetric. Then A is called

- **Negative definite** if $x^T Ax < 0$ for all $x \in \mathbb{R}^n - \{0\}$,

- **Positive definite** if $x^T Ax > 0$ for all $x \in \mathbb{R}^n - \{0\}$,
- **Positive semidefinite** if $x^T Ax \geqslant 0$ for all $x \in \mathbb{R}^n$.

Lemma 2.14. (see e.g. [27]) Let A be a symmetric matrix and let us denote $\lambda_{\min}$ ($\lambda_{\max}$, resp.) its smallest (largest, resp.) eigenvalue, then

$$\lambda_{\min}(x^T x) \leqslant x^T Ax \leqslant \lambda_{\max}(x^T x), \qquad x \in \mathbb{R}^n.$$

Definition 2.15. The *condition number* of an $m \times n$ matrix A is the ratio between its largest and smallest eigenvalues of $A^T A$:

$$\kappa(A) = \left(\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)}\right)^{1/2}.$$

2.2 Banach contraction principle

Definition 2.16. A **metric space** (X, d) is a set X and a function $d : X \times X \to \mathbb{R}$ satisfying for all $x, y, z \in X$,

1. $d(x, y) \geq 0$,
2. $d(x, y) = 0$ if and only if $x = y$,
3. $d(x, y) = d(y, x)$,
4. $d(x, z) \leq d(x, y) + d(y, z)$.

Definition 2.17. Suppose (X, d) is a metric space. A sequence $\{x_n\}_{n=1}^{\infty} \subset X$ is called a Cauchy sequence if for every positive real number $r > 0$ there is a positive integer N such that for all positive integers $m, n > N$,

$$d(x_m, x_n) < r.$$

Definition 2.18. A metric space (X, d) is complete if and only if every Cauchy sequence in X is convergence.

Definition 2.19. Let (X, d) be a non-empty complete metric space. Then a map $T : X \to X$ is called contraction mapping on X if there exist $z \in (0, 1]$ such that

$$d(T(x), T(y)) \leq z d(x, y),$$

for all x, y in X.

Theorem 2.20. (see e.g. [30]) Let (X, d) be a non-empty complete metric space with a contraction mapping $T : X \to X$. Then

(i) The map T admits a unique fixed-point x^* in X i.e. $T(x^*) = x^*$.

(ii) The fixed-point x^* can be found as follows: start with an arbitrary element x_0 in X and define a sequence $\{x_n\}$ by $x_n = T(x_{n-1})$ for $n \geq 1$. Then $x_n \to x^*$.

(iii) The following inequalities hold and describe the speed of convergence:

$$d(x^*, x_n) \leq \frac{z^n}{1-z} d(x_1, x_0) \tag{2.5}$$

$$d(x^*, x_{n+1}) \leq \frac{z}{1-z} d(x_{n+1}, x_n) \tag{2.6}$$

$$d(x^*, x_{n+1}) \leq z d(x^*, x_n). \tag{2.7}$$

Theorem 2.21. (see e.g. [31]) For a given matrix $A \in M_n(\mathbb{R})$, the following are equivalent:

1. A is a contraction relative to a norm in $\mathbb{C}^n$;
2. $\|A\| < 1$ for some induced matrix norm $\|\cdot\|$;
3. $\|A\| < 1$ for some matrix norm $\|\cdot\|$;
4. $\rho(A) < 1$.

2.3 Linear Iterative Systems

If an equation can be put into the form $f(x) = x$, and a solution x is an attractive fixed point of the function f, then one may begin with a point x_1 in the basin of attraction of x, and let $x_{n+1} = f(x_n)$ for $n \geq 1$, and the sequence $\{x_n\}_{n=1}^{\infty}$ will converge to the solution x. Here x_n is the n-th approximation or iteration of x and x_{n+1} is the next or $n+1$ iteration of x. We consider the linear system

$$Ax = b \tag{2.8}$$

where $A \in M_n(\mathbb{R})$ is an invertible matrix, $b \in \mathbb{R}^n$ is a known constant vector, and $x \in \mathbb{R}^n$ is an unknown vector to be solved. Indeed the equation has the exact solution $x^* = A^{-1}b$.

We shall be attempting to solve (2.8) by replacing it with a form of iterative system

$$x(k+1) = Tx(k) + c, \quad x(0) = x_0$$

in which T is an $n \times n$ matrix, and called an *iteration matrix*, and c is a vector.

2.4 Elements in finite difference method

The main concept behind any finite difference method is related to the definition of the derivative of a smooth function u at a point $x \in \mathbb{R}$:

$$u'(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h}.$$

For simplicity, we suppose that $u(x,t)$ is a function of only two independent variables, t and x.

Theorem 2.22. Let $k \geq 1$ be an integer and the function $u : \mathbb{R} \to \mathbb{R}$ be k times differentiable at the point $a \in \mathbb{R}$. The Taylor series expansions become

$$u(x+h,t) = u(x,t) + hu'(x,t) + \frac{h^2}{2!}u''(x,t) + \frac{h^3}{3!}u'''(x,t) + \cdots = \sum_{k=0}^{\infty} \frac{h^k}{k!}u^k(x,t), \tag{2.9}$$

$$u(x-h,t) = u(x,t) - hu'(x,t) + \frac{h^2}{2!}u''(x,t) - \frac{h^3}{3!}u'''(x,t) + \cdots = \sum_{k=0}^{\infty} (-1)^k \frac{h^k}{k!}u^k(x,t). \tag{2.10}$$

From above, a first order forward finite difference approximation to $u'(x,t)$ is

$$u'(x,t) = \frac{u(x+h,t) - u(x,t)}{h} + \mathcal{O}(h), \tag{2.11}$$

where $\mathcal{O}(h)$ denotes terms containing h and higher power of h. Similarly, a second order central difference approximation to $u''(x,t)$ is

$$u''(x,t) = \frac{u(x-h,t) - 2u(x,t) + u(x+h,t)}{h^2} + \mathcal{O}(h^2), \tag{2.12}$$

where $\mathcal{O}(h^2)$ denotes terms containing h^2 and higher power of h^2.

The convection-diffusion equation describes the flow of heat, particles, or other physical quantities in situations where there is both diffusion and convection. The general form of the convection-diffusion is

$$\frac{\partial u}{\partial t} + \mu \frac{\partial u}{\partial x} = \alpha \frac{\partial^2 u}{\partial x^2} \quad \text{for} \quad c \leq x \leq d \quad \text{and} \quad 0 \leq t \leq L, \tag{2.13}$$

where μ and α are the convection and diffusion coefficients, respectively. The equation (2.13) must be satisfied initial condition $u(x,0) = f(x)$ and boundary conditions $u(c,t) = g(t), u(d,t) = h(t)$ where f, g, h are given functions. A particular case $\mu = 0$ of Eq. (2.13) is called the diffusion equation, i.e.,

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \quad \text{for} \quad c \leq x \leq d \quad \text{and} \quad 0 \leq t \leq L. \tag{2.14}$$

To make a discretization of Eq. 2.13, we divide $[c,d]$ into M subintervals, each of equal length $h = (d-c)/M$. In the same manner, we define a grid for the N subintervals $l = L/N$. Then we make discretization at the grid point $u_m^n = u(x_m, t_n)$ where

$$x_m = c + mh \quad \text{and} \quad t_n = nl \tag{2.15}$$

for $1 \leq m \leq M$ and $1 \leq n \leq N$. By applying the forward time central space method, we have

$$\left(\frac{u_m^{n+1} - u_m^n}{l}\right) + \mu\left(\frac{u_{m+1}^n - u_{m-1}^n}{2h}\right) = \alpha\left(\frac{u_{m-1}^n - 2u_m^n + u_{m+1}^n}{h^2}\right).$$

Rearranging the above equation leads to

$$u_m^{n+1} = (p + \frac{1}{2}r)u_{m-1}^n + (1-2p)u_m^n + (p - \frac{1}{2}r)u_{m+1}^n$$

where $r = \mu l/n$ and $p = \alpha l/h^2$ are the convection and diffusion numbers, respectively.

Chapter 3

Gradient-based iterative algorithm with optimal convergence factor for a generalized Sylvester matrix equation

In this chapter, we discuss how to solve a generalized Sylvester matrix equations (1.3) indirectly by using an effective iterative algorithm. Let $m, n, p, q \in \mathbb{N}$ be such that $mq = np$. Consider the matrix equation (1.3) where $A_i \in M_{m,n}(\mathbb{R})$, $B_i \in M_{p,q}(\mathbb{R})$, $F \in M_{m,q}(\mathbb{R})$ are given constant matrices and $X \in M_{n,p}(\mathbb{R})$ is an unknown matrix to be solved.

3.1 A direct method for a generalized Sylvester matrix equation

A direct method to solve the matrix equation (1.3) is to take the vector operator and some properties of Kronecker product. Indeed, we arrive at an equivalent linear system $Px = b$ where

$$P := \sum_{i=1}^{p} B_i^T \otimes A_i.$$

The linear system $P\,\mathrm{vec}[X] = \mathrm{vec}[F]$ has a unique solution if and only if P is invertible, and in this case $\mathrm{vec}[X] = P^{-1}\,\mathrm{vec}[F]$. Due to the injectivity of $\mathrm{vec}[\cdot]$, we obtain the solution X from $\mathrm{vec}[X]$. Now, we discuss how to solve (1.3) indirectly by using an effective iterative algorithm.

3.2 Introducing a Gradient-based iterative algorithm for the matrix equation a generalized Sylvester matrix equation

According to the hierarchical identification principle the system (1.3) is decomposed into p subsystems. For each $i \in \{1, 2, ..., p\}$, set

$$M_i := F - \sum_{\substack{j=1 \\ j \neq i}}^{p} A_j X B_j. \tag{3.1}$$

Our aim is to approximate the solution of p subsystems:

$$M_i = A_i X B_i, \tag{3.2}$$

so that the following least-squares error is minimized:

$$L_i(X) := \frac{1}{2}\|A_i X B_i - M_i\|_F^2. \tag{3.3}$$

The gradient of each L_i can be computed as follows:

$$\begin{aligned}
\frac{\partial}{\partial x}L_i(X) &= \frac{1}{2}\frac{\partial}{\partial x}\,\mathrm{tr}[(A_iXB_i - M_i)^T(A_iXB_i - M_i)] \\
&= \frac{1}{2}\frac{\partial}{\partial x}\,\mathrm{tr}[(A_iXB_i)^T(A_iXB_i) - M_i^T(A_iXB_i) - (A_iXB_i)^TM_i + M_i^TM_i)] \\
&= \frac{1}{2}\left(\frac{\partial}{\partial x}\,\mathrm{tr}[B_i^TX^TA_i^TA_iXB_i] - 2\frac{\partial}{\partial x}\,\mathrm{tr}[M_i^TA_iXB_i]\right) \\
&= \frac{1}{2}(A_i^TA_iXB_iB_i^T + A_i^TA_iXB_iB_i^T) + \frac{1}{2}(2B_iM_i^TA_i)^T \\
&= A_i^T(F - \sum_{j=1}^{p} A_jXB_j)B_i^T. \qquad (3.4)
\end{aligned}$$

Let $X_i(k)$ be the estimate or iterative solution at iteration k, associated with the subsystem (3.2). From the gradient formula (3.4), the iterative scheme for $X_i(k)$ is given by the following equation:

$$X_i(k) = X(k-1) + \tau_1 A_i^T\left(F - \sum_{j=1}^{p} A_jXB_j\right)B_i^T, \qquad i = 1, 2, ..., p, \qquad (3.5)$$

where τ_1 is a convergent factor. According to the hierarchical identification principle, the unknown parameter X in (3.5) is replaced by its estimate $X(k-1)$. After taking the arithmetic mean of $X_i(k)$, we obtain the following process:

Algorithm 1: GIO algorithm for a generalized Sylvester matrix equation

$A_i \in \mathbb{R}^{m\times n}$, $B_i \in \mathbb{R}^{p\times q}$, for $i = 1, 2, \ldots, p$, and $F \in \mathbb{R}^{m\times q}$.
initialization;
Set $A_i' = A_i^T$ and $B_i' = B_i^T$. Choose $\tau_1 \in \mathbb{R}$ and $\epsilon > 0$. Set $k := 0$. Choose initial matrix $X(0)$.
while $k = 1, 2, \ldots, n$ **do**
  $E(k-1) = F - \sum_{i=1}^{p} A_iX(k-1)B_i$,
  if $\|E(k)\|_F/\|F\|_F < \epsilon$ **then**
    break;
  else
    $X(k) = X(k-1) + \tau_1(\sum_{i=1}^{p} A_i'E(k-1)B_i')$,
    $k = k + 1$;
  end
end

Note that the terms $E(k), A_i', B_i'$ were introduced in order to eliminate duplicated computations. To stop the process, one may impose a stopping rule such as the relative error $\|E(k)\|_F/\|F\|_F$ is less than a tolerance error ϵ. The convergence property of this algorithm depends on the convergent factor τ_1. A discussion of possible/optimal values of τ_1 will be in the next section.

3.3 Convergence Analysis

In this section, we show that the sequence of approximated solutions derived from Algorithm 1 converges to the exact solution. First, we transform a recursive equation of the error of approximated solutions into a first-order linear iterative system $x(k) = Tx(k-1)$ where $x(k)$ is a vector and T is an iteration matrix. Then we investigate the iteration matrix T to obtain the convergence criteria, convergence rate and error estimations. Finally we discuss the fastest convergence factor and find the number of iterations corresponding to a given satisfactory error.

Theorem 3.1. Assume that the matrix equation (1.3) has a unique solution X. Let $\tau_1 \in \mathbb{R}$. Then the approximate solutions derived from (3.5) converge to the exact solution for any initial value $X(0)$ if and only if

$$0 < \tau_1 < \frac{2}{\|P\|_2^2}. \tag{3.6}$$

In this case, the spectral radius of the associated iteration matrix $T = I_{np} - \tau_1 P^T P$ is given by

$$\rho[T] = \max\{|1 - \tau_1 \lambda_{\max}(P^T P)|, |1 - \tau_1 \lambda_{\min}(P^T P)|\}. \tag{3.7}$$

Proof. At each $k-th$ iteration, consider the error matrix $\hat{X}(k) = X(k) - X$. We have

$$\begin{aligned}\hat{X}(k) &= X(k-1) + \tau_1 \sum_{i=1}^{p} A_i^T E_i(k-1) B_i^T - X \\ &= \hat{X}(k-1) - \tau_1 \sum_{i=1}^{p} A_i^T E_i(k-1) B_i^T.\end{aligned}$$

We shall show that $\{X(k)\} \longrightarrow X$ by showing that $\{\hat{X}(k)\} \longrightarrow 0$ or $\{\operatorname{vec} \hat{X}(k)\} \longrightarrow 0$. By taking the vector operator to the above equation, we get

$$\begin{aligned}\operatorname{vec} \hat{X}(k) &= \operatorname{vec} \hat{X}(k-1) - \tau_1 \sum_{i=1}^{p} \operatorname{vec} A_i^T E_i(k-1) B_i^T \\ &= \operatorname{vec} \hat{X}(k-1) - \tau_1 \sum_{i=1}^{p} (B_i \otimes A_i^T) \operatorname{vec} \sum_{i=1}^{p} A_i \hat{X}(k-1) B_i \\ &= \operatorname{vec} \hat{X}(k-1) - \tau_1 \sum_{i=1}^{p} (B_i^T \otimes A_i)^T (\sum_{j=1}^{p} B_j^T \otimes A_j) \operatorname{vec} \hat{X}(k-1) \\ &= T \operatorname{vec} \hat{X}(k-1).\end{aligned} \tag{3.8}$$

We see that (3.8) is a 1st-order linear iterative system in the form $x(k) = Tx(k-1)$. Thus, $\{\operatorname{vec} \hat{X}(k)\} \longrightarrow 0$ for any initial values $X(0)$ if and only if the iteration matrix T has spectral radius less than 1. Since T is symmetric, all its eigenvalue are real. Note that any eigenvalue of T is of the form $1 - \tau_1 \lambda$ where λ is an eigenvalue of $P^T P$. Thus, its spectral radius is given by (3.7). It follows that $\rho[T] < 1$ if and only if

$$0 < \tau_1 \lambda_{\max}(P^T P) < 2 \quad \text{and} \quad 0 < \tau_1 \lambda_{\min}(P^T P) < 2. \tag{3.9}$$

Since P is invertible, the matrix P^TP is positive definite. Thus, $\lambda_{\max}(P^TP) > 0$. The condition (3.9) now becomes

$$0 < \tau_1 < \frac{2}{\lambda_{\max}(P^TP)} = \frac{2}{\|P\|_2^2}.$$

Theorem 3.2. Assume the hypothesis of Theorem 3.1, so that the sequence $\{X(k)\}$ converges to the exact solution X for any initial value $X(0)$.

(1). We have the following error estimates

$$\|X(k) - X\|_F \leq \rho[T] \, \|X(k-1) - X\|_F, \tag{3.10}$$

$$\|X(k) - X\|_F \leq \rho^k[T] \, \|X(0) - X\|_F. \tag{3.11}$$

Moreover, the asymptotic convergence rate of Algorithm 1 is governed by $\rho[T]$ in (3.7).

(2). Let $\varepsilon > 0$ be a satisfactory error. We have $\|X(k) - X\|_F < \epsilon$ after the k-th iteration for any $k \in \mathbb{N}$ that satisfies

$$k > \frac{\log \epsilon - \log \|X(0) - X\|_F}{\log \rho(T)}. \tag{3.12}$$

Proof. According to (3.8), we have

$$\begin{aligned} \|X(k) - X\|_F &= \|\hat{X}(k)\|_F \\ &= \|\operatorname{vec} \hat{X}(k)\|_F \\ &= \|T \operatorname{vec} \hat{X}(k-1)\|_F \\ &\leq \|T\|_2 \|\operatorname{vec} \hat{X}(k-1)\|_F. \end{aligned}$$

Since T is symmetric, we have $\|T\|_2 = \rho[T]$. Thus for each $k \in \mathbb{N}$, the approximation (3.10) holds. By induction, we obtain the estimation (3.11). The estimate (3.11) implies that the asymptotic convergence rate of the algorithm depends on $\rho[T]$. To prove the assertion, we have by taking anti-logarithms that the condition (3.12) is equivalent to

$$\rho^k(T) \|X(0) - X\|_F < \epsilon.$$

Thus if (3.12) holds, then $\|X(k) - X\|_F < \epsilon$. □

The convergence rate exhibits how fast of the sequence of the approximated solutions converging to the exact solution. Theorem 3.2 reveals that the smaller the spectral radius $\rho[T]$, the faster the approximated solutions go to the exact solution. Moreover, by taking $\epsilon = 0.5 \times 10^{-n}$ in (3.12), we have that $X(k)$ has an accuracy of n decimal digit if k satisfies

$$k > \frac{\log 0.5 - \log \|X(0) - X\|_F - n}{\log \rho(T)}.$$

Recall that the condition number of a matrix A (relative to the spectral norm) is defined by

$$\kappa(A) = \left(\frac{\lambda_{\max}(A^TA)}{\lambda_{\min}(A^TA)}\right)^{\frac{1}{2}}.$$

Theorem 3.3. Assume the hypothesis of Theorem 3.1. Then the optimal value of $\tau_1 > 0$ for which Algorithm 1 has the fastest asymptotic convergence rate is determined by

$$\tau_{opt} = \frac{2}{\lambda_{\max}(P^TP) + \lambda_{\min}(P^TP)}. \tag{3.13}$$

In this case, the spectral radius of the iteration matrix is given by

$$\begin{aligned}\rho[T] &= \frac{\lambda_{\max}(P^TP) - \lambda_{\min}(P^TP)}{\lambda_{\max}(P^TP) + \lambda_{\min}(P^TP)} \\ &= \frac{\kappa^2(P) - 1}{\kappa^2(P) + 1}.\end{aligned} \tag{3.14}$$

Proof. The convergence of Algorithm 1 implies that (3.6) holds. Then Algorithm 1 has the convergence rate as the same to the linear iteration (3.8), and thus, it is governed by the spectral radius $\rho[T]$ in (3.7). The fastest convergence rate is equivalent to the smallest of $\rho[T]$. Thus, we make the following minimization:

$$\begin{aligned} &\text{Minimize} && \rho[T] = \max\{|1 - \tau_1\lambda_{\min}(P^TP)|, |1 - \tau_1\lambda_{\max}(P^TP)|\} \\ &\text{subject to} && 0 < \tau_1 < \frac{2}{\lambda_{\max}(P^TP)}. \end{aligned}$$

Thus, the optimal value is reached at (3.13) so that the minimum is given by (3.14). □

We see that if the condition number of P is closer to 1 then the approximate solutions converge faster to the exact solution. Note that the condition number of P is close to 1 if and only if the maximum eigenvalue of P^TP is close to the minimum eigenvalue of P^TP.

3.4 The GIO algorithm for the Sylvester equation

In this section, we discuss the gradient-based iterative algorithm with optimal convergent factor for solving Sylvester matrix equation. Moreover we discover convergence criteria, convergence rate, error estimate and optimal factor.

Let $m, n, p, q \in \mathbb{N}$ be such that $m = n$ and $p = q$. Consider the Sylvester matrix equation (1.5) where $A \in M_{m,n}(\mathbb{R})$, $B \in M_{p,q}(\mathbb{R})$, $F \in M_{m,q}(\mathbb{R})$ are given constant matrices and $X \in M_{n,p}(\mathbb{R})$ is an unknown matrix to be solved. Suppose that the matrix equation (1.5) has a unique solution, i.e., $P_s := I_p \otimes A + B^T \otimes I_n$ is invertible.

Algorithm 2: GIO algorithm for a Sylvester equation

$A \in M_{m,n}(\mathbb{R})$, $B \in M_{p,q}(\mathbb{R})$, $F \in M_{m,q}(\mathbb{R})$.

initialization;

Set $A' = A^T$ and $B' = B^T$. Choose $\tau_2 \in \mathbb{R}$ and $\epsilon > 0$. Set $k := 0$. Choose initial matrix $X(0)$.

while $k = 1, 2, \ldots, n$ **do**

$E(k) = F - AX(k) - X(k)B$,

if $\|E(k)\|_F / \|F\|_F < \epsilon$ **then**

break;

else

$X(k) = X(k-1) + \tau_2 (A'E(k-1)B')$,

$k = k + 1$;

end

end

Corollary 3.4. Assume that the Sylvester matrix equation (1.5) has a unique solution X. Let $\tau_2 \in \mathbb{R}$. Then the following hold:

(i) The approximate solutions generated by Algorithm 2 converge to the exact solution for any initial value $X(0)$ if and only if

$$0 < \tau_2 < \frac{2}{\|P_s\|_2^2}. \tag{3.15}$$

In this case, the spectral radius of the associated iteration matrix $S = I_{np} - \tau_2 P_s^T P_s$ is given by

$$\rho[S] = \max\{|1 - \tau_2 \lambda_{\max}(P_s^T P_s)|, |1 - \tau_2 \lambda_{\min}(P_s^T P_s)|\}. \tag{3.16}$$

(ii) The asymptotic convergence rate of Method 2 is governed by $\rho[S]$ in (3.16).

(iii) The optimal value of $\tau_2 > 0$ for which Algorithm 2 has the fastest asymptotic convergence rate is determined by

$$\tau_{opt} = \frac{2}{\lambda_{\max}(P_s^T P_s) + \lambda_{\min}(P_s^T P_s)}. \tag{3.17}$$

Remark 3.5. Note that P_s is the Kronecker sum of A and B^T. Thus, if A and B are positive semidefinite, then

$$\|P_s\|_2^2 = \lambda_{\max}(P_s^T P_s) = \lambda_{\max}^2(P_s) = (\lambda_{\max}(A) + \lambda_{\max}(B))^2,$$
$$\lambda_{\min}(P_s^T P_s) = \lambda_{\min}^2(P_s) = (\lambda_{\min}(A) + \lambda_{\min}(B))^2.$$

3.5 Numerical simulations with discussion

In this section, we show the capability and efficiency of the proposed algorithm by illustrating some numerical examples. To compare the performance of any algorithms,

we must use the same PC environment, and consider informed errors together with iteration numbers (IT) and computational times (CT: in seconds). Our iterations have been carried out by MATLAB R2013a, Intel(R) Core(TM) i5-760 CPU @ 2.80 GHz, RAM 8.00 GB PC environment. We measure the computational time taken for an iterative process by the MATLAB functions **tic** and **toc**. In Example 3.6, we show that our algorithm is also efficient although matrices are non-square and we discuss the effect of changing the convergent factor τ. In Example 3.7, we consider a larger square matrix system and show that our algorithm is still efficient. In Example 3.8, we compare the efficiency of our algorithm to another recent iterative algorithms. The matrix equation considered in this example is the Sylvester equation with square coefficient matrices since it fits with all of the recent algorithms. In all illustrated examples, we compare the efficiency of iterative algorithms to the direct method $x = P^{-1}b$ mentioned in section 3.1.

Example 3.6. Consider the matrix equation $A_1XB_1 + A_2XB_2 + A_3XB_3 = F$ when $A_1, A_2, A_3 \in \mathbb{R}^{40\times 60}$, $B_1, B_2, B_3 \in \mathbb{R}^{20\times 30}$ and $F \in \mathbb{R}^{40\times 30}$ are tridiagonal matrices given by

$$A_1 = \text{tridiag}(-2, 2, -2), \quad A_2 = \text{tridiag}(2, -2, 5), \quad A_3 = \text{tridiag}(2, -1, 2),$$
$$B_1 = \text{tridiag}(4, 3, -1), \quad B_2 = \text{tridiag}(1, -2, -1), \quad B_3 = \text{tridiag}(3, 1, 3).$$

Here, the exact solution is given by $X = \text{tridiag}(1, -1, 1)$. We apply Algorithm 1 to compute the sequence $X(k)$ of approximated solutions. Take initial point

$$X(0) = 10^{-6} \times \text{tridiag}(1, 1, 1).$$

The optimal convergent factor can be computed as follows:

$$\begin{aligned}\tau_{opt} &= \frac{2}{\lambda_{\min}(P^TP) + \lambda_{\max}(P^TP)} \\ &\approx \frac{2}{4.15 \times 10^{-13} + 1009.74} \\ &\approx 0.0019806.\end{aligned}$$

The effect of changing convergent factors τ is illustrated in Figure 3.1. We see that as k large enough, the relative error $\|E(k)\|_F/\|F\|_F$ for τ_{opt} goes faster to 0 than another convergent factors. If τ does not satisfy the condition (3.6), then the approximated solutions diverge for the given initial matrices. Moreover, Table 3.1 shows that the computational time of our algorithm (GIO) is significantly less than the time of the direct method. Table 3.1 also demonstrates that, when we fix the error $\|E(k)\|_F$ to be less than 5×10^{-3}, the GIO algorithm outperforms another GI algorithms with different convergent factors in both iteration numbers and computational times.

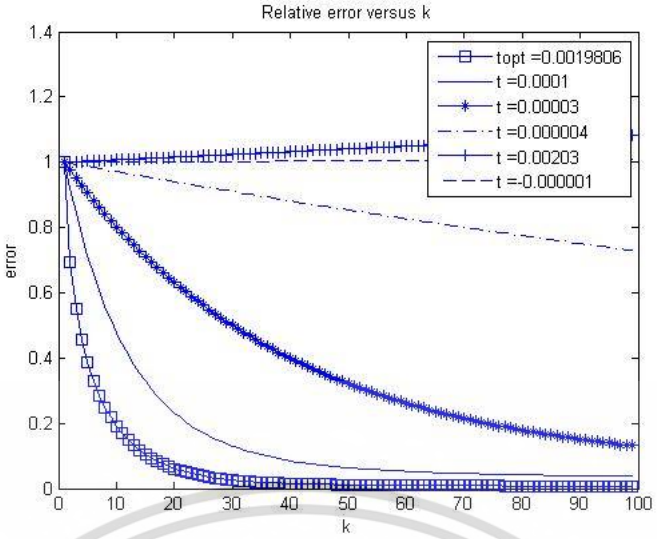


Figure 3.1*:* Relative error for Example 3.6

Table 3.1*:* Iteration numbers and computational times for Example 3.6

Method	IT	CT
Direct	-	3.1380
GIO	161	0.0413
GI ($\tau = 0.0001$)	3061	0.2508
GI ($\tau = 0.00003$)	10204	0.8994

Example 3.7. Consider the matrix equation $A_1XB_1 + A_2XB_2 + A_3XB_3 = F$ where all matrices are 100×100 tridiagonal matrices given by

$$A_1 = \text{tridiag}(1, 2, 1), \quad A_2 = \text{tridiag}(-1, -2, -1), \quad A_3 = \text{tridiag}(-1, 3, -1),$$
$$B_1 = \text{tridiag}(2, 2, 3), \quad B_2 = \text{tridiag}(1, 2, -2), \quad B_3 = \text{tridiag}(3, 2, -1).$$

Here, the exact solution is $X = \text{tridiag}(1, 1, 1)$. To apply Algorithm 1, we take initial matrix

$$X(0) = 10^{-6} \times \text{tridiag}(0, 2, 0).$$

We can compute $\tau_{opt} \approx 0.002553$. Figure 3.2 shows that the relative error $\|E(k)\|_F/\|F\|_F$ for τ_{opt} goes faster to 0 than another convergent factors. If τ does not satisfy (3.6), then the approximate solutions diverge for the given initial matrices. From Table 3.2, we see that the computational time of our algorithm is significantly less than the time of the direct method. Furthermore, when the satisfactory error $\|E(k)\|_F$ is less than $\epsilon = 0.5$, the GIO algorithm has more efficient than another GI algorithms in both iteration numbers and computational times.

Example 3.8. Consider the Sylvester equation $AX+XB = F$, where $A, X, B, F \in \mathbb{R}^{10\times 10}$ are given by $A = \text{tridiag}(-1, 3, 1)$, $B = \text{tridiag}(-3, 2, 3)$, $X = \text{tridiag}(-3, 1, 4)$. We compare

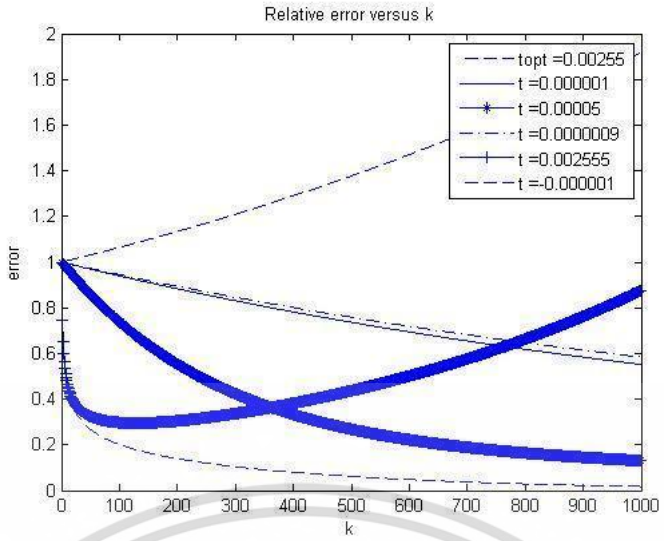


Figure 3.2: Relative errors for Example 3.7

Table 3.2: Iteration numbers and computational times for Example 3.7

Method	IT	CT
Direct	-	53.4063
GIO	389	0.5439
GI ($\tau = 0.00005$)	19314	28.0245
GI ($\tau = 0.000001$)	96557	148.4039

the efficiency of our algorithm (GIO) with another iterative algorithms such as GI, LS, RGI, MGI, JGI and AJGI. We choose the same convergent factor $\tau = 0.01836$ and the same initial matrix $X(0) = \text{tridiag}(0, 10^{-6}, 0)$. To compare the efficiency of these algorithms, we fix the iteration number to be 50 and consider the relative errors $\|E(k)\|_F/\|F\|_F$. The results are displayed in Figure 3.3. The iteration numbers and the computational times when we fix the error $\|E(k)\|_F$ to be less than 5×10^{-3} are illustrated in Table 3.3. We see that our algorithm is outperform to the direct method and another iterative algorithms with less iteration number and lower computational time. In particular, the approximated solutions generated from JGI algorithm diverge.

Table 3.3: Iteration numbers and computational times for Example 3.8

Method	GIO	GI	LS	RGI	MGI	JGI	AJGI	Direct
IT	18	33	167	70	25	-	51	-
CT	0.000273	0.000589	0.0114	0.0012	0.000789	-	0.0014	0.1704

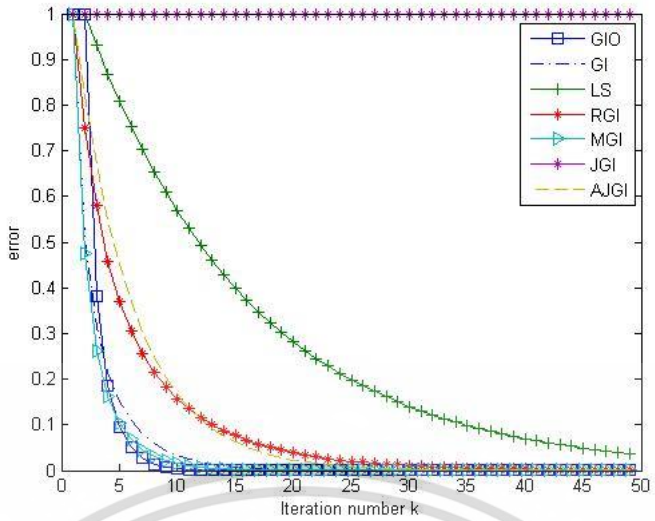


Figure 3.3*:* Relative errors for Example 3.8

3.6 An Application to discretization of the convection-diffusion equation

In this section, we apply the GIO algorithm to a discretization of convection-diffusion equation (2.13). Recall that the convection-diffusion equation (2.13) with initial condition $u(x,0) = f(x)$ and boundary conditions $u(c,t) = g(t), u(d,t) = h(t)$ where f, g, h are given functions is applied to the forward time central space method, we have

$$u_m^{n+1} = (p + \frac{1}{2}r)u_{m-1}^n + (1-2p)u_m^n + (p - \frac{1}{2}r)u_{m+1}^n$$

where $r = \mu l/n$ and $p = \alpha l/h^2$ are the convection and diffusion numbers, respectively. We can transform (2.13) into a linear system of MN unknowns $u_{11}, ..., u_{MN}$ in the form

$$P_{CD}\,\text{vec}(U) = b, \tag{3.18}$$

where $U = [u_m^n], P_{CD} \in \mathbb{R}^{MN \times MN}$ has $N \times N$ blocks of the form I_M on its diagonal and tridiag$(-p - \frac{1}{2}r, -1 + 2p, -p + \frac{1}{2}r)$ under its diagonal. The vector b is partitioned in M periods as $[b_1^T \;\; b_2^T \;\; \dots \; b_N^T]^T$ where $b_1 = \begin{bmatrix}\phi(1) & \phi(2) & \cdots & \phi(m)\end{bmatrix}^T$ and

$$b_j = \begin{bmatrix} (p + \frac{1}{2}r)g(t + (j-1)l) \\ 0 \\ \vdots \\ 0 \\ (p - \frac{1}{2}r)h(t + (j-1)l) \end{bmatrix}, \quad j = 2, \dots, N \tag{3.19}$$

here $\phi(i) = (p+\frac{1}{2}r)f(c+(i-1)h)+(1-2p)f(c+ih)+(p-\frac{1}{2}r)f(c+(i+1)h)$ for $i = 1, 2, \dots, m-1$ and $\phi(m) = h(t+l)$. Here is an example of P_{CD} and b_j where $j = 1, 2, \dots, N$ when $M = 3$ and $N = 2$:

$$P_{CD} = \left[\begin{array}{cccc:cccc} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ \hdashline (-1+2p) & (-p+\frac{1}{2}r) & & & 1 & & & \\ (-p-\frac{1}{2}r) & (-1+2p) & (-p+\frac{1}{2}r) & & & 1 & & \\ & (-p-\frac{1}{2}r) & (-1+2p) & (-p+\frac{1}{2}r) & & & 1 & \\ & & (-p-\frac{1}{2}r) & (-1+2p) & & & & 1 \end{array}\right],$$

$$b = \left[\begin{array}{c} b_1 \\ \hdashline b_2 \end{array}\right] = \left[\begin{array}{c} (p+\frac{1}{2}r)f(c)+(1-2p)f(c+h)+(p-\frac{1}{2}r)f(c+2h) \\ (p+\frac{1}{2}r)f(c+h)+(1-2p)f(c+2h)+(p-\frac{1}{2}r)f(c+3h) \\ \hdashline h(t+l) \\ (p+\frac{1}{2}r)g(t+l) \\ 0 \\ (p-\frac{1}{2}r)h(t+l) \end{array}\right].$$

We can see that Eq.(3.18) is the generalized Sylvester equation where $p = 1$, $A = P_{CD}$, $X = \text{vec}(U)$, $B = I$ and $F = b$. From Algorithm 1, we obtain the following:

Algorithm 3: GIO for a Convection diffusion equation

$M, N \in \mathbb{N}$ as number of partition.

initialization;

$P'_{CD} = P_{CD}^T$. Choose $u(0) \in \mathbb{R}^{MN}$ and $\epsilon > 0$. For each $m = 1, 2, \ldots, M$ and $n = 1, 2, \ldots, N$, compute x_m, t_n as in Eq.(2.15) and

$$\tau_{opt} = \frac{2}{\lambda_{\max}(P_{CD}^T P_{CD}) + \lambda_{\min}(P_{CD}^T P_{CD})}.$$

while $k = 1, 2, \ldots, n$ **do**

  $E(k) = b - P_{CD}u(k)$,

  if $\|E(k)\|_F / \|b\|_F < \epsilon$ **then**

    break;

  else

    $u(k) = u(k-1) + \tau_{opt} P'_{CD} E(k-1)$,

    $k = k + 1$.

  end

end

Now, we provide a numerical experiment for a convection-diffusion equation.

Example 3.9. Consider the convection-diffusion equation

$$\frac{\partial u}{\partial t} + 0.1\frac{\partial u}{\partial x} = 0.01\frac{\partial^2 u}{\partial x^2} \quad \text{for} \quad 0 \le x \le 1 \quad \text{and} \quad 0 \le t \le 10 \tag{3.20}$$

with the initial and boundary conditions given as:

$$u(x,0) = 100x \quad \text{and} \quad u(0,t) = u(100,t) = 0.$$

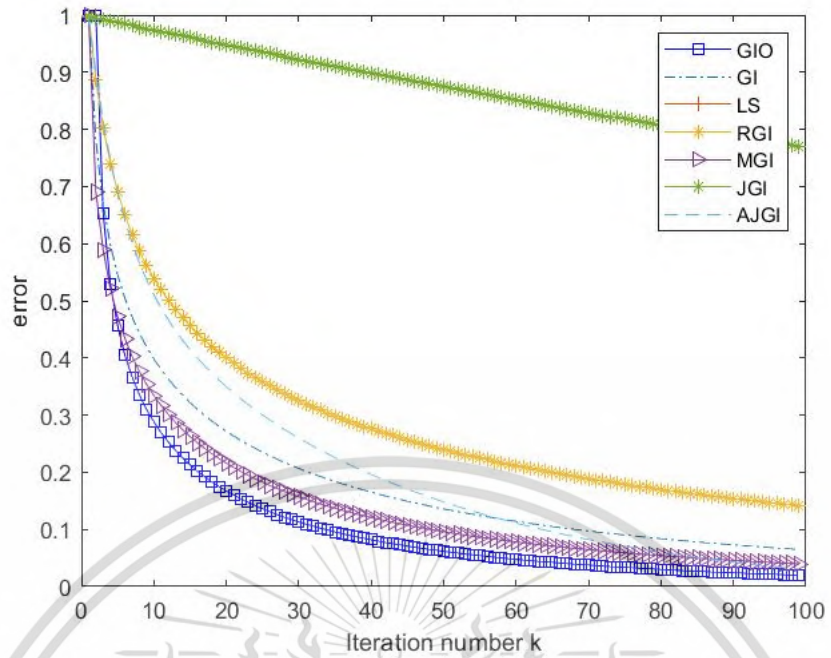


Figure 3.4: Relative errors for Example 3.9

Let $M = 5, N = 10$, so that P_{CD} is of dimension 50×50. In this case, we have $h = 0.2, l = 1, r = 0.5$ and $p = 0.25$. We choose $u(0) = 10^{-6}\begin{bmatrix}1 \cdots 1\end{bmatrix} \in \mathbb{R}^{50}$. After compiling Algorithm 3 for 100 iterations, we see from Figure 3.4 that the relative error $\|E(k)\|_F$ goes faster to 0 than another algorithms such as GI, LS, RGI, MGI,JGI and AJGI. Moreover Table 3.4 displays comparison of numerical and direct solutions for the convection-diffusion equation.

A particular case $\mu = 0$ of Eq. (2.13) is called the diffusion equation. In this case, the formulas of P_{CD} and $b_1, \ldots, b_N$ are reduced as $r = 0$.

Example 3.10. Consider the diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad \text{for} \quad 0 \leq x \leq 1 \quad \text{and} \quad 0 \leq t \leq 10$$

with the initial and boundary conditions given as:

$$u(x, 0) = 6\sin(\pi x) \quad \text{and} \quad u(0, t) = u(1, t) = 0. \tag{3.21}$$

The exact solution is

$$u^*(x, t) = 6e^{-\pi^2 t}\sin(\pi x).$$

Let $M = 10, l = 0.01$ In this case, we have $h = 0.1$, and $p = 1$. We choose initial matrix $u(0) = 10^{-6}\begin{bmatrix}1 \cdots 1\end{bmatrix} \in \mathbb{R}^{100}$. After compiling Algorithm 3 for 200 iterations (Figure 3.5), we see that our algorithm is outperform to another iterative algorithms with less iteration number and lower computational time. The 3D-plot in Figure 3.6 shows that the iterative solution is well approximated to the exact solution.

Table 3.4*:* Iteration numbers, computational times and errors for Example 3.9

Method	IT	CT	Error
Direct	-	2.085	0
GIO	100	0.0113	0.0199
GI	100	0.0281	0.0648
LS	100	0.0469	1.6574
RGI	100	0.0324	0.1417
MGI	100	0.0313	0.0397
JGI	100	0.2813	0.7698
AJGI	100	0.0938	0.0307

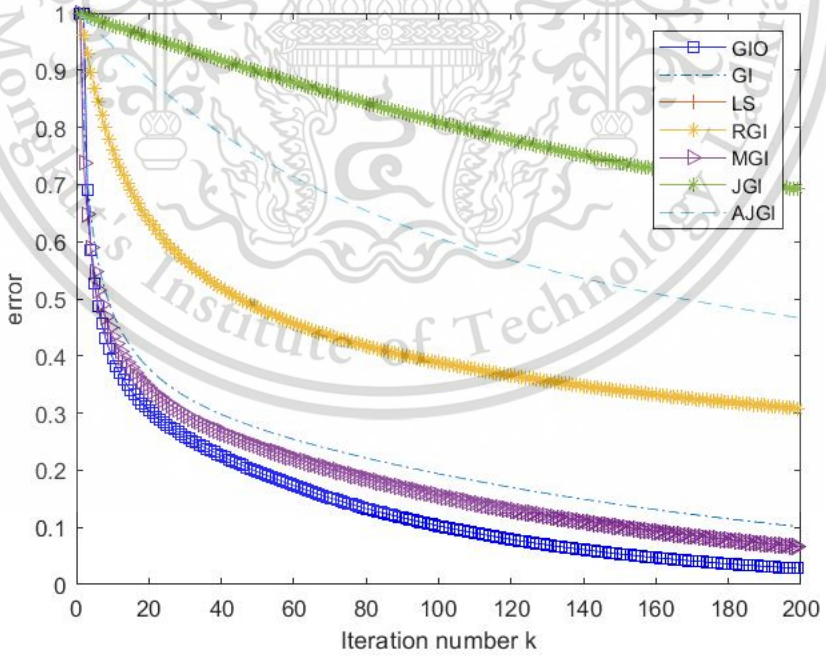


Figure 3.5*:* Relative errors for Example 3.10

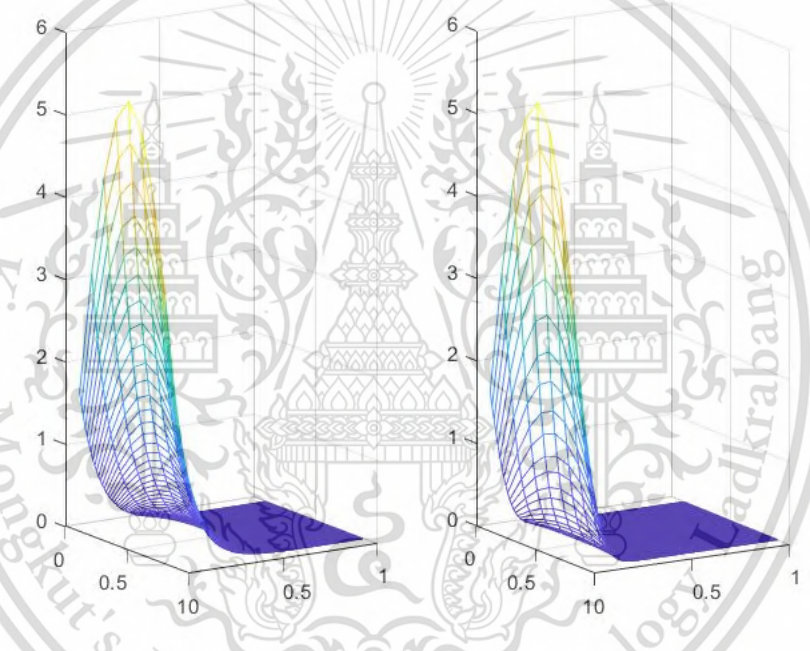


Figure 3.6*:* The exact (left) and the iterative (right) solutions for Example 3.10

Chapter 4

Gradient-based iterative algorithm with optimal convergence factor for a generalized Sylvester-transpose matrix equation

In this chapter, we discuss how to solve a generalized Sylvester-transpose matrix equations (1.4) indirectly by using an effective iterative algorithm. Consider a generalized Sylvester-transpose matrix equation (1.4) where $A_i \in M_{m,n}(\mathbb{R})$, $B_i \in M_{p,q}(\mathbb{R})$, $C_j \in M_{m,p}(\mathbb{R})$, $D_j \in M_{n,q}(\mathbb{R})$, $F \in M_{m,q}(\mathbb{R})$ are given coefficient matrices and $X \in M_{n,p}(\mathbb{R})$ is an unknown matrix to be determined. The dimension matching of matrices is assumed to be $mq = np$.

4.1 A direct method for the generalized Sylvester-transpose matrix equation

A direct method to solve the matrix equation (1.4) is to take the vector operator and utilize Lemma 2.6. Indeed, we arrive at an equivalent linear system $Qx = b$ where

$$Q := \sum_{i=1}^{p}(B_i^T \otimes A_i) + \sum_{j=1}^{q}(D_j^T \otimes C_j)K_{np} \in \mathbb{R}^{np\times np},$$

$x = \text{vec}[X]$ and $b = \text{vec}[F]$. From now on, assume that Q is invertible. So, the matrix equation (1.4) has a unique solution

$$x = Q^{-1}b. \tag{4.1}$$

However, if the dimensions of A_i, B_i, C_j, D_j are not small, e.g. $10^2 \times 10^2$, then the dimension of Q is $10^4 \times 10^4$. Such a dimension problem leads to computational difficulty for computation and inversion of large matrices. Hence, this approach is only applicable for small dimensional matrices.

4.2 Introducing a gradient-based iterative algorithm for a generalized Sylvester-transpose matrix equation

We now propose an effective iterative algorithm to solve (1.4). If $p \geqslant q$ then we set $C_j = 0$ and $D_j = 0$ for any $j > q$. If $q > p$ then we set $A_i = 0$ and $B_i = 0$ for any $i > p$. For each $i = 1, 2, \ldots, p$, we assume that

$$M_i := F - \left(\sum_{s\neq i}(A_sXB_s + C_sX^TD_s)\right). \tag{4.2}$$

From the main system (1.4), we would like to solve p subsystems

$$A_iXB_i + C_iX^TD_i = M_i \qquad i = 1, 2, \ldots, p, \tag{4.3}$$

so that the following errors are minimized:

$$L_i(X) := \|A_iXB_i + C_iX^TD_i - M_i\|_F^2, \qquad i = 1, 2, \ldots, p. \tag{4.4}$$

We can derive the gradient of each L_i as follows:

$$\begin{aligned}
\frac{\partial}{\partial X}L_i(X) &= \frac{\partial}{\partial X}\,\mathrm{tr}[(A_iXB_i + C_iX^TD_i - M_i)^T(A_iXB_i + C_iX^TD_i - M_i)] \\
&= \frac{\partial}{\partial X}\,\mathrm{tr}[(A_iXB_i)^T(A_iXB_i) + (C_iX^TD_i)^T(A_iXB_i) - M_i^T(A_iXB_i) \\
&\quad +(A_iXB_i)^T(C_iX^TD_i) + (C_iX^TD_i)^T(C_iX^TD_i) - M_i^T(C_iX^TD_i) \\
&\quad -(A_iXB_i)^TM_i - (C_iX^TD_i)^TM_i + M_i^TM_i \\
&= \frac{\partial}{\partial X}\,\mathrm{tr}(B_i^TX^TA_i^TA_iXB_i) + \frac{\partial}{\partial X}\,\mathrm{tr}(D_i^TXC_i^TA_iXB_i) - \frac{\partial}{\partial X}\,\mathrm{tr}(M_i^TA_iXB_i) \\
&\quad +\frac{\partial}{\partial X}\,\mathrm{tr}(B_i^TX^TA_i^TC_iX^TD_i) + \frac{\partial}{\partial X}\,\mathrm{tr}(D_i^TXC_i^TC_iX^TD_i) - \frac{\partial}{\partial X}\,\mathrm{tr}(M_i^TC_iX^TD_i) \\
&\quad -\frac{\partial}{\partial X}\,\mathrm{tr}(B_i^TX^TA_i^TM_i) - \frac{\partial}{\partial X}\,\mathrm{tr}(D_i^TXC_i^TM_i) + \frac{\partial}{\partial X}\,\mathrm{tr}(M_i^TM_i) \\
&= 2\big[A_i^T(A_iXB_i + C_iX^TD_i - M_i)B_i^T + D_i(A_iXB_i + C_iX^TD_i - M_i)^TC_i\big].
\end{aligned} \tag{4.5}$$

Let $X_i(k)$ be the approximated solution of the system (4.3) at iteration k. The recursive formula of $X_i(k)$ come from the gradient formula of $L_i(X)$ as follows:

$$X_i(k) = X_i(k-1) + \mu\frac{\partial}{\partial X}L_i(X),$$

where μ is a step size parameter. To avoid duplicated computation, we introduce a matrix

$$E = F - \sum_{i=1}^{p} A_iXB_i - \sum_{j=1}^{q} C_jX^TD_j.$$

Taking the arithmetic mean of $X_1(k), \ldots, X_p(k)$ to get $X(k)$:

$$X(k) = \frac{1}{p}\sum_{i=1}^{p} X_i(k) = X(k-1) + \tau_3\left(\sum_{i=1}^{p} A_i^TEB_i^T + \sum_{j=1}^{q} D_jE^TC_j\right), \tag{4.6}$$

where $\tau_3 := -2\mu/p$ is called a convergence factor. The hierarchical identification principle suggests to replace the unknown solution X in (4.6) by its previous estimate $X(k-1)$. Thus we obtain the following procedure:

Algorithm 4: GIO algorithm for a generalized Sylvester-transpose equation

$A_i \in \mathbb{R}^{m\times n}$, $B_i \in \mathbb{R}^{p\times q}$, $C_j \in \mathbb{R}^{m\times p}$, $D_j \in \mathbb{R}^{n\times q}$ for $i = 1, 2, \ldots, p$, $j = 1, 2, \ldots, q$ and $F \in \mathbb{R}^{m\times q}$.

initialization;

if $p \geq q$ **then**

  $C_j = 0$ and $D_j = 0$ for any $j > q$;

else

  $A_i = 0$ and $B_i = 0$ for any $i > p$.

end

Set $A_i' = A_i^T$ and $B_i' = B_i^T$. Choose $\tau \in \mathbb{R}$. Set $k := 0$. Choose initial matrix $X(0)$.

while $k = 0, 1, 2, \ldots, n$ **do**

  $E(k) = F - \sum_{i=1}^{p} A_i X(k) B_i - \sum_{j=1}^{q} C_j X^T(k) D_j$.

  if $\|E(k)\|_F / \|F\|_F < \epsilon$ **then**

    break;

  else

    $X(k+1) = X(k) + \tau_3 \left(\sum_{i=1}^{p} A_i' E(k) B_i' + \sum_{j=1}^{q} D_j E^T(k) C_j \right)$,

    update k.

  end

end

For convenience, we write $X^T(k)$ and $E^T(k)$ for $X(k)^T$ and $E(k)^T$, respectively. The matrices $E(k), A_i', B_i'$ were introduced to avoid duplicate manipulations.

Remark 4.1. To break the procedure, if $\|F\|_F$ is close to zero, then we should consider the error $\|E(k)\|_F$ or $\|X(k) - X(k-1)\|_F$ instead of the relative error $\|E(k)\|_F / \|F\|_F$.

The convergent property of the algorithm relies on the convergence factor τ_3, which will be discussed in the next section.

4.3 Convergence analysis

In this section, we discuss a convergence criteria for the applicability of Algorithm 4. Then, we analyse the performance of the algorithm through its convergence rate and error estimates. The last job is to determine the convergence factor for which the algorithm fits with the fastest asymptotic behaviour. The main idea for the analysis starts with transforming a recursive equation of the error of approximated solutions into a fixed-point iteration $x(k) = Sx(k-1)$, where $S : \mathbb{R}^{np} \to \mathbb{R}^{np}$ is a contraction mapping. Then, we apply the following Banach fixed-point theorem (Theorem 2.20) to analysis the proposed algorithm.

4.3.1 Convergence criteria

From Algorithm 4, at each k-th iteration, we start with considering the error matrix $\hat{X}(k) = X(k) - X$. Indeed, we have

$$\begin{aligned}\hat{X}(k) &= X(k-1) + \tau_3 \left(\sum_{i=1}^{p} A_i^T E(k-1) B_i^T + \sum_{j=1}^{q} D_j E^T(k-1) C_j \right) - X \\ &= \hat{X}(k-1) + \tau_3 \left(\sum_{i=1}^{p} A_i^T E(k-1) B_i^T + \sum_{j=1}^{q} D_j E^T(k-1) C_j \right), \end{aligned} \tag{4.7}$$

and

$$E(k-1) = -\left(\sum_{i=1}^{p} A_i \hat{X}(k-1) B_i + \sum_{j=1}^{q} C_j \hat{X}^T(k-1) D_j \right).$$

Thus, Lemma 2.6 implies that

$$\begin{aligned}\operatorname{vec} E(k-1) &= -\sum_{i=1}^{p} (B_i^T \otimes A_i) \operatorname{vec} \hat{X}(k-1) - \sum_{j=1}^{q} (D_j^T \otimes C_j) \operatorname{vec} \hat{X}^T(k-1) \\ &= -\sum_{i=1}^{p} (B_i^T \otimes A_i) \operatorname{vec} \hat{X}(k-1) - \sum_{j=1}^{q} (D_j^T \otimes C_j) K_{np} \operatorname{vec} \hat{X}(k-1) \\ &= \left(\sum_{i=1}^{p} (B_i^T \otimes A_i) + \sum_{j=1}^{q} (D_j^T \otimes C_j) K_{np} \right) \operatorname{vec} \hat{X}(k-1) \\ &= -Q \operatorname{vec} \hat{X}(k-1). \end{aligned} \tag{4.8}$$

Taking the vector operator to the equation (4.7) and utilizing (4.8), we get

$$\begin{aligned}\operatorname{vec} \hat{X}(k) &= \operatorname{vec} \hat{X}(k-1) + \tau_3 \left(\sum_{i=1}^{p} \operatorname{vec} A_i^T E(k-1) B_i^T + \sum_{j=1}^{q} \operatorname{vec} D_j E^T(k-1) C_j \right) \\ &= \operatorname{vec} \hat{X}(k-1) + \tau_3 \left(\sum_{i=1}^{p} (B_i \otimes A_i^T) \operatorname{vec} E(k-1) + \sum_{j=1}^{q} (C_j^T \otimes D_j) K_{mq} \operatorname{vec} E(k-1) \right) \\ &= \operatorname{vec} \hat{X}(k-1) - \tau_3 \left(\sum_{i=1}^{p} (B_i \otimes A_i^T) Q \operatorname{vec} \hat{X}(k-1) + \sum_{j=1}^{q} K_{pn} (D_j \otimes C_j^T) Q \operatorname{vec} \hat{X}(k-1) \right) \\ &= (I_{np} - \tau_3 Q^T Q) \operatorname{vec} \hat{X}(k-1). \end{aligned}$$

Letting $S = I_{np} - \tau_3 Q^T Q$ and $x(k) = \operatorname{vec} \hat{X}(k-1)$, we get a linear iteration

$$x(k) = Sx(k-1). \tag{4.9}$$

Using Theorem 2.20 and [31, Theorem 1], we deduce that the following are equivalent:

(i) the sequence $\{X(k)\}$ converges to X for any initial value $X(0)$;

(ii) S is a contraction mapping (here, we view $S : \mathbb{R}^{np} \to \mathbb{R}^{np}$ as a mapping);

(iii) the spectral norm of S is less than 1.

Since S is symmetric, all its eigenvalue are real. Note that the eigenvalues of S are of the form $1 - \tau_3\lambda$ where λ is any eigenvalue of Q^TQ. We can compute the spectral radius of S as follows:

$$\rho[S] = \max\{|1 - \tau_3\lambda_{\max}(Q^TQ)|, |1 - \tau_3\lambda_{\min}(Q^TQ)|\}. \tag{4.10}$$

Thus, $\rho[S] < 1$ if and only if

$$0 < \tau_3\lambda_{\max}(Q^TQ) < 2 \quad \text{and} \quad 0 < \tau_3\lambda_{\min}(Q^TQ) < 2. \tag{4.11}$$

Since Q is invertible, the matrix Q^TQ is positive definite and hence, $\|Q\|_2^2 = \lambda_{\max}(Q^TQ) > 0$. The condition (4.11) now becomes

$$0 \ < \ \tau_3 \ < \ \frac{2}{\|Q\|_2^2}. \tag{4.12}$$

We summarize convergence criteria for Algorithm 4 as follows:

Theorem 4.3.1. Consider the matrix equation (1.4) under the assumption that the matrix Q is invertible (i.e. Eq. (1.4) has a unique solution). Let $\tau_3 \in \mathbb{R}$. Then a necessary and sufficient condition for which Algorithm 4 is applicable for any initial matrix X(0) is the condition (4.12).

4.3.2 Performance of the algorithm

We now discuss the performance of Algorithm 4 through its convergence rate and error estimates. Considering Eq. (4.9), we get

$$\begin{aligned} \|X(k) - X\|_F \ &= \ \|\hat{X}(k)\|_F \\ &= \ \|\operatorname{vec}\hat{X}(k)\|_F \\ &= \ \|S\operatorname{vec}\hat{X}(k-1)\|_F \\ &\leq \|S\|_2\|\operatorname{vec}\hat{X}(k-1)\|_F. \end{aligned}$$

Since S is symmetric, we have $\|S\|_2 = \rho[S]$. Thus the Eq. (2.7) implies that

$$\|X(k) - X\|_F \ \leq \ \rho[S] \ \|X(k-1) - X\|_F. \tag{4.13}$$

By induction, we obtain

$$\|X(k) - X\|_F \ \leq \ \rho^k[S] \ \|X(0) - X\|_F. \tag{4.14}$$

According to the estimate (4.14), the asymptotic convergence rate of the algorithm relies on $\rho[S]$. Moreover, given an error $\epsilon > 0$, we would like to find the iteration number k for which

$$\rho^k(S)\|X(0) - X\|_F \ < \ \epsilon. \tag{4.15}$$

By taking the 10th-base logarithms, we obtain the following equivalent requirement:

$$k > \frac{\log\epsilon - \log\|X(0) - X\|_F}{\log\rho(S)}. \tag{4.16}$$

Moreover, by Theorem 2.20(iii), we have

$$\|X(k) - X\|_F \leq \frac{\rho^k[S]}{1-\rho[S]} \|X(1) - X(0)\|_F, \tag{4.17}$$

$$\|X(k+1) - X\|_F \leq \frac{\rho[S]}{1-\rho[S]} \|X(k+1) - X(k)\|_F. \tag{4.18}$$

The following results are discussed to concluding the convergence rate and several estimates of the proposed algorithm.

Theorem 4.3.2. Suppose the convergence factor τ_3 is chosen so that Algorithm 4 is applicable for any initial matrix X(0).

(i) The spectral radius $\rho[S]$ in (4.10) governs the asymptotic convergence rate of the algorithm.

(ii) Equations (4.13) and (4.14) show the error estimates $\|X(k) - X\|_F$ compared to the previous step and the first step, respectively. Meanwhile, the error at each iteration reduces from the previous one.

(iii) The prior and posterior estimates are presented in (4.17) and (4.18), respectively.

(iv) For each given error $\varepsilon > 0$, we have $\|X(k) - X\|_F < \epsilon$ after the k-th iteration for any $k \in \mathbb{N}$ that satisfies (4.16).

We can see that A_i, B_i, C_j, D_j affect the convergence rate of Algorithm 4 but E is not. However, the matrix E is required for stopping process. Furthermore, If we take $\epsilon = 0.5 \times 10^{-n}$ in (4.16) and k satisfies

$$k > \frac{\log 0.5 - \log \|X(0) - X\|_F - n}{\log \rho(S)},$$

then the approximated $X(k)$ has an accuracy of n decimal digit.

4.3.3 Optimal convergence factor

The fastest convergence factor of Algorithm 4 is discussed intensively. Recall that the condition number of a matrix A (relative to the spectral norm) is defined by

$$\kappa(A) = \left(\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)} \right)^{\frac{1}{2}}.$$

Assume that Eq. (4.12) holds. The convergence rate of Algorithm 4 is the same as that of the linear iteration (4.9), and thus, it is given by the spectral radius (4.10) of the iteration matrix S. The fastest convergence rate is equivalent to the smallest of $\rho[S]$. Thus, we would like to minimize this spectral radius subject to the condition (4.12). Now, we apply the following lemma:

Lemma 4.2. ([22]) 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 $x_{opt} = \frac{2}{a+b}$.

Then the minimum value of $\rho[S]$ is

$$\tau_{opt} = \frac{2}{\lambda_{\max}(Q^TQ) + \lambda_{\min}(Q^TQ)}. \tag{4.19}$$

Meanwhile, we can notice that spectral radius of the iteration matrix is

$$\begin{aligned} \rho[S] &= \frac{\lambda_{\max}(Q^TQ) - \lambda_{\min}(Q^TQ)}{\lambda_{\max}(Q^TQ) + \lambda_{\min}(Q^TQ)} \\ &= \frac{\kappa^2(Q) - 1}{\kappa^2(Q) + 1}. \end{aligned} \tag{4.20}$$

Thus, we obtain the following theorem:

Theorem 4.3.3. Among the convergence factors τ_3 that meet the criteria of Algorithm 4, the one in Eq. (4.19) attains the fastest asymptotic convergence rate of the algorithm, which is governed by the spectral radius (4.20).

The above theorem tells us that if $\lambda_{\max}(Q^TQ)$ is neighbouring to $\lambda_{\min}(Q^TQ)$, or equivalently, the condition number is close to one then Algorithm 4 has a fast convergence.

4.4 The GIO algorithm for the Sylvester-transpose equation

In this section, we consider an important special case of the matrix equation (1.4), namely, the Sylvester-transpose matrix equation. We apply GIO algorithm for this equation, and investigate the convergence property of the algorithm.

Let $m, n \in \mathbb{N}$. Consider the Sylvester-transpose matrix equation (1.6) where $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times m}$, $F \in \mathbb{R}^{m\times m}$ are given constant matrices and $X \in \mathbb{R}^{n\times m}$ is an unknown matrix to be solved. Suppose that (1.6) has a unique solution, i.e., that following matrix is invertible:

$$Q_s := (I \otimes A) + (B^T \otimes I)K_{nm} \in \mathbb{R}^{mn\times mn}.$$

Algorithm 5: GIO algorithm for a Sylvester-transpose equation

initialization;

Set $A_i' = A_i^T$ and $B_i' = B_i^T$. Choose $\tau \in \mathbb{R}$. Set $k := 0$. Choose initial matrix $X(0)$.

while $k = 0, 1, 2, \ldots, n$ **do**

  $E(k) = F - AX(k) - X^T(k)B.$

  if $\|E(k)\|_F / \|F\|_F < \epsilon$ **then**

    break;

  else

    $X(k+1) = X(k) + \tau_4 \left(A'E(k) + BE^T(k)\right),$

    update k.

  end

end

Corollary 4.3. Consider the matrix equation (1.6) when the matrix Q_s is invertible. Let $\tau_4 \in \mathbb{R}$. Then the given statements hold:

(i) A equivalent condition for which Algorithm 5 is applicable for any initial matrix $X(0)$ is

$$0 < \tau_4 < \frac{2}{\|Q_s\|_2^2}. \tag{4.21}$$

Meanwhile, the spectral radius of the iteration matrix $T = I_{np} - \tau_4 Q_s^T Q_s$ is given by

$$\rho[T] = \max\{|1 - \tau_4 \lambda_{\max}(Q_s^T Q_s)|, |1 - \tau_4 \lambda_{\min}(Q_s^T Q_s)|\}. \tag{4.22}$$

(ii) The spectral radius $\rho[T]$ in (4.22) represents the asymptotic convergence rate of Algorithm 5.

(iii) The fastest asymptotic convergence factor is determined by

$$\tau_{opt} = \frac{2}{\lambda_{\max}(Q_s^T Q_s) + \lambda_{\min}(Q_s^T Q_s)}. \tag{4.23}$$

4.5 Numerical simulations with discussion

In this section, we report some numerical results to illustrate the applicability the effectiveness of Algorithm 4. All simulations have been carried out by MATLAB R2018a, AMD Ryzen7 3700U with Redeon Vega Mobile Gfx @ 2.30 GHz, RAM 12.00 GB PC environment. In Example 4.4, we show that our algorithm is applicable for small-square-matrices in the case $p > q$. We also show that our algorithm is applicable and efficient for large-square-matrices and consider the effect of changing the convergence factor τ in Example 4.5. In Example 4.6, we illustrate the efficiency of Algorithm 4 when coefficients are non-square matrices of different moderate sizes. In Examples 4.7 and 4.8, we do experiments in the cases $p > q$ and $q > p$ with rectangular coefficient matrices. In Example 4.9, we test Algorithm 5 for the Sylvester-transpose equation with square coefficient matrices. In all examples, we compare the proposed algorithm to both the traditional method (Eq. (4.1)) and recent iterative algorithms. The computational time is measured in seconds by MATLAB functions **tic** and **toc**.

Example 4.4. Consider the matrix equation

$$A_1 X B_1 + C_1 X^T D_1 + A_2 X B_2 = F,$$

where

$$A_1 = \begin{bmatrix} -0.123 & 0.002 & 0.780 & -0.563 & 0.009 \\ -0.123 & -0.008 & 0.005 & 0.097 & 0.002 \\ 0.398 & 0.007 & -0.023 & 0.094 & 0.001 \\ -0.009 & 0.478 & -0.994 & 0.001 & 0.005 \\ 0.013 & -0.003 & 0.028 & 0.004 & -0.456 \end{bmatrix}, A_2 = \begin{bmatrix} 0.112 & -0.302 & -0.785 & 0.312 & -0.049 \\ 0.709 & -0.996 & -0.733 & 0.219 & -0.005 \\ 0.261 & -0.005 & -0.003 & 0.114 & -0.111 \\ 0.219 & 0.005 & -0.123 & -0.125 & 0.009 \\ 0.001 & 0.000 & 0.018 & -0.994 & 0.956 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0.667 & -0.209 & 0.346 & -0.675 & -0.099 \\ 0.099 & -0.218 & 0.278 & -0.219 & 0.004 \\ -0.002 & 0.005 & 0.109 & 0.678 & -0.234 \\ 0.056 & -0.005 & -0.006 & 0.195 & 0.009 \\ 0.004 & 0.065 & -0.187 & -0.984 & 0.000 \end{bmatrix}, B_2 = \begin{bmatrix} -0.004 & 0.056 & -0.005 & 0.004 & 0.049 \\ 0.579 & 0.096 & 0.114 & -0.008 & 0.112 \\ -0.113 & -0.119 & 0.284 & -0.003 & 0.014 \\ 0.089 & 0.027 & -0.009 & -0.145 & 0.036 \\ -0.001 & -0.079 & 0.456 & -0.458 & 1.000 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} -0.163 & 0.021 & 0.007 & -0.152 & 0.193 \\ -0.474 & -0.098 & 0.001 & 0.384 & 0.193 \\ -0.085 & 0.109 & 0.093 & -0.017 & 0.173 \\ 0.812 & -0.742 & -0.841 & 0.941 & 0.485 \\ 0.197 & 0.934 & 0.012 & 0.845 & -0.917 \end{bmatrix}, D_1 = \begin{bmatrix} -0.002 & 0.074 & 0.004 & -0.072 & 0.284 \\ 0.056 & 0.037 & 0.485 & 0.188 & 0.485 \\ 0.863 & -0.072 & 0.475 & 0.945 & -0.594 \\ 0.016 & -0.034 & 0.004 & 0.001 & 0.855 \\ 0.854 & 0.003 & 0.927 & -0.923 & 0.567 \end{bmatrix}.$$

In fact, the unique solution is given by

$$X = \begin{bmatrix} 1.000 & 0.010 & -0.224 & -0.111 & 0.908 \\ 0.980 & 0.765 & -0.365 & 0.482 & 0.528 \\ -0.649 & 0.309 & 0.849 & -0.030 & 0.612 \\ -0.495 & 0.008 & 0.862 & -0.001 & -0.004 \\ 0.239 & 0.937 & 0.251 & 0.364 & 0.062 \end{bmatrix}.$$

Let us apply Algorithm 4 to compute the sequence $\{X(k)\}$ of approximated solutions. Take an initial point $X(0) = 0$. The optimal convergence factor can be computed according to Theorem 4.3.3 as follows:

$$\begin{aligned} \tau_{opt} &= \frac{2}{\lambda_{\min}(Q^TQ) + \lambda_{\max}(Q^TQ)} \\ &\approx \frac{2}{8.3389 \times 10^{-6} + 14.5024} \\ &\approx 0.1379. \end{aligned}$$

Table 4.1 shows that the direct method consumes 15 ms to get the exact solution, while GIO algorithm takes 6 ms to perform 10 iterations in order to get an approximated solution with a small relative error. Figure 4.1 and Table 4.1 illustrate that the computational time of GIO algorithm is less than that for GI algorithm with preferable relative error. Figure 4.1 and Table 4.1 also show that GIO algorithm is outperform than LS algorithm.

Example 4.5. Consider the matrix equation

$$\sum_{i=1}^{2} A_i X B_i + \sum_{j=1}^{2} C_j X^T D_j = F$$

where all coefficients are 100×100 tridiagonal matrices given by

$$\begin{aligned} A_1 &= \text{tridiag}(3, 1, -1),\ A_2 = \text{tridiag}(1, 0, 4), \\ B_1 &= \text{tridiag}(-1, 3, 2),\ B_2 = \text{tridiag}(-1, -2, -1), \\ C_1 &= \text{tridiag}(1, 0, -2),\ C_2 = \text{tridiag}(1, -2, 3), \\ D_1 &= \text{tridiag}(0, 2, -4),\ D_2 = \text{tridiag}(1, -1, 1). \end{aligned}$$

Table 4.1: Numerical results for Example 4.4

Algorithm	IT	CT	relative error
Direct	-	0.0150	-
GIO	10	0.0060	0.5088
GI with $\tau = 0.127$	10	0.0078	0.7510
GI with $\tau = 0.009$	10	0.0069	0.9755
LS	10	0.0223	1.0000

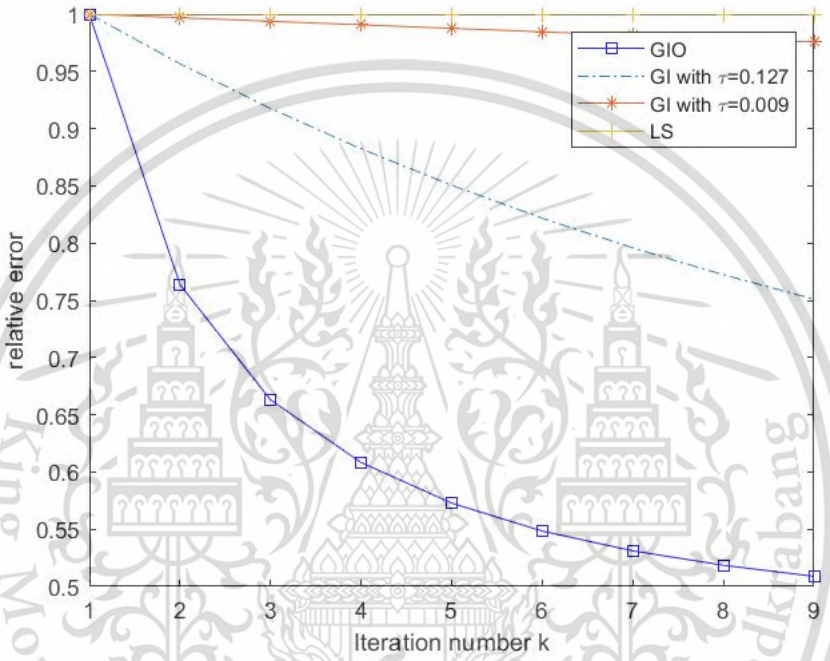


Figure 4.1: Relative error for Example 4.4

In fact, the unique solution is given by $X = \text{tridiag}(0, 1, -1)$. Let us apply Algorithm 4 to compute the sequence $\{X(k)\}$ of approximated solutions. Take an initial point

$$X(0) = 10^{-6} \times \text{tridiag}(-1, -1, -1).$$

The optimal convergence factor can be computed according to Theorem 4.3.3 as follows:

$$\begin{aligned} \tau_{opt} &= \frac{2}{\lambda_{\min}(Q^TQ) + \lambda_{\max}(Q^TQ)} \\ &\approx \frac{2}{4.15 \times 10^{-13} + 7.15 \times 10^4} \\ &\approx 0.000028. \end{aligned}$$

The computing results of changing τ are listed in Figure 4.2 and Table 4.2. Figure 4.2 shows that, for the given initial point, as k increases, for $\tau = \tau_{opt}, \tau = 0.00002, \tau = 0.000015$, and $\tau = 0.00001$, we see that the terms $\|E(k)\|_F/\|F\|_F$ are becoming smaller

and approaching to zero. Moreover, the relative errors $\|E(k)\|_F/\|F\|_F$ for τ_{opt} go faster to 0 than those for another convergence factors. Furthermore, when $\tau = 0.00004$ and $\tau = -0.00001$ which do not satisfy the criteria (4.12), the approximated solutions diverge. Table 4.2 confirms that the computational time of the proposed algorithm is significantly less than the time of the traditional method. Thus, in this case, Algorithm 4 is applicable and effective.

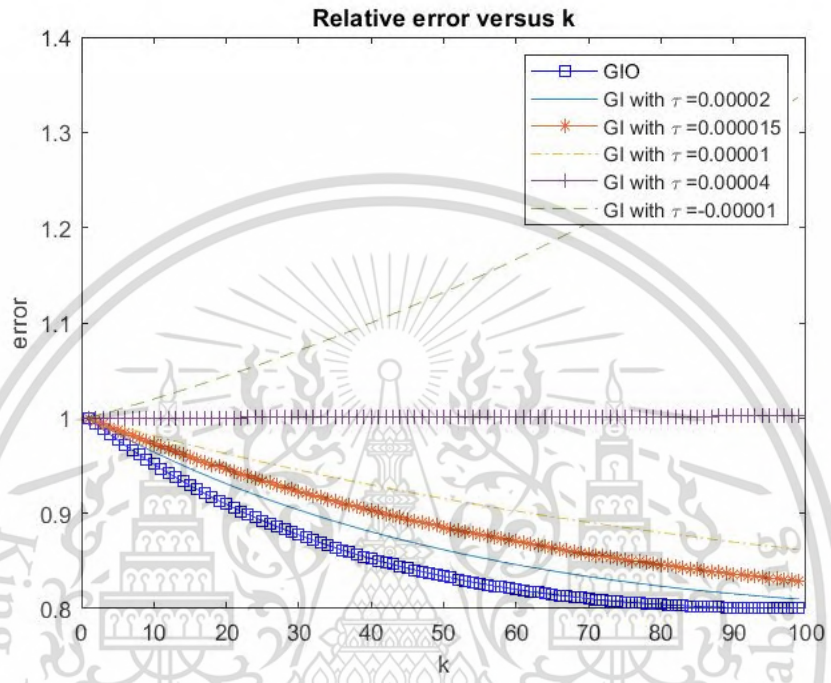


Figure 4.2: Relative errors for Example 4.5

Table 4.2: Numerical results for Example 4.5

Algorithm	IT	CT	relative error
Direct	-	15.2058	-
GIO	100	0.5467	0.8005
GI with $\tau = 0.00002$	100	1.6245	0.8099
GI with $\tau = 0.000015$	100	1.7007	0.8512
GI with $\tau = 0.00001$	100	2.8129	0.8619
GI with $\tau = 0.00004$	100	2.9998	1.0055
GI with $\tau = -0.00001$	100	4.0097	1.3314

Example 4.6. Consider the matrix equation

$$\sum_{i=1}^{3} A_i X B_i + \sum_{j=1}^{3} C_j X^T D_j = F$$

when $A_i \in \mathbb{R}^{40\times 60}$, $B_i \in \mathbb{R}^{20\times 30}$, $C_j \in \mathbb{R}^{40\times 20}$, $D_j \in \mathbb{R}^{60\times 30}$ and $F \in \mathbb{R}^{40\times 30}$ are tridiagonal matrices given by

$$A_1 = \text{tridiag}(1,-1,1), \quad A_2 = \text{tridiag}(2,0,-3), \quad A_3 = \text{tridiag}(-2,-1,-2),$$
$$B_1 = \text{tridiag}(1,-3,0), \quad B_2 = \text{tridiag}(-1,-2,-1), \quad B_3 = \text{tridiag}(0,1,-3),$$
$$C_1 = \text{tridiag}(-3,0,-2), \quad C_2 = \text{tridiag}(-1,-2,3), \quad C_3 = \text{tridiag}(2,-1,2),$$
$$D_1 = \text{tridiag}(0,2,-1), \quad D_2 = \text{tridiag}(1,2,-1), \quad D_3 = \text{tridiag}(0,1,-1).$$

Then the unique solution is $X = \text{tridiag}(0,1,-1) \in \mathbb{R}^{60\times 20}$. We take the initial matrix

$$X(0) \;=\; 10^{-6} \times \text{tridiag}(1,1,1).$$

The computing results of changing τ are listed in Figure 4.3. As k large enough, the terms $\|E(k)\|_F/\|F\|_F$ for $\tau_{opt} \approx 0.000085$ are becoming smaller and go faster to zero than another convergence factors. Moreover, for $\tau = 0.0003$ and $\tau = -0.000001$ which do not satisfy the condition (3.6), we see that the term $\|E(k)\|_F/\|F\|_F$ do not approach zero, so the approximated solutions diverge. From Table 4.3, it is clear that the computational time of GIO algorithm is significantly less than the time of the traditional method and another GI with different convergence factors.

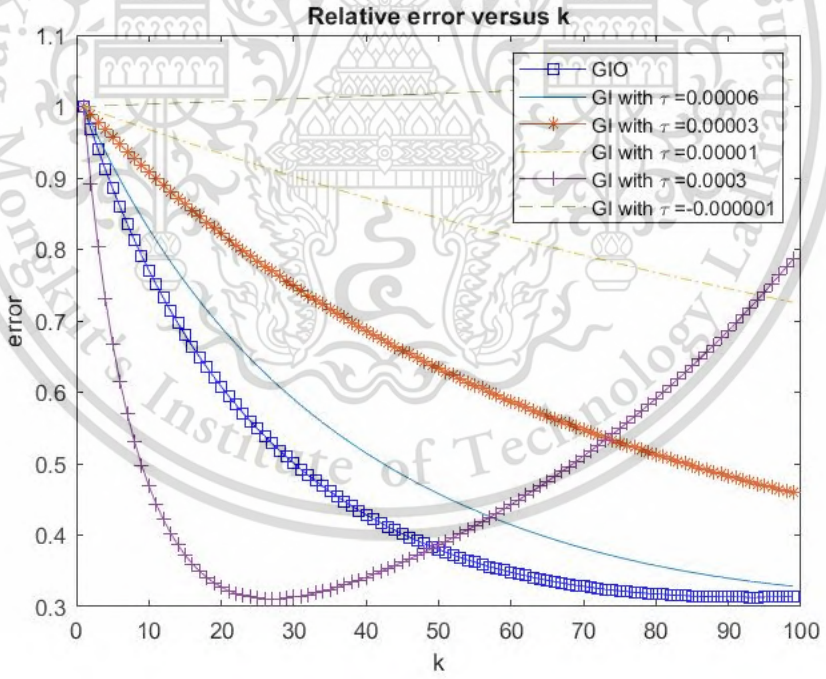


Figure 4.3*:* Relative error for Example 4.6

Example 4.7. Consider the matrix equation

$$\sum_{i=1}^{3} A_i X B_i + \sum_{j=1}^{2} C_j X^T D_j = F$$

Table 4.3*:* Numerical results for Example 4.6

Algorithm	IT	CT	relative error
Direct	-	4.7478	-
GIO	100	0.2350	0.3012
GI with $\tau = 0.00006$	100	1.6954	0.3465
GI with $\tau = 0.00003$	100	2.7760	0.4802
GI with $\tau = 0.00001$	100	3.9008	0.7219
GI with $\tau = 0.0003$	100	4.4435	0.7906
GI with $\tau = -0.00001$	100	4.0978	1.0244

when $A_i \in \mathbb{R}^{40\times 60}$, $B_i \in \mathbb{R}^{20\times 30}$, $C_j \in \mathbb{R}^{40\times 20}$, $D_j \in \mathbb{R}^{60\times 30}$ and $F \in \mathbb{R}^{40\times 30}$ are tridiagonal matrices given by

$$A_1 = \text{tridiag}(-2,2,2), \quad A_2 = \text{tridiag}(3,3,-4), \quad A_3 = \text{tridiag}(3,-1,2),$$
$$B_1 = \text{tridiag}(2,3,5), \quad B_2 = \text{tridiag}(-2,-3,1), \quad B_3 = \text{tridiag}(-1,0,3),$$
$$C_1 = \text{tridiag}(-3,4,-2), \quad C_2 = \text{tridiag}(2,2,-3),$$
$$D_1 = \text{tridiag}(5,3,-1), \quad D_2 = \text{tridiag}(1,2,3).$$

Then the unique solution is $X = \text{tridiag}(1,2,-1) \in \mathbb{R}^{60\times 20}$. We take the initial matrix

$$X(0) = 10^{-6} \times \text{tridiag}(-1,-1,-1) \in \mathbb{R}^{60\times 20}.$$

The numerical results in Figure 4.4 show that the direct method consumes around 11 seconds, while GIO algorithm spends 0.047 seconds (50 iterations) with satisfactory error. Figure 4.4 and Table 4.4 show that the computational time of GIO algorithm is slightly more than that of GI algorithm, as the relative error of GIO is significantly less than that of GI algorithm. Figure 4.4 and Table 4.4 also show that GIO algorithm is outperform than LS algorithm in both computational time and relative error.

Table 4.4*:* Numerical results for Example 4.7

Algorithm	IT	CT	relative error
Direct	-	10.8753	-
GIO	50	0.0470	0.1163
GI with $\tau = 0.00005$	50	0.0411	0.2952
GI with $\tau = 0.00003$	50	0.0367	0.4376
LS	50	0.2664	1.0000

Example 4.8. Consider the matrix equation

$$\sum_{i=1}^{2} A_i X B_i + \sum_{j=1}^{3} C_j X^T D_j = F$$

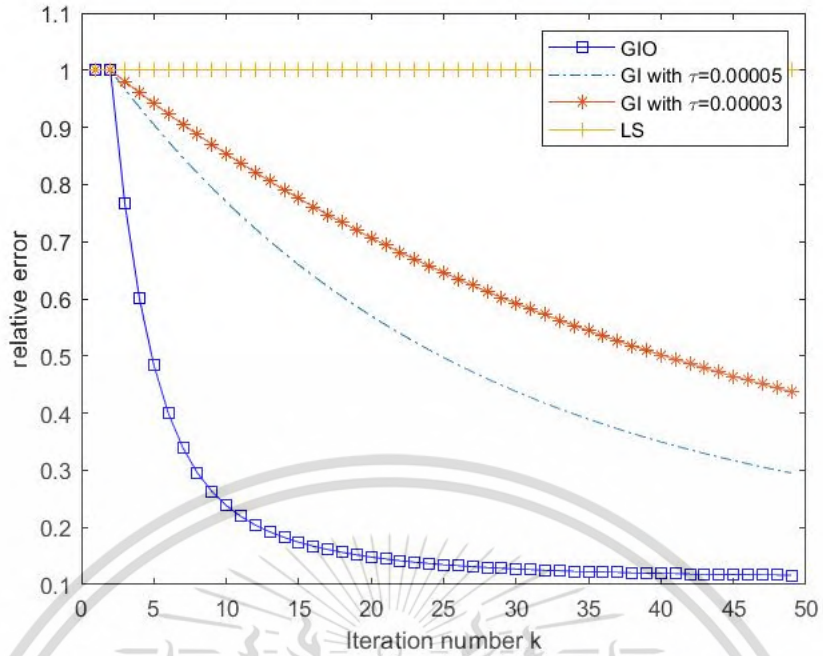


Figure 4.4: Relative error for Example 4.7

where all coefficients are 10×10 tridiagonal matrices given by

$$A_1 = \text{tridiag}(-1, 2, 1),\ A_2 = \text{tridiag}(2, -4, -3),$$
$$B_1 = \text{tridiag}(1, 3, 2),\ B_2 = \text{tridiag}(-2, -3, -1),$$
$$C_1 = \text{tridiag}(2, 3, 1),\ C_2 = \text{tridiag}(1, -3, -1),\ C_3 = \text{tridiag}(5, 3, 4),$$
$$D_1 = \text{tridiag}(-1, 2, -1),\ D_2 = \text{tridiag}(4, 2, 1),\ D_3 = \text{tridiag}(2, 3, 1).$$

In fact, the unique solution is given by $X = \text{tridiag}(1, 1, 1)$. Let us apply Algorithm 4 to compute the sequence $\{X(k)\}$ of approximated solutions using an initial point

$$X(0) = 10^{-6} \times \text{tridiag}(-1, -1, -1).$$

Table 4.5 shows that for small size matrices, the difference between the computational times for the direct method and GIO algorithm (50 iterations) is not much as that for the moderate sizes in Example 4.7. Figure 4.5 and Table 4.5 illustrate that the computational time of GIO algorithm is less than that for GI algorithm with preferable relative error. Figure 4.5 and Table 4.5 also show that GIO algorithm is outperform than LS algorithm.

Example 4.9. The Sylvester-transpose equation $AX + X^T B = F$, where $A, B, F, X \in \mathbb{R}^{10\times 10}$ are given by

$$A = \text{tridiag}(1, -3, 1),\ B = \text{tridiag}(2, 2, 4),\ X = \text{tridiag}(4, 1, 4)$$

is considered by using the initial matrix $X(0) = \text{tridiag}(10^{-6}, 10^{-6}, 10^{-6})$. The objective is to compare the capability of GIO algorithm to GI [28], LS [28], and AGBI [29] algorithms.

Table 4.5: Numerical results for Example 4.8

Algorithm	IT	CT	relative error
Direct	-	0.0266	-
GIO	50	0.0112	0.0370
GI with $\tau = 0.00005$	50	0.0146	0.2813
GI with $\tau = 0.00008$	50	0.0134	0.1526
LS	50	0.0621	0.9942

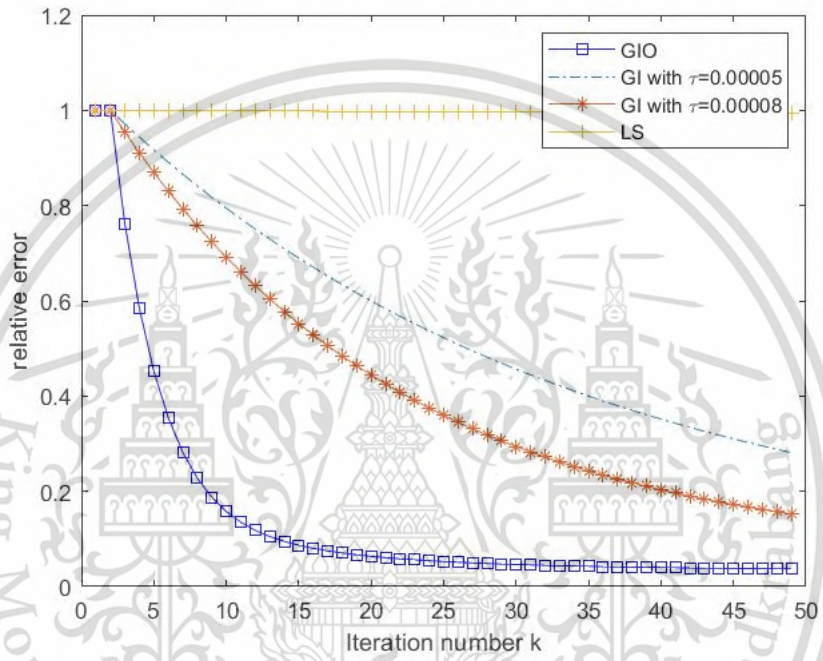


Figure 4.5: Relative error for Example 4.8

We fix the iteration number to be 50 and investigate the relative error $\|E(k)\|_F/\|F\|_F$. Figure 4.6 and Table 4.6 indicate that Algorithm 5 is more efficient than the direct method and another iterative algorithms in spite of a little more computational time.

Appendix

This appendix contains MATLAB-code of Example 4.4

```
tic;
τ = 0.1379;
Xt = X;
XtT = transpose(Xt);
Et = F - (A1*Xt*B1 + A2*Xt*B2 + A3*Xt*B3 + C1*XtT*D1 + C2*XtT*D2 + C3*XT*D3);
EtT = transpose(Et);
```

Table 4.6*:* Numerical results for Example 4.9

Algorithm	IT	CT	relative error
Direct	-	0.0303	-
GIO	50	0.0056	0.8621
GI	50	0.0070	0.8770
LS	50	0.0121	0.9884
AGBI	50	0.0105	0.8838

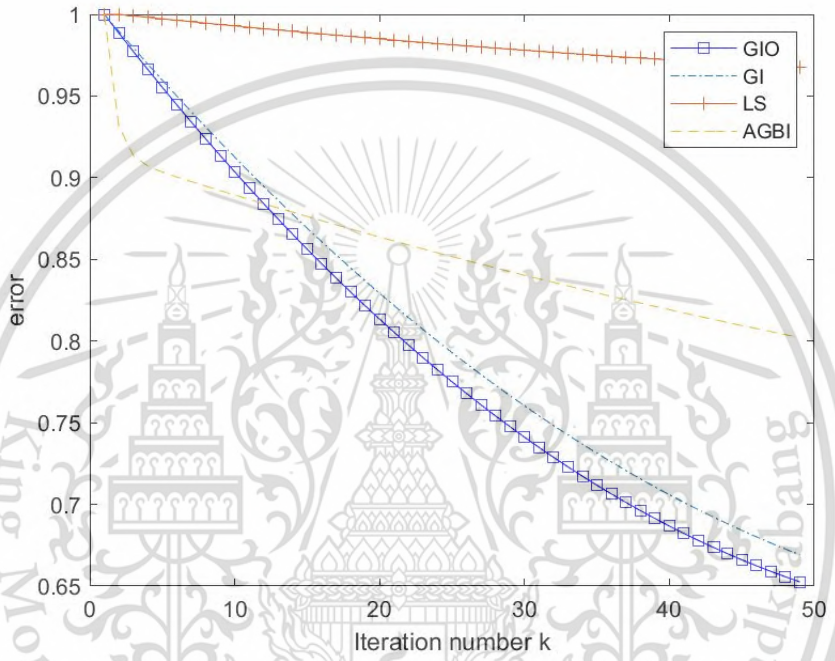


Figure 4.6*:* Relative error for Example 4.9

et(1) = e(1)/norm(F,'fro');
let(1) = le(1);
kt = 2;
while $(kt < 50)$
Et = F - (A1*Xt*B1 + A2*Xt*B2 + A3*Xt*B3 + C1*XtT*D1 + C2*XtT*D2 + C3*XT*D3);
EtT = transpose(Et); Xt = Xt + τ*(A1T*Et*B1T+ A2T*Et*B2T + A3T*Et*B3T + D1*EtT*C1 + D2*EtT*C2 + D3*EtT*C3);
et(kt) = norm(Et,'fro')/norm(F,'fro');
let(kt) = log(norm(Et,'fro'));
kt = kt+1;
end
tt = toc;

Chapter 5

Conclusions and Suggestions

In this chapter, we describe the conclusion of the overall work and give some suggestions regarding for this work.

5.1 Conclusions

In this section, First, we conclude the gradient-based iterative algorithm with an optimal convergence factor for a generalized Sylvester matrix equation (Algorithm 1). Second, we summarize the gradient-based iterative algorithm with an optimal convergence factor for a generalized Sylvester-transpose matrix equation (Algorithm 2). We will simplify by dividing this conclusion into 2 parts.

1 The gradient-based iterative algorithm with an optimal convergence factor is applicable for a generalized Sylvester matrix equation (1.3) under the assumption that the matrix $P := \sum_{i=1}^{p} B_i^T \otimes A_i$ is invertible. The convergence analysis reveals that the sequence of approximated solutions converge to the exact solution for any initial value if and only if the convergent factor τ_1 is satisfied

$$0 < \tau_1 < \frac{2}{\|P\|_2^2}.$$

The convergence rate is the spectral radius of the associated iteration matrix $T = I_{np} - \tau_1 P^T P$ in the form $\rho[T]$. The error estimates $\|X(k) - X\|_F$ compared to the previous step and the first step are presented, respectively

$$\|X(k) - X\|_F \leq \rho[T] \, \|X(k-1) - X\|_F,$$

$$\|X(k) - X\|_F \leq \rho^k[T] \, \|X(0) - X\|_F.$$

Moreover, we obtain the fastest convergence factor

$$\tau_{opt} = \frac{2}{\lambda_{\max}(P^T P) + \lambda_{\min}(P^T P)}$$

that make the associated iteration matrix has the smallest spectral radius. Furthermore, Algorithm 1 is applicable for the discretization of the convection-diffusion and diffusion equations. The numerical experiments illustrate that Algorithm 1 is applicable for any conformable square/rectangular matrices of small/large sizes. Moreover, they reveal that Algorithm 1 performs well comparing to recent iterative algorithms.

2 The gradient-based iterative algorithms with an optimal convergence factor is applicable for a generalized Sylvester-transpose matrix equation (1.4) under the

assumption that the matrix equation $Q = \sum_{i=1}^{p}(B_i^T \otimes A_i) + \sum_{j=1}^{q}(D_j^T \otimes C_j)K_{np}$ is invertible. The convergence analysis of Algorithm 2 that is considered by Banach fixed point theorem tells us that a necessary and sufficient condition for which Algorithm 2 is applicable for any initial matrix is the convergence factor τ_2 is satisfied

$$0 < \tau_2 < \frac{2}{\|Q\|_2^2}.$$

The convergence rate is the spectral radius of the associated iteration matrix $S = I_{np} - \tau Q^T Q$. We also have several estimations

$$\begin{aligned}
\|X(k) - X\|_F &\leq \rho[S]\, \|X(k-1) - X\|_F, \\
\|X(k) - X\|_F &\leq \rho^k[S]\, \|X(0) - X\|_F, \\
\|X(k) - X\|_F &\leq \frac{\rho^k[S]}{1-\rho[S]}\, \|X(1) - X(0)\|_F, \\
\|X(k+1) - X\|_F &\leq \frac{\rho[S]}{1-\rho[S]}\, \|X(k+1) - X(k)\|_F,
\end{aligned}$$

known as the error compared to the previous step and the first step, prior and posterior estimate, respectively. Moreover, we can determine the fastest asymptotic convergence rate that is in the form

$$\tau_{opt} = \frac{2}{\lambda_{\max}(Q^T Q) + \lambda_{\min}(Q^T Q)}.$$

The numerical experiments illustrate that Algorithm 2 can be applied for any matrix problems with conformable coefficient matrices of small/moderate/large sizes and square/non-square sizes. Moreover, Algorithm 2 has more efficient than the direct method and another recent gradient-based iterative algorithms.

5.2 Suggestions

For future research, we may investigate an application of Algorithm 1 to certain discretizations of another ordinary/partial differential equations, e.g.

- heat equation,
- wave equation,
- Laplace equation,
- Poison equation.

Moreover, other researchers can expand the generalized Sylvester or the generalized Sylvester-transpose matrix equation to the system of generalized Sylvester or the generalized Sylvester-transpose matrix equations.

References

[1] Dullerud, G. and Paganini, F. 1994. **A course in robust control theory.** New York: Springer.

[2] Benner, P. 2014. "Factorized Solution of Sylvester Equations with Application in Control." *Theory networks and system.* International Symposium of Mathematics.

[3] Tsui, C.C. 1988. "On robust observer compensator design." *Automatica.* 24(5): 687-692.

[4] Dooren, P.V. 1984. "Reduce order observer: A new algorithm and proof." *System and Control Letters.* 4(5): 243-251.

[5] Wimmer, H.K. 1994. "Consistency of a pair of generalized Sylvester equations." *IEEE Trans. Automat. Control.* 39(5) 1014-1016.

[6] Wu, A.G. Duan, G.R. Xue, Y. 2007. "Kronecker Maps and Sylvester-Polynomial Matrix Equations." *IEEE Trans. Automat. Control.* 52(5): 905-910.

[7] Wu, A.G. Duan, G.R. Zhou, B. 2008. "Solution to Generalized Sylvester Matrix Equations." *IEEE Trans. Automat. Control.* 53(3): 811-815.

[8] Bartels, R.H. Stewart, G.W. 1994. "Solution of the matrix equation $AX + XB = C$." *Circuits Systems and Signal Processing* 13: 820-826.

[9] Benner, P. Quintana, S. 1999. "Solving stable generalized Lyapunov matrix equations with the matrix sign function." *Numer. Algorithms.* 20(1): 75-100.

[10] Jonsson, I. Kagstrom, B. 2002. "Recursive blocked algorithms for solving triangular systems-Part I: One-sided and couple Sylvester-type matrix equations." *ACM T. Math. Software.* 28(4): 392-415.

[11] Jonsson, I. Kagstrom, B. 2002. "Recursive blocked algorithms for solving triangular systems-Part II: Two-sided and generalized Sylvester and Lyapunov matrix equations." *ACM T. Math. Software.* 28(4): 416-435.

[12] Wang, X. Li, Y. Dai, L. 2013. "On the Hermitian and skew-Hermitian splitting iteration methods for the linear matrix equation $AXB = C$." *Comput. Math. Appl.* 65(4): 657-664.

[13] Zhang, H.M. Ding, F. 2014. "A property of the eigenvalues of the symmetric positive definite matrix and the iterative algorithm for couple Sylvester matrix equations." *J. Franklin Inst.* 351(1): 340-357.

[14] Zheng, Q.Q., Ma, Ch.F. 2014. "On normal and skew-Hermitian splitting iteration methods for large sparse continuous Sylvester equation." *J Comp. Appl. Math.* 268: 145-154.

[15] Bai, Z.Z. 2011. "On Hermitian and skew-Hermitian splitting iteration methods for continuous Sylvester equation." *J. Comput. Math.* 29(2): 185-198.

[16] Ding, F. Chen, T. 2005. "Gradient based iterative algorithms for solving a class of matrix equations." *IEEE Trans. Automat. Control.* 50(8): 1216-1221.

[17] Ding, F. Chen, T. 2005. "Iterative least squares solutions of coupled Sylvester matrix equations." *Systems Control Lett.* 54(2): 95-107.

[18] Ding, F. Liu, P.X. Ding, J. 2008. "Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle." *Appl. Math. Comp.* 197(1): 41-50.

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

[20] Fan, W. Gu, C. Tian, Z. 2007. "Jacobi-gradient iterative algorithms for Sylvester matrix equations." 16-20. in 14th Conference of the International Linear Algebra Society. **Linear Algebra Society Topics.** Shanghai:Shanghai University.

[21] Li, S.K. Huang, T.Z. 2011. "A shift-splitting Jacobi-gradient algorithm for Lyapunov matrix equation arising form control theory." *J.Comput. Anal. Appl.* 13(7): 1246-1257.

[22] Xie, Y.J. Ma, C.F. 2016. "The accelerated gradient based iterative algorithm for solving a class of generalized Sylvester-transpose matrix equation." *Appl. Math. Comp* 273: 1257-1269.

[23] Tian, Z. Tian, M. Gu, C. Hao, X. 2017. "An Accelerated Jacobi-gradient Based Iterative Algorithm for Solving Sylvester Matrix Equation." *Filomat.* 31(8): 2381-2390.

[24] Kittisopaporn, A. Chansangiam, P. 2020. "Gradient-descent iterative algorithm for solving a class of linear matrix equations with applications to heat and Poisson equations." *Adv. Differ. Equ.* 2020(1).
Available from:https://doi.org/10.1186/s13662-020-02785-9.

[25] Kittisopaporn, A. Chansangiam, P. Lewkeeratiyukul, W. 2021. "Convergence analysis of gradient-based iterative algorithms for a class of rectangular Sylvester matrix equation based on Banach contraction principle." *Adv. Differ. Equ.* 2021(17).
Available from:https://doi.org/10.1186/s13662-020-03185-9.

[26] Sasaki, N. Chansangiam, P. 2020. "Modified Jacobi-gradient iterative method for generalized Sylvester matrix equation." *Symmetry*, 12(1): 1831. Available from:https://doi.org/10.3390/sym12111831.

[27] Horn, R.A. Johnson, C.R. 1991. **Topics in Matrix Analysis.** New York: Cambridge University Press.

[28] Xie, L. Lui, Y. Yang, H. 2010. "Gradient based and least square based iterative algorithms for matrix equation $AXB + CX^TB = F$." *Appl. Math. Comp.* 217(5): 2191-2199.

[29] Wang, X. Dai, L. Liao, D. 2012. "A modified gradient based algorithm for solving Sylvester equations." *Appl. Math. Comput.* 218: 5620-5628.

[30] E. Kreyszig, *Introductory functional analysis with applications*, University of Windsor, John Wiley & Sons: New York, USA, (1978).

[31] Lim, T.K. 2009. "Nonexpansive matrices with applications to solutions of linear systems by fixed point iterations." *Fixed Point Theory and Applications.* 2010. Available from: https://doi.org/10.1155/2010/821928.

[32] Zhu, M.Z. Zhang G.F. 2015. "A class of iteration methods based on the HSS for Toeplitz system of weakly nonlinear equation." *Comput. Appl. Math.* 290: 433-444.

NO THIS PAGE IN APPENDIX

Author Biography

Name	Miss Nunthakarn Boonruangkan
Date of Birth	25^{th} May 1992
Address	77/2, Pattipad Road, Taladnua Sub-district, Mueang District, Phuket Province, Thailand
Education	(2015) Bachelor of Science(B.Sc.), Faculty of Science, Burapha University, Chonburi, Thailand (2017) Master of Science(M.Sc.), Faculty of Science, Burapha University, Chonburi, Thailand (2020) Doctor of Philosophy in Applied Mathematics King Mongkut's of Institute of Technology Ladkrabang
Scholarship	Ministry of Education, Thailand

Academic Publications

1. Boonruangkan N, Chansangiam P, " Gradient Iterative Method with Optimal Convergent Factor for Solving a Generalized Sylvester Matrix Equation with Applications, to Diffusion Equations", Symmetry, 12(10):1732, 2020.
2. Boonruangkan N, Chansangiam P, " Convergence analysis of a gradient iterative algorithm with optimal convergence factor for a generalized Sylvester-transpose matrix equation", Mathematics, 6(8), 8477-8496, 2021.