Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

The relaxed gradient based iterative algorithm for solving the generalized coupled complex conjugate and transpose Sylvester matrix equations

Yanping Long, Jingjing Cui, Zhengge Huang & Xiaowen Wu

To cite this article: Yanping Long, Jingjing Cui, Zhengge Huang & Xiaowen Wu (2024) The relaxed gradient based iterative algorithm for solving the generalized coupled complex conjugate and transpose Sylvester matrix equations, Automatika, 65:3, 1241-1258, DOI: 10.1080/00051144.2024.2362050

To link to this article: https://doi.org/10.1080/00051144.2024.2362050

AUTOMATIKA
2024, VOL. 65, NO. 3, 1241–1258
https://doi.org/10.1080/00051144.2024.2362050

RESEARCH ARTICLE

OPEN ACCESS Check for updates

The relaxed gradient based iterative algorithm for solving the generalized coupled complex conjugate and transpose Sylvester matrix equations

Yanping Long, Jingjing Cui, Zhengge Huang and Xiaowen Wu

Faculty of Mathematics and Physics, Guangxi Minzu University, Guangxi, People's Republic of China

ABSTRACT
Inspired by the idea of Ma et al. (Journal of the Franklin Institute, 2018), we adopt relaxation technique and introduce relaxation factors into the gradient based iterative (GI) algorithm, and the relaxed based iterative (RGI) algorithm is established to solve the generalized coupled complex conjugate and transpose Sylvester matrix equations. By applying the real representation and straighten operation, we contain the sufficient and necessary condition for convergence of the RGI method. In order to effectively utilize this algorithm, we further derive the optimal convergence parameter and some related conclusions. Moreover, to overcome the high dimension calculation problem, a sufficient condition for convergence with less computational complexity is determined. Finally, numerical examples are reported to demonstrate the availability and superiority of the constructed iterative algorithm.

ARTICLE HISTORY
Received 18 July 2023
Accepted 21 May 2024

KEYWORDS
Generalized coupled complex conjugate and transpose matrix equations; relaxed gradient based iterative algorithm; convergence analysis; optimal convergence parameter

1. Introduction

Solving matrix equations is one of the research focuses of computational mathematics [1–4]. The Sylvester matrix equation is an important type of matrix equations, which has a wide range of applications in control and system theory, pole assignment, model reduction and so further [5–7]. Therefore, finding feasible and effective algorithms for the Sylvester matrix equation has important theoretical significance and practical application value.

In this paper, we aim to find the solution of the generalized coupled complex conjugate and transpose Sylvester matrix equations

$$\sum_{j=1}^{q}(A_{ij}Y_jB_{ij}+C_{ij}\overline{Y_j}D_{ij} + E_{ij}Y_j^TF_{ij}+G_{ij}Y_j^HH_{ij})=M_i,\quad i\in\mathbb{I}[1,p],\tag{1}$$

where $A_{ij},C_{ij}\in\mathbb{C}^{m_i\times r_j}$, $B_{ij},D_{ij}\in\mathbb{C}^{s_j\times n_i}$, $E_{ij},G_{ij}\in\mathbb{C}^{m_i\times s_j}$, $F_{ij},H_{ij}\in\mathbb{C}^{r_j\times n_i}$, $M_i\in\mathbb{C}^{m_i\times n_i}$, $i\in\mathbb{I}[1,p]$, $i\in\mathbb{I}[1,q]$ are the known matrices and $Y_j\in\mathbb{C}^{r_j\times s_j}$ are unknown matrices to be solved. Equation (1) is involved in both science and engineering. Besides, its form includes many other special matrix equations, such as complex conjugate Sylvester matrix equations, complex transpose Sylvester matrix equations and complex conjugate and transpose Sylvester matrix equations [8–10]. Therefore, it is meaningful to research efficient methods for solving Equation (1).

At present, the methods for solving the Sylvester matrix equation mainly include direct methods and iterative methods. However, when solving high-dimensional matrix equations, the direct methods may lead to lengthy computation time. In order to efficiently solve these matrix equations, we prefer to apply iterative methods. In the past few decades, many scholars are devoted to establishing iterative methods to solve various types of Sylvester matrix equations [7,11–14].

From the previous works [15,16], we know that it is difficult to calculate the exact solution of matrix equations, which consumes large computing costs. In the field of systems and control, calculating approximate solutions is sufficient. So iterative solutions have been widely concerned by researchers, and many researchers paid attention to iterative methods and got excellent results. Ding and Chen developed various iterative algorithms to solve $Ax=b$, $AXB=F$ and other Sylvester matrix equations [17–19]. Subsequently, many effective iterative methods were proposed. In [10], the least squares based iterative method has been applied to find the solutions of the Sylvester transpose matrix equation $AXB+CX^TD=F$. Xie and Ding constructed the gradient based iterative (GI) methods for the matrix equations $AXB+CXD=F$ [9]. Wu et al. also investigated the GI method for solving the Sylvester conjugate matrix equation $AXB+C\overline{X}D=F$ [20]. Owing to the availability of the GI algorithm, the GI algorithm has been extended to solve general Sylvester matrix equations by researchers. For instance,

CONTACT Jingjing Cui jingjingcui1990@163.com

Wu et al. proposed the GI algorithm to find the solution of coupled Sylvester-conjugate matrix equations [21]

$$\sum_{i=1}^{l}(A_{ij}X_jB_{ij}+C_{ij}\overline{X}_jD_{ij})=F_i,\quad i\in\mathbb{I}\left[1,s\right], \tag{2}$$

where $A_{ij},C_{ij}\in\mathbb{C}^{m_i\times r_j}$, $B_{ij},D_{ij}\in\mathbb{C}^{t_j\times n_i}$, $F_{ij}\in\mathbb{C}^{m_i\times n_i}$ ($i\in\mathbb{I}[1,s]$, $j\in\mathbb{I}[1,l]$) are the known matrices. Song et al. applied the GI method to the coupled Sylvester-transpose matrix equations [22]

$$\sum_{i=1}^{l}(A_{ij}X_jB_{ij}+C_{ij}X_j^TD_{ij})=F_i,\quad i\in\mathbb{I}\left[1,s\right], \tag{3}$$

where $A_{ij}\in\mathbb{R}^{m_i\times r_j}$, $C_{ij}\in\mathbb{R}^{m_i\times t_j}$, $B_{ij}\in\mathbb{C}^{t_j\times n_i}$, $D_{ij}\in\mathbb{C}^{r_j\times n_i}$, ($i\in\mathbb{I}[1,s]$, $j\in\mathbb{I}[1,l]$) are the known matrices. The above two matrix equations are important types of matrix equations, which are frequently involved in the fields of systems and control. Not only that, they are also the generalized forms of matrix equations in [10,20], respectively. The convergence properties and the optimal convergence parameters of the GI algorithm for Equations (2) and (3) have been investigated.

Subsequently, Beik et al. proposed the GI algorithm for solving the generalized coupled Sylvester-transpose and conjugate matrix equations [23]

$$\begin{aligned}\mathrm{T}_v(X)=\sum_{i=1}^{p}\Bigg(&\sum_{\mu=1}^{s_1}A_{vi\mu}X_iB_{vi\mu}+\sum_{\mu=1}^{s_2}C_{vi\mu}X_i^TD_{vi\mu}\\&+\sum_{\mu=1}^{s_3}M_{vi\mu}\overline{X}_iN_{vi\mu}+\sum_{\mu=1}^{s_4}H_{vi\mu}X_i^HG_{vi\mu}\Bigg)\\=F_v,\end{aligned} \tag{4}$$

where $A_{vi\mu},B_{vi\mu},C_{vi\mu},D_{vi\mu},M_{vi\mu},N_{vi\mu},H_{vi\mu},G_{vi\mu}$, F_v ($v=1,2,\ldots,N$) are known matrices with proper dimensions. The form of the above matrix equations is quite general. When p and s_1,s_2,s_3,s_4, are taken to be some special values, Equation (4) can be transformed into other matrix equations.

Except for the above classical Sylvester matrix equations, the GI algorithm has also been applied to solve periodic matrix equations [24,25]. Li et al. established the GI method for the forward periodic Sylvester matrix equations and backward forward periodic Sylvester matrix equations [25]

$$A_iX_iB_i+C_iX_{i+1}D_i=F_i,\quad i\in\mathbb{I}\left[1,\gamma\right], \tag{5}$$

and

$$A_iX_{i+1}B_i+C_iX_iD_i=F_i,\quad i\in\mathbb{I}\left[1,\gamma\right], \tag{6}$$

where $A_i,B_i,C_i,D_i,F_i\in\mathbb{R}^{n\times n}$ are the known matrices. In theory, Li et al. proposed the sufficient and necessary conditions for the convergence of the GI algorithm. Numerical experiments have also showed the effectiveness of the GI algorithm.

Although the theory of the GI algorithm has been systematically proposed by researchers, this algorithm still has some drawbacks. In [26], Fan et al. pointed out that the GI algorithm costs large computation time and storage space when encountering ill-posed problems. In order to further optimize the convergence performance of the GI algorithm, the relaxed gradient based iterative (RGI) algorithm has been proposed by introducing the relaxed factor to adjust the weight of the iteration sequences. Niu et al. developed the RGI algorithm to solve the Sylvester matrix equations [27]. Numerical experiments have shown that relaxation techniques can effectively reduce computation time and storage space, and improve the convergence rate of the GI algorithm.

Due to the superiority of the RGI algorithm, many scholars have extended this algorithm to solve more general matrix equations. Recently, in [28,29], Huang et al. applied the RGI algorithm to solve coupled Sylvester-conjugate matrix equation (2) and coupled Sylvester-transpose matrix equation (3). And the experiments results illustrate that the convergence rate of the RGI algorithm is faster than the GI one. Then Wang et al. consider the solution of the complex conjugate and transpose matrix equations

$$A_1XB_1+A_2\overline{X}B_2+A_3X^TB_3+A_4X^HB_4=E, \tag{7}$$

where A_i, B_i, $E\in\mathbb{C}^{n\times n}$ ($i\in\mathbb{I}[1,4]$) are the known matrices. By introducing relaxation factors and applying the hierarchical identification principle [30], Wang et al. presented the RGI method to solve Equation (7). However, Wang et al. didn't discuss the generalized form of Equation (7). Based on the ideas of [30], we extend the RGI algorithm to the generalized coupled complex conjugate and transpose Sylvester matrix equations.

Inspired by the idea of [28], we construct the RGI algorithm for solving Equation (1). Its form can be specific written as

$$\begin{cases}\sum_{j=1}^{q}A_{1j}Y_jB_{1j}+\sum_{j=1}^{q}C_{1j}\overline{Y_j}D_{1j}\\ \quad+\sum_{j=1}^{q}E_{1j}Y_j^TF_{1j}+\sum_{j=1}^{q}G_{1j}Y_j^HH_{1j}=M_1,\\ \sum_{j=1}^{q}A_{2j}Y_jB_{2j}+\sum_{j=1}^{q}C_{2j}\overline{Y_j}D_{2j}\\ \quad+\sum_{j=1}^{q}E_{2j}Y_j^TF_{2j}+\sum_{j=1}^{q}G_{2j}Y_j^HH_{2j}=M_2,\\ \quad\vdots\\ \sum_{j=1}^{q}A_{pj}Y_jB_{pj}+\sum_{j=1}^{q}C_{pj}\overline{Y_j}D_{pj}\\ \quad+\sum_{j=1}^{q}E_{pj}Y_j^TF_{pj}+\sum_{j=1}^{q}G_{pj}Y_j^HH_{p}=M_p.\end{cases} \tag{8}$$

The form of the above matrix equations is quite general, which contains several classic Sylverster matrix equations. Especially, Equations (2)–(3) are special cases of Equation (1). If $i = j = p = q = 1$, Equation (1) will reduce to Equation (7). Therefore, finding faster algorithms to solve Equation (1) is of great significance.

To accelerate convergence rate of the GI algorithm for Equation (1), we combine relaxation technology with hierarchical identification principle, and we derive the relaxed gradient based iterative (RGI) algorithm to solve Equation (1). This principle regards the unknown matrix as the system parameter matrix to be solved, then it builds a recursive formula to approach the unknown solution [27,28,30,31]. Furthermore, we can effectively control the weight of the iteration sequence by introducing relaxation factors. In theory, we exploit the real representation and the straightening operator to prove the convergence properties of the constructed algorithm. Meanwhile, the sufficient and necessary condition for convergence is presented. Finally, numerical experiments further demonstrate the effectiveness and superiority of the RGI algorithm. The main motivation and contribution of this paper are summarized as follows:

- In order to accelerate the convergence rate of the GI algorithm [23], we combine the GI algorithm with relaxation technique. By introducing l relaxation factors, we construct the RGI algorithm for Equation (1). Due to that Equation (1) extremely is general, the algorithm constructed in this paper is also more general. It is meaningful to promote the development of the field of solving matrix equations.
- To optimize convergence theory, we utilize real representation and straighten operation as tool, and present the sufficient and necessary condition for convergence of the RGI method. To overcome high-dimensional computing problems, the sufficient condition for convergence and some related results are proposed. Besides, we use numerical experiments to fully demonstrate the effectiveness and superiority of the RGI algorithm.

The remainder of this paper is structured as follows. In Section 2, we list several useful notations and definitions. Moreover, we construct the relaxed gradient based iterative (RGI) algorithm to find the iterative solution of Equation (1) in Section 3. In Section 4, we deduce the convergence properties of the proposed method, including the sufficient and necessary condition for convergence, the optimal convergence factor and the related corollary. In Section 5, two numerical experiments are reported to validate the superior of convergence for the new algorithm. In the end, Section 6 proposes the some conclusions.

2. Preliminaries

For the sake of convenience, we provide several main notations and lemmas which are used throughout this paper. The set of $m \times n$ complex matrix is denoted by $\mathbb{C}^{m\times n}$. For $A \in \mathbb{C}^{m\times n}$, there are some related notations as follows:

- $\overline{A}$ indicates the conjugate of the matrix A;
- A^T represents the transpose of the matrix A;
- A^H stands for the conjugate transpose of the matrix A;
- $\sigma_{\max}(A)$ stands for the maximal singular of the matrix A;
- $\sigma_{\min}(A)$ stands for the minimal singular of the matrix A;
- $cond(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$ is defined as the condition number of A;
- $\lambda_{\max}(A)$ represents the maximal eigenvalue of the matrix A;
- $\lambda_{\min}(A)$ indicates the minimal eigenvalue of the matrix A;
- $\| A \|_2$ is defined as the the spectral norm of the matrix A.
- $\| A \|$ indicates the Frobenius norm of the matrix A.
- $\rho(A)$ represents the spectral radius of the matrix A;

Then, some significant definitions and lemmas are listed below.

Definition 2.1 ([28]): Let $A \in \mathbb{C}^{m\times n}$, then A can be uniquely expressed as $A = A_1 + iA_2$ with $A_1, A_2 \in \mathbb{R}^{m\times n}$. A^{∇} denotes the real representation of a complex matrix A

$$A^{\nabla} = \begin{pmatrix} A_1 & -A_2 \\ A_2 & A_1 \end{pmatrix} \in \mathbb{R}^{2m\times 2n}. \tag{9}$$

Definition 2.2 ([32]): For two matrices $A = (a_{ij}) \in \mathbb{C}^{m\times n}$, $B = (b_{ij}) \in \mathbb{C}^{k\times l}$, the Kronecker product is defined as

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}. \tag{10}$$

Definition 2.3 ([28]): Let e_{in} denote an n-dimensional column vector which has 1 in the ith position and 0's elsewhere. The vec-permutation matrix P_{mn} can be defined as

$$P_{mn} := \begin{pmatrix} I_m \otimes e_{1n}^T \\ I_m \otimes e_{2n}^T \\ \vdots \\ I_m \otimes e_{nn}^T \end{pmatrix}. \tag{11}$$

If $X, A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{p\times q}$, we have

$$P_{mn}P_{mn} = I_{mn}, \quad P_{mn}^T = P_{mn}^{-1} = P_{mn}, \tag{12}$$

and

$$B \otimes A = P_{mp}^T (A \otimes B) P_{nq},$$
$$(A \otimes B) P_{nq} = P_{mp}(B \otimes A). \tag{13}$$

Next, we review several lemmas which are used to prove the convergence property.

Lemma 2.1 ([33]): *If $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{s\times t}$, $X \in \mathbb{C}^{n\times s}$, then*

$$vec(ABC) = (C^T \otimes A) vec(B), \tag{14}$$
$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD). \tag{15}$$

Lemma 2.2 ([29]): *For two matrices A and B, it has*

$$\|A \otimes B\|_2 = \|A\|_2 \|B\|_2. \tag{16}$$

Lemma 2.3 ([28]): *For $A \in \mathbb{C}^{m\times r}$, $B \in \mathbb{C}^{s\times n}$, $F \in \mathbb{C}^{m\times n}$, if the matrix equation $AXB = F$ has unique solution, then the iterative sequences $\{X(k)\}$ converges to the exact solution X^* for any initial matrix $X(0)$ by the following algorithm*

$$X(k+1) = X(k) + \mu A^H (F - AX(k)B) B^H, \tag{17}$$

and the algorithm is convergent if and only if

$$0 < \mu < \frac{2}{\|A\|_2^2 \|B\|_2^2}. \tag{18}$$

Meanwhile, the optimal convergence factor is

$$\mu_0 = \frac{2}{\lambda_{\max}(A^H A)\lambda_{\max}(B^H B) + \lambda_{\min}(A^H A)\lambda_{\min}(B^H B)}. \tag{19}$$

Proof: Define error matrix

$$\widetilde{X}(k) = X(k) - X^*.$$

According to the expression (17), it has

$$\widetilde{X}(k+1) = \widetilde{X}(k) - \mu A^H A \widetilde{X}(k) B B^H.$$

Let $Z(k) = A\widetilde{X}(k)B$, utilizing the properties of matrix Frobenius norm, Lemmas 2.1 and 2.2, it follows that

$$\begin{aligned}
&\|\widetilde{X}(k+1)\|^2 \\
&\quad = \|\widetilde{X}(k)\|^2 - \mu tr(B^H \widetilde{X}^H(k) A^H Z(k)) \\
&\qquad - \mu tr(Z^H(k) A \widetilde{X}(k) B) + \mu^2 \|A^H Z(k) B^H\|^2 \\
&\quad \le \|\widetilde{X}(k)\|^2 - \mu tr(Z^H(k)Z(k)) - \mu tr(Z^H(k)Z(k)) \\
&\qquad + \mu^2 \|(\overline{B} \otimes A^H) vec(Z(k))\|^2 \\
&\quad = \|\widetilde{X}(k)\|^2 - \mu(2 - \mu\|\overline{B} \otimes A^H\|_2^2)\|Z(k)\|^2 \\
&\quad = \|\widetilde{X}(k)\|^2 - \mu(2 - \mu\|A\|_2^2\|B\|_2^2)\|Z(k)\|^2.
\end{aligned}$$

Repeatedly applying the relationship of the above expression leads to

$$\begin{aligned}
&\|\widetilde{X}(k+1)\|^2 \\
&\quad \le \|\widetilde{X}(k-1)\|^2 - \mu(2 - \mu\|A\|_2^2\|B\|_2^2)(\|Z(k)\|^2 \\
&\qquad + \|Z(k-1)\|^2) \\
&\quad \le \|\widetilde{X}(0)\|^2 - \mu(2 - \mu\|A\|_2^2\|B\|_2^2)\left(\sum_{i=0}^{k} \|Z(i)\|^2\right).
\end{aligned}$$

If the convergence parameter μ is selected to satisfy

$$0 < \mu < \frac{2}{\|A\|_2^2\|B\|_2^2},$$

the following inequality holds

$$0 < \mu(2 - \mu\|A\|_2^2\|B\|_2^2)\sum_{i=0}^{\infty}\|Z(i)\|^2 \le \|\widetilde{X}(0)\|^2.$$

This means that $\lim_{i\to\infty}\|Z(i)\|^2 = 0$. Due to that the matrix equation $AXB = F$ has unique solution, then it has $\lim_{k\to\infty}\widetilde{X}(k) = 0$. The proof of Equation (18) is completed. ■

Taking the vec-operator of both sides of the expression (17) and applying Lemma 2.2, it can get

$$vec(\widetilde{X}(k+1)) = (I - \mu BB^H \otimes A^H A) vec(\widetilde{X}(k)).$$

The above equation implies that $I - \mu BB^H \otimes A^H A$ is the iterative matrix of the algorithm. Thus, the optimal convergence parameter satisfies the following equation

$$\begin{aligned}
&\min \max\{|1 - \mu\lambda_1(BB^H \otimes A^H A)|, \ldots, \\
&\quad 1 - \mu\lambda_{sm}(BB^H \otimes A^H A)|\} \\
&\quad = \min \max\{|1 - \mu\lambda_{\max}(BB^H \otimes A^H A)|, \\
&\quad |1 - \mu\lambda_{\min}(BB^H \otimes A^H A)|\},
\end{aligned}$$

which means that $|1 - \mu\lambda_{\max}(BB^H \otimes A^H A)| = |1 - \mu\lambda_{\min}(BB^H \otimes A^H A)|$ has a non-trivial solution. By simple deductions, Expression (19) can be obtained.

Lemma 2.4 ([28]): *The properties of real representation $(.)^\nabla$ are as follows:*

For two complex matrices $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{n\times r}$, then

$$(AB)^\nabla = A^\nabla B^\nabla, \quad (A^T)^\nabla = E_n (A^\nabla)^T E_m,$$
$$(A^H)^\nabla = (A^\nabla)^T, \quad (\overline{A})^\nabla = E_m A^\nabla E_n. \tag{20}$$

Here, unitary matrices E_n is defined as

$$E_n = \begin{pmatrix} 0 & I_n \\ I_n & 0 \end{pmatrix}. \tag{21}$$

Furthermore, based on the definition of matrix Frobenius norm and real representation, then

$$\|A^\nabla\|^2 = 2\|A\|^2, \tag{22}$$

$$\|A^{\nabla}\|_2 = \|A\|_2. \tag{23}$$

Lemma 2.5 ([29]): *If $m_i, i \in \mathbb{I}[1,n]$ are any given positive number, denote the maximum and minimum values of m_i as $m_{\max} = \max_{1\le i\le n} m_i$ and $m_{\min} = \min_{1\le i\le n} m_i$, respectively. It has*

$$\min_{0<\mu<\frac{2}{m_{\max}}} \max_{1\le i\le n} |1-\mu m_i| = \frac{m_{\max}-m_{\min}}{m_{\max}+m_{\min}}, \tag{24}$$

then, the optimal convergence parameter μ is selected as

$$\mu_{\text{opt}} = \frac{2}{m_{\max}+m_{\min}}.$$

Proof: Build function $y = \max_{1\le i\le n} |1-\mu m_i|$, and then Equation (24) has been obtained by drawing graph in [29]. Besides, $|1-\mu m_i| < 1$ if and only if $0 < u < \frac{2}{m_{\max}}$. The optimal convergence factor μ satisfies

$$\begin{aligned}&\min\max\{|1-\mu m_1|, |1-\mu m_2|, \ldots, |1-\mu m_n|\}\\ &\quad = \min\max\{|1-\mu m_1|, |1-\mu m_n|\}.\end{aligned}$$

The above equation indicates that $|1-\mu m| = |1-\mu m_n|$, that is, $-1+\mu m_1 = 1-\mu m_n$. By simple calculations, it has

$$\mu_{\text{opt}} = \frac{2}{m_1+m_n} = \frac{2}{m_{\max}+m_{\min}}.$$

Thus, the proof is completed. ■

3. The relaxed gradient-based iterative algorithm

In this section, we mainly propose the relaxed gradient based iterative (RGI) algorithm to solve the generalized coupled complex conjugate and transpose matrix equation. The main idea of this algorithm is to use the hierarchical identification principle to divide Equation (1) into several subsystems. The unknown matrixes Y_j are regarded as the identified parameters matrices. Meanwhile, we construct intermediate matrices and adopt an average strategy. Then, the relaxation factors ω_l, $l \in \mathbb{I}[1,q]$ are introduced, which are utilized to adjust the weights of matrix schemes. The construction process of the RGI algorithm is as follows.

Firstly, define the following intermediate matrices, $i \in \mathbb{I}[1,p]$, $l \in \mathbb{I}[1,q]$,

$$\begin{aligned}\Pi_{il} = M_i - \sum_{j=1}^{q}(&A_{ij}Y_jB_{ij} + C_{ij}\overline{Y_j}D_{ij} + E_{ij}Y_j^TF_{ij}\\ &+ G_{ij}Y_j^HH_{ij}) + A_{il}Y_lB_{il},\end{aligned} \tag{25}$$

$$\Upsilon_{il} = \overline{\begin{aligned}M_i - \sum_{j=1}^{q}(&A_{ij}Y_jB_{ij} + C_{ij}\overline{Y_j}D_{ij} + E_{ij}Y_j^TF_{ij}\\ &+G_{ij}Y_j^HH_{ij}) + C_{il}\bar{Y}_lD_{il},\end{aligned}} \tag{26}$$

$$\begin{aligned}\Phi_{il} = (M_i - \sum_{j=1}^{q}(&A_{ij}Y_jB_{ij} + C_{ij}\overline{Y_j}D_{ij} + E_{ij}Y_j^TF_{ij}\\ &+ G_{ij}Y_j^HH_{ij}) + E_{il}Y_l^TF_{il})^T,\end{aligned} \tag{27}$$

$$\begin{aligned}\Psi_{il} = (M_i - \sum_{j=1}^{q}(&A_{ij}Y_jB_{ij} + C_{ij}\overline{Y_j}D_{ij} + E_{ij}Y_j^TF_{ij}\\ &+ G_{ij}Y_j^HH_{ij}) + G_{il}Y_l^HH_{il})^H.\end{aligned} \tag{28}$$

From the expression of Equation (1), some subsystems are given below, $i \in \mathbb{I}[1,p]$, $l \in \mathbb{I}[1,q]$,

$$A_{il}Y_lB_{il} = \Pi_{il}, \tag{29}$$

$$\overline{C_{il}}Y_l\overline{D_{il}} = \Upsilon_{il}, \tag{30}$$

$$F_{il}^TY_lE_{il}^T = \Phi_{il}, \tag{31}$$

$$H_{il}^HY_lG_{il}^H = \Psi_{il}. \tag{32}$$

According to the above fictitious subsystems and Lemma 2.3, we can put forward the iterative schemes as follows, $i \in \mathbb{I}[1,p]$, $l \in \mathbb{I}[1,q]$,

$$Y_l^{1,i}(k+1) = Y_l^{1,i}(k) + \mu A_{il}^H[\Pi_{il} - A_{il}Y_l^{1,i}(k)B_{il}]B_{il}^H, \tag{33}$$

$$Y_l^{2,i}(k+1) = Y_l^{2,i}(k) + \mu C_{il}^T[\Upsilon_{il} - \overline{C_{il}}Y_l^{2,i}(k)\overline{D_{il}}]D_{il}^T, \tag{34}$$

$$Y_l^{3,i}(k+1) = Y_l^{3,i}(k) + \mu\overline{F_{il}}[\Phi_{il} - F_{il}^TY_l^{3,i}(k)E_{il}^T]\overline{E_{il}}, \tag{35}$$

$$Y_l^{4,i}(k+1) = Y_l^{4,i}(k) + \mu H_{il}[\Psi_{il} - H_{il}^HY_l^{4,i}(k)G_{il}^H]G_{il}. \tag{36}$$

For the sake of convenience, we provide the following notations, $s \in \mathbb{I}[1,4]$,

$$\begin{aligned}\Gamma_{ij}^{s,i}(k) = \sum_{j=1}^{q}(&A_{ij}Y_j^{s,i}(k)B_{ij} + C_{ij}\overline{Y_j^{s,i}(k)}D_{ij}\\ &+ E_{ij}Y_j^{s,i}(k)^TF_{ij} + G_{ij}Y_j^{s,i}(k)^HH_{ij}).\end{aligned} \tag{37}$$

Combining Equations (25)–(28) with Equations (33)–(36) and utilizing the hierarchical identification principle, the recursive systems are established. Due to that the unknown matrices Y_j are included in the expressions, we replace Y_j in (25)–(28) with $Y_j^{1,i}(k)$, $Y_j^{2,i}(k)$, $Y_j^{3,i}(k)$, $Y_j^{4,i}(k)$, respectively. Therefore, the following expressions are given, $i \in \mathbb{I}[1,p]$, $l \in \mathbb{I}[1,q]$,

$$X_l^{1,i}(k+1) = X_l^{1,i}(k) + \mu A_{il}^H\left[M_i - \sum_{j=1}^{q}\Gamma_{ij}^{1,i}(k)\right]B_{il}^H, \tag{38}$$

$$X_l^{2,i}(k+1) = X_l^{2,i}(k) + \mu C_{il}^T\overline{\left[M_i - \sum_{j=1}^{q}\Gamma_{ij}^{2,i}(k)\right]}D_{il}^T, \tag{39}$$

$$X_l^{3,i}(k+1) = X_l^{3,i}(k) + \mu \overline{F_{il}} \left[M_i - \sum_{j=1}^{q} \Gamma_{ij}^{3,i}(k) \right]^T \overline{E_{il}}, \quad (40)$$

$$X_l^{4,i}(k+1) = X_l^{4,i}(k) + \mu H_{il} \left[M_i - \sum_{j=1}^{q} \Gamma_{ij}^{4,i}(k) \right]^H G_{il}. \quad (41)$$

Then, by taking the average value of $Y_j^{1,i}(k)$, $Y_j^{2,i}(k)$, $Y_j^{3,i}(k)$ and $Y_j^{4,i}(k)$, for $i \in \mathbb{I}[1,p]$ we have

$$\begin{aligned} Y_l'(k+1) &= Y_l'(k) + \frac{\mu}{p} \sum_{i=1}^{p} A_{il}^H \\ &\quad \times [M_i - \sum_{j=1}^{q} (A_{ij} Y_j'(k) B_{ij} + C_{ij} \overline{Y_j'(k)} D_{ij} \\ &\quad + E_{ij} Y_j'(k)^T F_{ij} + G_{ij} Y_j'(k)^H H_{ij})] B_{il}^H, \\ Y_l''(k+1) &= Y_l''(k) + \frac{\mu}{p} \sum_{i=1}^{p} C_{il}^T \\ &\quad \overline{\begin{matrix} [M_i - \sum_{j=1}^{q} (A_{ij} Y_j''(k) B_{ij} \\ +C_{ij} \overline{Y_j''(k)} D_{ij} + E_{ij} Y_j''(k)^T F_{ij} \\ +G_{ij} Y_j''(k)^H H_{ij})] \end{matrix}} D_{il}^T, \\ \check{Y}_l(k+1) &= \check{Y}_l(k) + \frac{\mu}{p} \sum_{i=1}^{p} \overline{F_{il}} \\ &\quad \times [M_i - \sum_{j=1}^{q} (A_{ij} \check{Y}_j(k) B_{ij} + C_{ij} \overline{\check{Y}_j(k)} D_{ij} \\ &\quad + E_{ij} \check{Y}_j(k)^T F_{ij} + G_{ij} \check{Y}_j(k)^H H_{ij})]^T \overline{E_{il}}, \\ \acute{Y}_l(k+1) &= \acute{Y}_l(k) + \frac{\mu}{p} \sum_{i=1}^{p} H_{il} \\ &\quad \times [M_i - \sum_{j=1}^{q} (A_{ij} \acute{Y}_j(k) B_{ij} + C_{ij} \overline{\acute{Y}_j(k)} D_{ij} \\ &\quad + E_{ij} \acute{Y}_j(k)^T F_{ij} + G_{ij} \acute{Y}_j(k)^H H_{ij})]^H G_{il}. \end{aligned}$$

Inspire by the idea of the RGI method in [28], we introduce the relaxation factors ω_l ($l \in \mathbb{I}[1,q]$) into the above recursive systems. Based on the previous analysis process, the relaxed gradient-based iterative (RGI) algorithm for Equation (1) is presented as follows.

In the RGI algorithm μ indicates the convergence parameter. The relaxation factors ω_l ($l \in \mathbb{I}[1,q]$) are used to control the weight of iterative sequences, and it can effectively improve the convergence rate of the GI method. In particular, if the relaxed factors are selected as $\omega_l = \frac{1}{2}$ for all l, Algorithm 1 will reduce to the GI algorithm [23]. Besides, Algorithm 1 with $\omega_l = \frac{1}{2}$ and $p = q = i = j = 1$ will change into the iterative method in [30]. Compared with the RGI algorithm

Algorithm 1 The relaxed gradient-based iterative algorithm

Step 1: Given matrices $A_{ij}, C_{ij} \in \mathbb{C}^{m_i \times r_j}$, $B_{ij}, D_{ij} \in \mathbb{C}^{s_j \times n_i}$, $E_{ij}, G_{ij} \in \mathbb{C}^{m_i \times s_j}$, $F_{ij}, H_{ij} \in \mathbb{C}^{r_j \times n_i}$, $M_i \in \mathbb{C}^{m_i \times n_i}$, $i \in \mathbb{I}[1,p]$, $j \in \mathbb{I}[1,q]$, and given any positive constants number ε and μ. Let $0 < \omega_l < 1$, $l \in \mathbb{I}[1,q]$. Select the initial matrices $Y_l^{(1)}(0) \sim Y_l^{(4)}(0)$, $l \in \mathbb{I}[1,q]$. We take $k = 0$;

Step 2: If $\delta_{k,i} = \|M_i - \sum_{j=1}^{q}(A_{ij} Y_j(k) B_{ij} + C_{ij} \overline{Y_j}(k) D_{ij} + E_{ij} Y_j(k)^T F_{ij} + G_{ij} Y_j(k)^H H_{ij})\| / \|M_i\| < \varepsilon$, stop; otherwise, go to Step 3;

Step 3: Update the sequence

$$\begin{aligned} Y_l^{(1)}(k+1) &= Y_l^{(1)}(k) + \frac{1}{2}\mu\omega_l \sum_{i=1}^{p} A_{il}^H \\ &\quad \times \left[M_i - \sum_{j=1}^{q} (A_{ij} Y_j(k) B_{ij} + C_{ij} \overline{Y_j}(k) D_{ij} + E_{ij} Y_j(k)^T F_{ij} \right. \\ &\quad \left. + G_{ij} Y_j(k)^H H_{ij}) \right] B_{il}^H, \\ Y_l^{(2)}(k+1) &= Y_l^{(2)}(k) + \frac{1}{2}\mu\omega_l \sum_{i=1}^{p} C_{il}^T \\ &\quad \times \left[\overline{\begin{matrix} M_i - \sum_{j=1}^{q} (A_{ij} Y_j(k) B_{ij} + C_{ij} \overline{Y_j}(k) D_{ij} \\ + E_{ij} Y_j(k)^T F_{ij} + G_{ij} Y_j(k)^H H_{ij} \end{matrix}} \right] D_{il}^T, \\ Y_l^{(3)}(k+1) &= Y_l^{(3)}(k) + \frac{1}{2}\mu(1-\omega_l) \sum_{i=1}^{p} \overline{F_{il}} \\ &\quad \times \left[M_i - \sum_{j=1}^{q} (A_{ij} Y_j(k) B_{ij} + C_{ij} \overline{Y_j}(k) D_{ij} + E_{ij} Y_j(k)^T F_{ij} \right. \\ &\quad \left. + G_{ij} Y_j(k)^H H_{ij} \right]^T \overline{E_{il}}, \\ Y_l^{(4)}(k+1) &= Y_l^{(4)}(k) + \frac{1}{2}\mu(1-\omega_l) \sum_{i=1}^{p} H_{il} \\ &\quad \times \left[M_i - \sum_{j=1}^{q} (A_{ij} Y_j(k) B_{ij} + C_{ij} \overline{Y_j}(k) D_{ij} + E_{ij} Y_j(k)^T F_{ij} \right. \\ &\quad \left. + G_{ij} Y_j(k)^H H_{ij} \right]^H G_{il}. \end{aligned}$$

Calculate

$$\begin{aligned} Y_l(k+1) &= \frac{1}{2}(1-\omega_l) Y_l^{(1)}(k+1) \\ &\quad + \frac{1}{2}(1-\omega_l) Y_l^{(2)}(k+1) \\ &\quad + \frac{1}{2}\omega_l Y_l^{(3)}(k+1) + \frac{1}{2}\omega_l Y_l^{(4)}(k+1). \end{aligned}$$

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

in [28], the new algorithm is more general which includes many kind of iterative formulas.

In what follows, the convergence properties of the RGI method are analysed below. At the same time, we also provide the detailed proof of convergence theory.

4. Convergence analysis of the RGI algorithm

The section presents the sufficient and necessary condition for convergence of the RGI algorithm. Furthermore, to overcome the high-dimensional calculation problem of the iterative matrix, we further discuss the sufficient condition for convergence.

Theorem 4.1: *Assume that the generalized coupled complex conjugate and transpose matrix Equation (1) has a unique solution, then the iterative sequences $Y_l(k)$ $l \in \mathbb{I}[1,q]$ converge to the exact solution Y_l^* by the RGI algorithm for any initial matrices $Y_l(0)$ for all $l \in \mathbb{I}[1,q]$ if and only if the convergence factor μ is selected to satisfy*

$$0 < \mu < \frac{2}{\|QM^{\frac{1}{2}}\|_2^2}, \tag{42}$$

where

$$M = \begin{pmatrix} \frac{1}{4}\omega_1(1-\omega_1)I_{4s_1r_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{4}\omega_2(1-\omega_2)I_{4s_2r_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{4}\omega_q(1-\omega_q)I_{4s_qr_q} \end{pmatrix}, \tag{43}$$

$$Q = \begin{pmatrix} Q_{11} & Q_{12} & \cdots & Q_{1q} \\ Q_{21} & Q_{22} & \cdots & Q_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{p1} & Q_{p2} & \cdots & Q_{pq} \end{pmatrix}, \tag{44}$$

$$\begin{aligned} Q_{ij} = {} & (B_{ij}^{\nabla})^T \otimes (A_{ij}^{\nabla}) + (D_{ij}^{\nabla})^T E_{sj} \otimes (C_{ij}^{\nabla}) E_{rj} \\ & + \left((F_{ij}^{\nabla})^T E_{rj} \otimes (E_{ij}^{\nabla}) E_{sj}\right) P_{4r_js_j} \\ & + \left((H_{ij}^{\nabla})^T \otimes (G_{ij}^{\nabla})\right) P_{4r_js_j}, i \in \mathbb{I}\left[1,p\right], \\ & l \in \mathbb{I}\left[1,q\right]. \end{aligned} \tag{45}$$

Proof: Denote

$$\begin{aligned} & \tilde{Y}_l^{(1)}(k) = Y_l^{(1)}(k) - Y_l^*, \tilde{Y}_l^{(2)}(k) = Y_l^{(2)}(k) - Y_l^*, \\ & \tilde{Y}_l^{(3)}(k) = Y_l^{(3)}(k) - Y_l^*, \\ & \tilde{Y}_l^{(4)}(k) = Y_l^{(4)}(k) - Y_l^*, \\ & \quad \tilde{Y}_l(k) = Y_l(k) - Y_l^*, l \in \mathbb{I}[1,q]. \end{aligned} \tag{46}$$

To facilitate our statement, the expressions of $Z_i(k)$ ($i \in \mathbb{I}[1,p]$) are defined as follows

$$\begin{aligned} Z_i(k) = {} & \sum_{j=1}^{q} (A_{ij}\widetilde{Y}_j(k)B_{ij} + C_{ij}\overline{\widetilde{Y}}_j(k)D_{ij} + E_{ij}\widetilde{Y}_j(k)^T F_{ij} \\ & + G_{ij}\widetilde{Y}_j(k)^H H_{ij}). \end{aligned} \tag{47}$$

From the definition of error matrices and the expression of $Y_l^{(1)}(k+1)$ in the RGI algorithm, we derive that for $l \in \mathbb{I}[1,q]$

$$\begin{aligned} \widetilde{Y}_l^{(1)}(k+1) & = Y_l^{(1)}(k+1) - Y_l^* \\ & = \widetilde{Y}_l^{(1)}(k) - \frac{1}{2}\mu\omega_l \sum_{i=1}^{p} A_{il}^H \\ & \times \left[\sum_{j=1}^{q} (A_{ij}\widetilde{Y}_j(k)B_{ij} + C_{ij}\overline{\widetilde{Y}}_j(k)D_{ij} \right. \\ & \left. + E_{ij}\widetilde{Y}_j(k)^T F_{ij} + G_{ij}\widetilde{Y}_j(k)^H H_{ij}\right] B_{il}^H \\ & = \widetilde{Y}_l^{(1)}(k) - \frac{1}{2}\mu\omega_l \sum_{i=1}^{p} A_{il}^H Z_i(k) B_{il}^H. \end{aligned} \tag{48}$$

It follows from the expression of $Y_l^{(2)}(k+1)$ in the RGI method that

$$\begin{aligned} \widetilde{Y}_l^{(2)}(k+1) & = Y_l^{(2)}(k+1) - Y_l^* \\ & = \widetilde{Y}_l^{(2)}(k) - \frac{1}{2}\mu\omega_l \sum_{i=1}^{p} C_{il}^T \\ & \times \left[\overline{\begin{array}{c} \sum_{j=1}^{q} (A_{ij}\widetilde{Y}_j(k)B_{ij} + C_{ij}\overline{\widetilde{Y}}_j(k)D_{ij} \\ + E_{ij}\widetilde{Y}_j(k)^T F_{ij} + G_{ij}\widetilde{Y}_j(k)^H H_{ij} \end{array}}\right] D_{il}^T \\ & = \widetilde{Y}_l^{(2)}(k) - \frac{1}{2}\mu\omega_l \sum_{i=1}^{p} C_{il}^T \overline{Z_i(k)} D_{il}^T. \end{aligned} \tag{49}$$

By the expression of $Y_l^{(3)}(k+1)$ in Algorithm 1, for $l \in \mathbb{I}[1,q]$ one has

$$\begin{aligned} \widetilde{Y}_l^{(3)}(k+1) & = Y_l^{(3)}(k+1) - Y_l^* \\ & = \widetilde{Y}_l^{(3)}(k) - \frac{1}{2}\mu(1-\omega_l) \sum_{i=1}^{p} \overline{F_{il}} \\ & \times \left[\sum_{j=1}^{q} (A_{ij}\widetilde{Y}_j(k)B_{ij} + C_{ij}\overline{\widetilde{Y}}_j(k)D_{ij} \right. \\ & \left. + E_{ij}\widetilde{Y}_j(k)^T F_{ij} + G_{ij}\widetilde{Y}_j(k)^H H_{ij}\right]^T \overline{E_{il}} \end{aligned}$$

$$= \widetilde{Y}_l^{(3)}(k) - \frac{1}{2}\mu(1-\omega_l)\sum_{i=1}^{p}\overline{F_{il}}Z_i(k)^T\overline{E_{il}}. \quad (50)$$

It follows from the expression of $Y_l^{(4)}(k+1)$ in Algorithm 1 that for $l \in \mathbb{I}[1,q]$

$$\begin{aligned}
\widetilde{Y}_l^{(4)}(k+1) &= Y_l^{(4)}(k+1) - Y_l^* \\
&= \widetilde{Y}_l^{(4)}(k) - \frac{1}{2}\mu(1-\omega_l)\sum_{i=1}^{p}H_{il} \\
&\quad \times \left[\sum_{j=1}^{q}(A_{ij}\widetilde{Y}_j(k)B_{ij} + C_{ij}\overline{\widetilde{Y}_j(k)}D_{ij} + E_{ij}\widetilde{Y}_j(k)^TF_{ij} \right. \\
&\quad \left. + G_{ij}\widetilde{Y}_j(k)^HH_{ij}\right]^H G_{il} \\
&= \widetilde{Y}_l^{(4)}(k) - \frac{1}{2}\mu(1-\omega_l)\sum_{i=1}^{p}H_{il}Z_i(k)^HG_{il}. \quad (51)
\end{aligned}$$

Combing (48)–(51) with Line 5 of the RGI algorithm leads to

$$\begin{aligned}
\widetilde{Y}_l(k+1) &= Y_l(k+1) - Y_l^* \\
&= \frac{1}{2}(1-\omega_l)Y_l^{(1)}(k+1) \\
&\quad + \frac{1}{2}(1-\omega_l)Y_l^{(2)}(k+1) \\
&\quad + \frac{1}{2}\omega_lY_l^{(3)}(k+1) \\
&\quad + \frac{1}{2}\omega_lY_l^{(4)}(k+1) - Y_l^* \\
&= \frac{1}{2}(1-\omega_l)\widetilde{Y}_l^{(1)}(k+1) \\
&\quad + \frac{1}{2}(1-\omega_l)\widetilde{Y}_l^{(2)}(k+1) \\
&\quad + \frac{1}{2}\omega_l\widetilde{Y}_l^{(3)}(k+1) \\
&\quad + \frac{1}{2}\omega_l\widetilde{Y}_l^{(4)}(k+1) \\
&= \widetilde{Y}_l(k) - \frac{1}{4}\mu\omega_l(1-\omega_l)\sum_{i=1}^{p}(A_{il}^HZ_i(k)B_{il}^H \\
&\quad + C_{il}^T\overline{Z_i(k)}D_{il}^T + \overline{F_{il}}Z_i(k)^T\overline{E_{il}} \\
&\quad + H_{il}Z_i(k)^HG_{il}). \quad (52)
\end{aligned}$$

According to Lemma 2.4, taking the real representation on both sides of (52) results in

$$\begin{aligned}
&(\widetilde{Y}_l(k+1))^\triangledown \\
&= (\widetilde{Y}_l(k))^\triangledown - \frac{1}{4}\mu\omega_l(1-\omega_l)\sum_{i=1}^{p}(A_{il}^HZ_i(k)B_{il}^H \\
&\quad + C_{il}^T\overline{Z_i(k)}D_{il}^T + \overline{F_{il}}Z_i(k)^T\overline{E_{il}} + H_{il}Z_i(k)^HG_{il})^\triangledown \\
&= (\widetilde{Y}_l(k))^\triangledown - \frac{1}{4}\mu\omega_l(1-\omega_l)\sum_{i=1}^{p} \\
&\quad \times \Big[(A_{il}^H)^\triangledown(Z_i(k))^\triangledown(B_{il}^H)^\triangledown + (C_{il}^T)^\triangledown(\overline{Z_i(k)})^\triangledown(D_{il}^T)^\triangledown \\
&\quad + (\overline{F_{il}})^\triangledown(Z_i(k)^T)^\triangledown(\overline{E_{il}})^\triangledown + (H_{il}^\triangledown)(Z_i(k)^H)^\triangledown(G_{il}^\triangledown)\Big] \\
&= (\widetilde{Y}_l(k))^\triangledown - \frac{1}{4}\mu\omega_l(1-\omega_l)\sum_{i=1}^{p} \\
&\quad \times \Big[(A_{il}^\triangledown)^T(Z_i(k))^\triangledown(B_{il}^\triangledown)^T \\
&\quad + E_{r_l}(C_{il}^\triangledown)^TE_{m_i}E_{m_i}(Z_i(k))^\triangledown E_{n_i}E_{n_i}(D_{il}^\triangledown)^TE_{s_l} \\
&\quad + E_{r_l}(F_{il}^\triangledown)E_{n_i}E_{n_i}(Z_i(k)^\triangledown)^TE_{m_i}E_{m_i}(E_{il}^\triangledown)E_{s_l} \\
&\quad + (H_{il}^\triangledown)(Z_i(k)^\triangledown)^T(G_{il}^\triangledown)\Big] \\
&= (\widetilde{Y}_l(k))^\triangledown - \frac{1}{4}\mu\omega_l(1-\omega_l)\sum_{i=1}^{p} \\
&\quad \times \Big[(A_{il}^\triangledown)^T(Z_i(k))^\triangledown(B_{il}^\triangledown)^T \\
&\quad + E_{r_l}(C_{il}^\triangledown)^T(Z_i(k))^\triangledown(D_{il}^\triangledown)^TE_{s_l} \\
&\quad + E_{r_l}(F_{il}^\triangledown)(Z_i(k)^\triangledown)^T(E_{il}^\triangledown)E_{s_l} \\
&\quad + (H_{il}^\triangledown)(Z_i(k)^\triangledown)^T(G_{il}^\triangledown)\Big]. \quad (53)
\end{aligned}$$

Using straightening operator in (53) and applying Definition 2.3, for $l \in \mathbb{I}[1,q]$ we have

$$\begin{aligned}
&vec[(\widetilde{Y}_l(k+1))^\triangledown] \\
&= vec[(\widetilde{Y}_l(k))^\triangledown] - \frac{1}{4}\mu\omega_l(1-\omega_l)\sum_{i=1}^{p} \\
&\quad \times \Big[(B_{il}^\triangledown)\otimes(A_{il}^\triangledown)^T + E_{s_l}(D_{il}^\triangledown)\otimes E_{r_l}(C_{il}^\triangledown)^T \\
&\quad + P_{4r_ls_l}{}^T\left(E_{r_l}(F_{il}^\triangledown)\otimes E_{s_l}(E_{il}^\triangledown)^T\right) \\
&\quad + P_{4r_ls_l}{}^T\left((H_{il}^\triangledown)\otimes(G_{il}^\triangledown)^T\right)\Big]vec[(Z_i(k))^\triangledown]. \quad (54)
\end{aligned}$$

Furthermore, by applying the real representation on two sides of (47), we get

$$\begin{aligned}
(Z_i(k))^\triangledown &= \sum_{j=1}^{q}(A_{ij}\widetilde{Y}_j(k)B_{ij} + C_{ij}\overline{\widetilde{Y}_j(k)}D_{ij} \\
&\quad + E_{ij}\widetilde{Y}_j(k)^TF_{ij} + G_{ij}\widetilde{Y}_j(k)^HH_{ij})^\triangledown \\
&= \sum_{j=1}^{q}((A_{ij}^\triangledown)(\widetilde{Y}_j(k))^\triangledown(B_{ij}^\triangledown) \\
&\quad + (C_{ij}^\triangledown)E_{r_j}(\widetilde{Y}_j(k))^\triangledown E_{s_j}(D_{ij}^\triangledown) \\
&\quad + (E_{ij}^\triangledown)E_{s_j}((\widetilde{Y}_j(k))^\triangledown)^TE_{r_j}(F_{ij}^\triangledown) \\
&\quad + (G_{ij}^\triangledown)(\widetilde{Y}_j(k)^\triangledown)^T(H_{ij}^\triangledown)). \quad (55)
\end{aligned}$$

Then, utilizing the vec-operator in (55) can deduce

$$vec[(Z_i(k))^\triangledown]$$

$$
\begin{aligned}
&= vec\Bigg[\sum_{j=1}^{q}\Big((A_{ij}^{\triangledown})(\widetilde{Y}_j(k))^{\triangledown}(B_{ij}^{\triangledown})\\
&\quad + (C_{ij}^{\triangledown})E_{rj}(\widetilde{Y}_j(k))^{\triangledown}E_{sj}(D_{ij}^{\triangledown})\\
&\quad + (E_{ij}^{\triangledown})E_{s_j}((\widetilde{Y}_j(k))^{\triangledown})^{T}E_{r_j}(F_{ij}^{\triangledown})\\
&\quad +(G_{ij}^{\triangledown})(\widetilde{Y}_j(k)^{\triangledown})^{T}(H_{ij}^{\triangledown})\Big)\Bigg]\\
&= \sum_{j=1}^{q}\Big[(B_{ij}^{\triangledown})^{T}\otimes(A_{ij}^{\triangledown}) + (D_{ij}^{\triangledown})^{T}E_{s_j}\otimes(C_{ij}^{\triangledown})E_{r_j}\\
&\quad + \Big((F_{ij}^{\triangledown})^{T}E_{r_j}\otimes(E_{ij}^{\triangledown})E_{s_j}\Big)P_{4r_js_j}\\
&\quad + \Big((H_{ij}^{\triangledown})^{T}\otimes(G_{ij}^{\triangledown})\Big)P_{4r_js_j}\Big]vec[\widetilde{Y}_j(k)^{\triangledown}]. \qquad (56)
\end{aligned}
$$

Finally, substituting (56) into (54) results in

$$
\begin{aligned}
&vec[(\widetilde{Y}_l(k+1))^{\triangledown}]\\
&= vec[(\widetilde{Y}_l(k))^{\triangledown}] - \frac{1}{4}\mu\omega_l(1-\omega_l)\sum_{i=1}^{p}\\
&\quad\times\Big[(B_{il}^{\triangledown})\otimes(A_{il}^{\triangledown})^{T} + E_{s_l}(D_{il}^{\triangledown})\otimes E_{r_l}(C_{il}^{\triangledown})^{T}\\
&\quad + P_{4r_ls_l}^{T}\Big(E_{r_l}(F_{il}^{\triangledown})\otimes E_{s_l}(E_{il}^{\triangledown})^{T}\Big)\\
&\quad + P_{4r_ls_l}^{T}\Big((H_{il}^{\triangledown})\otimes(G_{il}^{\triangledown})^{T}\Big)\Big]\sum_{j=1}^{q}\\
&\quad\times\Big[(B_{ij}^{\triangledown})^{T}\otimes(A_{ij}^{\triangledown})\\
&\quad + (D_{ij}^{\triangledown})^{T}E_{s_j}\otimes(C_{ij}^{\triangledown})E_{r_j}\\
&\quad + \Big((F_{ij}^{\triangledown})^{T}E_{r_j}\otimes(E_{ij}^{\triangledown})E_{s_j}\Big)P_{4r_js_j}\\
&\quad + \Big((H_{ij}^{\triangledown})^{T}\otimes(G_{ij}^{\triangledown})\Big)P_{4r_js_j}\Big]vec[\widetilde{Y}_j(k)^{\triangledown}]. \qquad (57)
\end{aligned}
$$

Denote

$$
\begin{aligned}
vec[(\widetilde{Y}(k))^{\triangledown}] &= \Big[vec((\widetilde{Y}_1(k))^{\triangledown})^{T}, vec((\widetilde{Y}_2(k))^{\triangledown})^{T},\\
&\qquad \ldots, vec((\widetilde{Y}_q(k))^{\triangledown})^{T}\Big]^{T}. \qquad (58)
\end{aligned}
$$

Thus, Equation (57) can be written as the following expression

$$
\begin{aligned}
vec[(\widetilde{Y}(k+1))^{\triangledown}] &= vec[(\widetilde{Y}(k))^{\triangledown}]\\
&\quad - \mu MQ^{T}Qvec[(\widetilde{Y}(k))^{\triangledown}]\\
&= [I-\mu MQ^{T}Q]vec[(\widetilde{Y}(k))^{\triangledown}]. \qquad (59)
\end{aligned}
$$

The matrices M and Q are given in (43) and (44). It follows from Equation (59) that the matrix $[I-\mu MQ^{T}Q]$ is the iterative matrix of Algorithm 1. So the sufficient and necessary condition for convergence of the RGI algorithm is

$$
\rho(I-\mu MQ^{T}Q) < 1. \qquad (60)
$$

Due to the fact that the iterative matrix $MQ^{T}Q$ is similar to $M^{\frac{1}{2}}Q^{T}QM^{\frac{1}{2}}$, and $M^{\frac{1}{2}}Q^{T}QM^{\frac{1}{2}}$ is symmetric matrix, so one obtains

$$
\begin{aligned}
\lambda_i(I-\mu MQ^{T}Q) &= 1-\mu\lambda_i(M^{\frac{1}{2}}Q^{T}QM^{\frac{1}{2}})\\
&= 1-\mu\sigma_i^{2}(QM^{\frac{1}{2}}), \quad i\in\mathbb{I}[1,p]. \qquad (61)
\end{aligned}
$$

Since $\rho(I-\mu MQ^{T}Q) < 1$, it follows that

$$
-1 < 1-\mu\sigma_i^{2}(QM^{\frac{1}{2}}) < 1 \; or \; 0 < \mu < \frac{2}{\sigma_i^{2}(QM^{\frac{1}{2}})}, \quad i\in\mathbb{I}[1,p]. \qquad (62)
$$

Finally, the range of convergence parameter μ making the RGI algorithm convergent is

$$
0 < \mu < \frac{2}{\|QM^{\frac{1}{2}}\|_2^{2}}. \qquad (63)
$$

Here, we complete the proof of Theorem 4.1. ■

In order to further effectively utilize the RGI algorithm, we should get the optimal convergence parameter μ of this method. When $\rho(I-M^{\frac{1}{2}}Q^{T}QM^{\frac{1}{2}})$ reaches minimum value, the convergence behaviour of the RGI method achieves the optimum. According to Lemma 2.5, the necessary and sufficient condition for $\rho(I-M^{\frac{1}{2}}Q^{T}QM^{\frac{1}{2}})$ is

$$
|1-\mu\sigma_{\min}^{2}(QM^{\frac{1}{2}})| = |1-\mu\sigma_{\max}^{2}(QM^{\frac{1}{2}})|. \qquad (64)
$$

By simple calculations, the optimal convergence parameter is obtained as

$$
\mu_{\text{opt}} = \frac{2}{\sigma_{\min}^{2}\left(QM^{\frac{1}{2}}\right)+\sigma_{\max}^{2}\left(QM^{\frac{1}{2}}\right)}. \qquad (65)
$$

Then, we will further discuss the convergence properties of the RGI method with the relaxation parameters $\omega_l=\omega$ for $l\in\mathbb{I}[1,q]$. Some relevant conclusions are proposed below.

Theorem 4.2: *Assume that $Y(0)=(Y_1(0),Y_2(0),\ldots,Y_q(0))$ and $Y^{*}(k)=(Y_1^{*}(k),Y_2^{*}(k),\ldots,Y_q^{*}(k))$ represent the initial value and unique solution of the RGI algorithm, respectively. Based on the conditions of Theorem 4.1, if the relaxation factor are selected as $\omega_l=\omega$ for $l\in\mathbb{I}[1,q]$, it holds that*

$$
\begin{aligned}
&\|Y(k)-Y^{*}\|\\
&\quad\le \rho^{k}\left(I-\frac{1}{4}\mu\omega(1-\omega)Q^{T}Q\right)\|Y(0)-Y^{*}\|. \qquad (66)
\end{aligned}
$$

And the optimal convergence parameter is

$$\mu_{\text{opt}} = \frac{8}{\omega(1-\omega)(\sigma_{\max}^2(Q)+\sigma_{\min}^2(Q))}. \quad (67)$$

Under this situation, the following inequality holds

$$\|Y(k)-Y^*\| \leq \left(\frac{cond^2(QM^{\frac{1}{2}})-1}{cond^2(QM^{\frac{1}{2}})+1}\right)^k \|Y(0)-Y^*\|. \quad (68)$$

Proof: According to the fact that $I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ$ is the symmetric matrix, it has

$$\left\|I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ\right\|_2 = \rho\left(I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ\right). \quad (69)$$

Combining Expression (59) with the properties of matrix norms, we derive

$$\begin{aligned}
&\|(\widetilde{Y}(k+1))^{\triangledown}\| \\
&= \|vec[(\widetilde{Y}(k+1))^{\triangledown}]\|_2 \\
&= \left\|\left[I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ\right] vec[(\widetilde{Y}(k))^{\triangledown}]\right\|_2 \\
&\leq \|[I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ]\|_2 \|vec[(\widetilde{Y}(k))^{\triangledown}]\|_2 \\
&= \left\|\left[I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ\right]\right\|_2 \|vec[(\widetilde{Y}(k))^{\triangledown}]\| \\
&= \rho\left(I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ\right)\|(\widetilde{Y}(k))^{\triangledown}\|, \quad (70)
\end{aligned}$$

with

$$(\widetilde{Y}(k))^{\triangledown} = [(\widetilde{Y}_1(k))^{\triangledown}, (\widetilde{Y}_2(k))^{\triangledown}, \dots, (\widetilde{Y}_q(k))^{\triangledown}]. \quad (71)$$

Based on Lemma 2.4 and $\|A^{\triangledown}\|^2 = 2\|A\|^2$, it holds

$$\begin{aligned}
\|\widetilde{Y}(k+1)\| &= \frac{1}{\sqrt{2}}\|(\widetilde{Y}(k+1))^{\triangledown}\| \\
&\leq \rho\left(I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ\right)\frac{1}{\sqrt{2}}\|(\widetilde{Y}(k))^{\triangledown}\| \\
&= \rho\left(I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ\right)\|\widetilde{Y}(k)\|. \quad (72)
\end{aligned}$$

By the definition of the error matrix and Inequality (70), we derive that

$$\begin{aligned}
\|Y(k)-Y^*\| &= \|\widetilde{Y}(k)\| \\
&\leq \rho^k\left(I-\frac{1}{4}\mu\omega(1-\omega)MQ^TQ\right)\|Y(0)-Y^*\|. \quad (73)
\end{aligned}$$

Moreover, when $\rho(I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ)$ is minimized, the convergence performance of the RGI algorithm can achieve optimal. So we should choose the optimal parameter μ_{opt} to minimize $\rho(I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ)$. The minimum value of $\rho(I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ)$ is

$$\begin{aligned}
&\min\ \rho\left(I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ\right) \\
&= \min\ \max\left\{\left|1-\frac{1}{4}\mu\omega(1-\omega)\sigma_i^2(Q)\right|\right\} \\
&= \min\ \max\left\{\left|1-\frac{1}{4}\mu\omega(1-\omega)\sigma_{\max}^2(Q)\right|,\right. \\
&\quad \left.\left|1-\frac{1}{4}\mu\omega(1-\omega)\sigma_{\min}^2(Q)\right|\right\}, \quad (74)
\end{aligned}$$

which indicates that $|1-\frac{1}{4}\mu\omega(1-\omega)\sigma_{\max}^2(Q)| = |1-\frac{1}{4}\mu\omega(1-\omega)\sigma_{\min}^2(Q)|$ has a non-trivial solution. By simple derivations, the best convergence parameter is

$$\mu_{\text{opt}} = \frac{8}{\omega(1-\omega)(\sigma_{\max}^2(Q)+\sigma_{\min}^2(Q))}. \quad (75)$$

If the convergence parameter μ is selected as the one in (75), it will lead to

$$\begin{aligned}
&\rho(I-\frac{1}{4}\mu\omega(1-\omega)Q^TQ) \\
&= \max\left\{1-\frac{1}{4}\mu\omega(1-\omega)\lambda_i(Q^TQ)\right\} \\
&= 1-\frac{1}{4}\frac{8}{\omega(1-\omega)(\sigma_{\max}^2(Q)+\sigma_{\min}^2(Q))}\} \\
&\quad \times \omega(1-\omega)\lambda_{\min}(Q^TQ) \\
&= 1-\frac{2}{(\sigma_{\max}^2(Q)+\sigma_{\min}^2(Q))}\lambda_{\min}(Q^TQ) \\
&= 1-\frac{2\sigma_{\min}^2(Q)}{\sigma_{\max}^2(Q)+\sigma_{\min}^2(Q)} \\
&= \frac{\sigma_{\max}^2(Q)-\sigma_{\min}^2(Q)}{\sigma_{\max}^2(Q)+\sigma_{\min}^2(Q)} \\
&= \frac{cond^2(Q)-1}{cond^2(Q)+1}. \quad (76)
\end{aligned}$$

Then Equation (68) can be derived by substituting Equation (76) into (73). ■

Remark 4.1: In Theorem 4.1, the sufficient and necessary condition for convergence of the RGI method is obtained. However, $\|QM^{\frac{1}{2}}\|_2^2$ involves the calculation of real representation and Kronecker product, which leads to high-dimensional problems. In order to overcome this drawback and develop computational efficiency, we further derive sufficient condition for the convergence with less computational complexity.

Corollary 4.1: *Assume that the conditions of Theorem* 4.1 *are satisfied, then Algorithm* 1 *is convergent for any initial* *matrix if the parameters ω and μ are selected to satisfy the following inequality*

$$0<\mu\le\frac{2}{\sum_{i=1}^{p}\sum_{j=1}^{q}\omega_j(1-\omega_j)\left[\|B_{ij}\|_2^2\|A_{ij}\|_2^2+\|D_{ij}\|_2^2\|C_{ij}\|_2^2+\|F_{ij}\|_2^2\|E_{ij}\|_2^2+\|H_{ij}\|_2^2\|G_{ij}\|_2^2\right]}. \tag{77}$$

Proof: By the properties of Frobenius norm of matrix, one has

$$\begin{aligned}\|QM^{\frac{1}{2}}\|_2^2 &\le \sum_{i=1}^{p}\sum_{j=1}^{q}\Bigg\|\left(\sqrt{\frac{1}{4}\omega_j(1-\omega_j)}\right)(B_{ij}^{\triangledown})^T\otimes(A_{ij}^{\triangledown})\\&\quad+(D_{ij}^{\triangledown})^TE_{s_j}\otimes(C_{ij}^{\triangledown})E_{r_j}\\&\quad+\left((F_{ij}^{\triangledown})^TE_{r_j}\otimes(E_{ij}^{\triangledown})E_{s_j}\right)P_{4r_js_j}\\&\quad+\left((H_{ij}^{\triangledown})^T\otimes(G_{ij}^{\triangledown})\right)P_{4r_js_j}\Bigg\|_2^2\\&\le\sum_{i=1}^{p}\sum_{j=1}^{q}\frac{1}{4}\omega_j(1-\omega_j)\Big[\|(B_{ij}^{\triangledown})^T\otimes(A_{ij}^{\triangledown})\|_2\\&\quad+\|(D_{ij}^{\triangledown})^TE_{s_j}\otimes(C_{ij}^{\triangledown})E_{r_j}\|_2\\&\quad+\|\left((F_{ij}^{\triangledown})^TE_{r_j}\otimes(E_{ij}^{\triangledown})E_{s_j}\right)P_{4r_js_j}\|_2\\&\quad+\|\left((H_{ij}^{\triangledown})^T\otimes(G_{ij}^{\triangledown})\right)P_{4r_js_j}\|_2\Big]^2\\&\le\sum_{i=1}^{p}\sum_{j=1}^{q}\frac{1}{2}\omega_j(1-\omega_j)\Big[\Big(\|(B_{ij}^{\triangledown})^T\otimes(A_{ij}^{\triangledown})\|_2\\&\quad+\|(D_{ij}^{\triangledown})^TE_{s_j}\otimes(C_{ij}^{\triangledown})E_{r_j}\|_2\Big)^2\\&\quad+\Big(\|\left((F_{ij}^{\triangledown})^TE_{r_j}\otimes(E_{ij}^{\triangledown})E_{s_j}\right)P_{4r_js_j}\|_2\\&\quad+\|\left((H_{ij}^{\triangledown})^T\otimes(G_{ij}^{\triangledown})\right)P_{4r_js_j}\|_2\Big)^2\Big]\\&\le\sum_{i=1}^{p}\sum_{j=1}^{q}\omega_j(1-\omega_j)\Big[\|(B_{ij}^{\triangledown})^T\otimes(A_{ij}^{\triangledown})\|_2^2\\&\quad+\|(D_{ij}^{\triangledown})^TE_{s_j}\otimes(C_{ij}^{\triangledown})E_{r_j}\|_2^2\\&\quad+\|\left((F_{ij}^{\triangledown})^TE_{r_j}\otimes(E_{ij}^{\triangledown})E_{s_j}\right)P_{4r_js_j}\|_2^2\\&\quad+\|\left((H_{ij}^{\triangledown})^T\otimes(G_{ij}^{\triangledown})\right)P_{4r_js_j}\|_2^2\Big]\\&=\sum_{i=1}^{p}\sum_{j=1}^{q}\omega_j(1-\omega_j)\Big[\|(B_{ij}^{\triangledown})^T\otimes(A_{ij}^{\triangledown})\|_2^2\\&\quad+\|(D_{ij}^{\triangledown})^TE_{s_j}\otimes(C_{ij}^{\triangledown})E_{r_j}\|_2^2\\&\quad+\|\left((F_{ij}^{\triangledown})^TE_{r_j}\otimes(E_{ij}^{\triangledown})E_{s_j}\right)\|_2^2\\&\quad+\|\left((H_{ij}^{\triangledown})^T\otimes(G_{ij}^{\triangledown})\right)\|_2^2\Big].\end{aligned}\tag{78}$$

Notice the fact that $\|A\otimes B\|_2=\|A\|_2\|B\|_2$, $\|A^{\triangledown}\|_2=\|A\|_2$, and we have the following inequality

$$\begin{aligned}\|QM^{\frac{1}{2}}\|_2^2&\le\sum_{i=1}^{p}\sum_{j=1}^{q}\omega_j(1-\omega_j)\Big[\|(B_{ij}^{\triangledown})^T\|_2^2\|(A_{ij}^{\triangledown})\|_2^2\\&\quad+\|(D_{ij}^{\triangledown})^TE_{s_j}\|_2^2\|(C_{ij}^{\triangledown})E_{r_j}\|_2^2\\&\quad+\|(F_{ij}^{\triangledown})^TE_{r_j}\|_2^2\|E_{ij}^{\triangledown}E_{s_j}\|_2^2\\&\quad+\|(H_{ij}^{\triangledown})^T\|_2^2\|(G_{ij}^{\triangledown})\|_2^2\Big]\\&=\sum_{i=1}^{p}\sum_{j=1}^{q}\omega_j(1-\omega_j)\Big[\|E_{n_i}(B_{ij}^T)^{\triangledown}E_{s_j}\|_2^2\|A_{ij}\|_2^2\\&\quad+\|E_{n_i}(D_{ij}^T)^{\triangledown}\|_2^2\|C_{ij}\|_2^2\\&\quad+\|E_{n_i}(F_{ij}^T)^{\triangledown}\|_2^2\|E_{ij}^{\triangledown}E_{s_j}\|_2^2\\&\quad+\|E_{n_i}(H_{ij}^T)^{\triangledown}E_{r_i}\|_2^2\|G_{ij}^{\triangledown}\|_2^2\Big]\\&=\sum_{i=1}^{p}\sum_{j=1}^{q}\omega_j(1-\omega_j)\left[\|B_{ij}\|_2^2\|A_{ij}\|_2^2\right.\\&\quad+\|D_{ij}\|_2^2\|C_{ij}\|_2^2+\|F_{ij}\|_2^2\|E_{ij}\|_2^2\\&\quad\left.+\|H_{ij}\|_2^2\|G_{ij}\|_2^2\right].\end{aligned}\tag{79}$$

By combining (79) with (42), the conclusion of Corollary 4.1 is correct. ■

5. Numerical experimental results

In this section, we present two numerical examples to testify the effectiveness and feasibility of the RGI algorithm proposed in this paper. All experiments are performed on a personal computer with AMD Ryzen 5 5600U with Radeon Graphics 2.30 GHz, 16.0 GB. The programming language is computed in MATLAB R2021b. In our experiment, we compare the convergence behaviour of the RGI algorithm with the GI one in terms of the iterative number (IT), calculation time (CPU) in seconds and the relative error (ERR).

Example 5.1 ([34]): We consider the generalized coupled complex conjugate and transpose Sylvester matrix Equation (1) in the case of $p=q=4$, and its form is as follows

$$\begin{cases}A_{11}Y_1B_{11}+C_{13}\overline{Y}_3D_{13}\\\quad+E_{12}Y_2^TF_{12}+G_{14}Y_4^HH_{14}=M_1,\\A_{22}Y_2B_{22}+C_{24}\overline{Y}_4D_{24}\\\quad+E_{23}Y_3^TF_{23}+G_{21}Y_1^HH_{21}=M_2,\\A_{33}Y_3B_{33}+C_{31}\overline{Y}_1D_{31}\\\quad+E_{34}Y_4^TF_{34}+G_{32}Y_2^HH_{32}=M_3,\\A_{44}Y_4B_{44}+C_{42}\overline{Y}_2D_{42}\\\quad+E_{41}Y_1^TF_{41}+G_{43}Y_3^HH_{43}=M_4,\end{cases}\tag{80}$$

with

$$A_{11} = \begin{bmatrix} -12-7i & 10-11i & -9+10i \\ 2-32i & 27-3i & 1-3i \\ 10+11i & 3-7i & -14-4i \end{bmatrix},$$

$$B_{11} = \begin{bmatrix} -17-7i & -8-25i & 13+1i \\ 7+4i & -2-9i & 0+6i \\ 7-11i & -4-2i & 7+6i \end{bmatrix},$$

$$A_{22} = \begin{bmatrix} 11-9i & -8-7i & -18-2i \\ 33-25i & -3+6i & 5-23i \\ 7-7i & -3+11i & -12-15i \end{bmatrix},$$

$$B_{22} = \begin{bmatrix} -4+13i & 7-14i & -10+2i \\ 12+5i & -4+3i & 8-16i \\ 1-5i & 19+7i & 7-7i \end{bmatrix},$$

$$A_{33} = \begin{bmatrix} 7+6i & -5+11i & 4+8i \\ -24-1i & 11-3i & 0+22i \\ 0-7i & -2-9i & 0-6i \end{bmatrix},$$

$$B_{33} = \begin{bmatrix} 4-9i & -20+15i & -23+20i \\ -25+4i & -4-2i & 11+1i \\ -14-4i & -4+8i & 10-1i \end{bmatrix},$$

$$A_{44} = \begin{bmatrix} -12+12i & -2+5i & 1-9i \\ 2-2i & 10-6i & -26+3i \\ -19+13i & -3+15i & -7-17i \end{bmatrix},$$

$$B_{44} = \begin{bmatrix} -1-5i & 2+9i & -3+4i \\ 16-7i & 1+4i & -9-5i \\ -8-6i & -6+4i & 1 \end{bmatrix},$$

$$C_{13} = \begin{bmatrix} 5+6i & 11+7i & -12+4i \\ -15+2i & -5+7i & 0-14i \\ 11+4i & -9-17i & 2+21i \end{bmatrix},$$

$$D_{13} = \begin{bmatrix} 6+3i & -22+9i & 10-4i \\ 16+17i & 6+2i & 0+2i \\ 14+3i & -12-2i & -7-2i \end{bmatrix},$$

$$C_{24} = \begin{bmatrix} -16+1i & 0-3i & -7-10i \\ -3 & -4+6i & -9+7i \\ -9-2i & 19-9i & 6-3i \end{bmatrix},$$

$$D_{24} = \begin{bmatrix} -6-14i & -7+20i & 4 \\ 0-19i & -6-8i & -5+8i \\ -12-6i & 0 & -7-17i \end{bmatrix},$$

$$C_{31} = \begin{bmatrix} -13-2i & 15+16i & 12-23i \\ -10-4i & -10-3i & -15+12i \\ -3-9i & 5+20i & -5+5i \end{bmatrix},$$

$$D_{31} = \begin{bmatrix} -6+6i & -9+6i & -20-19i \\ 9+14i & 1+21i & -8-12i \\ -12-12i & -17-4i & 3+8i \end{bmatrix},$$

$$C_{42} = \begin{bmatrix} -1+5i & -5+25i & -2-4i \\ 19-11i & -2-6i & -2+9i \\ -10+2i & -5-9i & 13+10i \end{bmatrix},$$

$$D_{42} = \begin{bmatrix} -5 & 0+1i & 10+16i \\ 2-1i & -24+2i & -6+1i \\ -1-12i & -6+14i & 4-13i \end{bmatrix},$$

$$E_{12} = \begin{bmatrix} -14+2i & 2+5i & 4-3i \\ 9+5i & -3+10i & 8+6i \\ -8-1i & 5+9i & -21-1i \end{bmatrix},$$

$$F_{12} = \begin{bmatrix} -13+1i & -21+6i & -11-3i \\ 2+7i & -10+4i & -10+10i \\ -3+5i & 9-11i & -13+13i \end{bmatrix},$$

$$E_{23} = \begin{bmatrix} 8-2i & -8+2i & 0-5i \\ 3+6i & 8-2i & -3-1i \\ 10-3i & -13+3i & -14+7i \end{bmatrix},$$

$$F_{23} = \begin{bmatrix} -16-27i & -26+2i & -12-10i \\ 6-8i & 2+33i & 1-9i \\ -5-12i & 12-18i & 29+7i \end{bmatrix},$$

$$E_{34} = \begin{bmatrix} 14+21i & -9+8i & 4-14i \\ 7+7i & 19-2i & 0+5i \\ 7+10i & -5+8i & 7-18i \end{bmatrix},$$

$$F_{34} = \begin{bmatrix} 0+8i & 16-12i & -8+17i \\ -5+18i & 4+12i & 10+9i \\ 10-1i & 12-14i & 17-5i \end{bmatrix},$$

$$E_{41} = \begin{bmatrix} 3+8i & -18+10i & -9+14i \\ -17+6i & 3+6i & 14+7i \\ 5 & -5-17i & -6+5i \end{bmatrix},$$

$$F_{41} = \begin{bmatrix} -5-1i & 16+3i & -5-13i \\ -19+6i & -12 & -9-3i \\ -12+4i & -6-12i & 6+3i \end{bmatrix},$$

$$G_{14} = \begin{bmatrix} 2-29i & -6-3i & 8+8i \\ -3+3i & -6-3i & 1-14i \\ 1 & -3-11i & -1+19i \end{bmatrix},$$

$$H_{14} = \begin{bmatrix} 8-8i & -10-2i & -5+6i \\ 1 & -3+6i & 16+6i \\ -9+8i & 9 & -3-2i \end{bmatrix},$$

$$G_{21} = \begin{bmatrix} 15+15i & -5+19i & -16-4i \\ -7-14i & 5+12i & -11-16i \\ -1+5i & 5-3i & -1-3i \end{bmatrix},$$

$$H_{21} = \begin{bmatrix} -4+15i & -18+3i & 6-10i \\ 6-1i & 4-3i & 9-16i \\ 10+2i & -17+12i & -2-8i \end{bmatrix},$$

$$G_{32} = \begin{bmatrix} 14-4i & 25-3i & 5-2i \\ 4-5i & 2-16i & 3-2i \\ 7-22i & 2+5i & 7-18i \end{bmatrix},$$

$$H_{32} = \begin{bmatrix} 10+5i & 8+1i & 12-9i \\ 12-3i & 6-7i & -3+13i \\ 4-3i & -7-4i & -6-10i \end{bmatrix},$$

$$G_{43} = \begin{bmatrix} 5+21i & 1+20i & -4+12i \\ -7-8i & 6-6i & -2+15i \\ -13+5i & -22+4i & -2+5i \end{bmatrix},$$

$$H_{43} = \begin{bmatrix} -13-18i & 3-26i & -3-8i \\ -2+16i & 9-4i & 11-5i \\ 9+7i & 5+4i & 2-14i \end{bmatrix},$$

$$M_1 = \begin{bmatrix} -2418 + 3322i & 10353 - 966i & 5238 + 1933i \\ -11927 - 7210i & -7206 - 12568i & 4614 + 7638i \\ 1619 - 9753i & 16692 + 11938i & -3798 - 2865i \end{bmatrix},$$

$$M_2 = \begin{bmatrix} -4750 + 14828i & 10137 - 3634i & -18315 + 2472i \\ -11651 + 15269i & -11063 - 9783i & -16515 + 18210i \\ -2388 + 17370i & 2934 - 1222i & -3823 + 2947i \end{bmatrix},$$

$$M_3 = \begin{bmatrix} 22162 - 7358i & 22122 - 18790i & -3570 - 2827i \\ 7263 + 4634i & 4142 + 7164i & 6578 + 26838i \\ 12844 - 6822i & -11828 - 13695i & 9789 - 20700i \end{bmatrix},$$

$$M_4 = \begin{bmatrix} -5068 - 9357i & 6306 - 13376i & -3738 - 6683i \\ 9215 + 5146i & 8946 - 8825i & 14012 - 7731i \\ -2927 - 1660i & -5342 + 4327i & 6279 - 2721i \end{bmatrix}.$$

This matrix equation has the following exact solution

$$Y_1 = \begin{bmatrix} -9 + 8i & -12 - 5i & 9 + 7i \\ 6 + 7i & -10 - 4i & 14 + 2i \\ 6 - 10i & 10 + 11i & -1 \end{bmatrix},$$

$$Y_2 = \begin{bmatrix} 13 + 6i & -19 + 6i & -11 - 11i \\ 16 - 15i & 16 - 16i & 9 + 11i \\ -7 - 10i & 5 + 6i & -5 - 2i \end{bmatrix},$$

$$Y_3 = \begin{bmatrix} -13 + 7i & 14 + 3i & 2 + i \\ -3 + 9i & 8 + 3i & -1 + 6i \\ -15 - i & -6 - 4i & 5 + 4i \end{bmatrix},$$

$$Y_4 = \begin{bmatrix} -2 - 14i & 9 + 11i & -9 - 17i \\ -9 - 13i & 2 + 13i & 9 + 8i \\ 10 - 13i & -2 - 6i & 7 - 14i \end{bmatrix}.$$

In this example, the initial iterative matrices are taken as $Y_i(0) = 10 \times I_3, i \in \mathbb{I}[1,4]$, and the relative iterative error is defined as

$$ERR = \frac{\sqrt{\begin{array}{c}\|Y_1 - Y_1(k)\|^2 + \|Y_2 - Y_2(k)\|^2 \\ +\|Y_3 - Y_3(k)\|^2 + \|Y_4 - Y_4(k)\|^2\end{array}}}{\sqrt{\|Y_1\|^2 + \|Y_2\|^2 + \|Y_3\|^2 + \|Y_4\|^2}}, \tag{81}$$

where $Y_i(k) (i \in \mathbb{I}[1,4])$ stand for the kth iteration solution. The iteration is terminated if the relative iterative error is less than δ or the number of the prescribed iteration step $k_{\max} = 30000$ is exceeded. Here, δ is a positive number.

Table 1. Iterative number of the RGI algorithm with different μ for Example 5.1.

μ	$\mu_1 = 5.1499e-06$	$\mu_2 = 5.2559e-06$	$\mu_3 = 5.0499e-06$	$\mu_{\text{opt}} = 5.2499e-06$
IT	2184	3000	2227	2142

By some calculations, we find that Example 5.1 satisfies the condition of Theorem 4.1. Then the optimal parameter of the RGI algorithm is obtained as $\mu = 5.2559e-06$ when relaxation factors are chosen as $\omega_1 = 0.25, \omega_2 = 0.52, \omega_3 = 0.32, \omega_4 = 0.48$. However, there are errors in the experiment, and the convergence rate of the RGI method is not the fastest if μ is chosen to be 5.2559e−06. Thus, we try to find the optimal experimental parameter near the value $\mu = 5.2559e-06$. In Figure 1, we compare the convergence performance of the RGI algorithm under $\mu_1 = 5.1499e-06$, $\mu_2 = 5.2559e-06$, $\mu_3 = 5.0499e-06$ and $\mu_{\text{opt}} = 5.2499e-06$, respectively. As shown in Figure 1, if the convergence parameter μ is selected as different values, the convergence curve also has corresponding change. In order to more intuitively observe the performance of the RGI algorithm under different convergence parameters, we list the IT of the RGI algorithm in Table 1. It is evident that the convergence performance is the best when parameter μ is chosen to be $\mu_{RGI} = 5.2499e-06$.

Moreover, the RGI algorithm with $\omega_1 = \omega_2 = \omega_3 = \omega_4 = 0.5$ reduces to the GI algorithm. Similarly, we adopt the method of experiment debugging to find the optimal experimental parameter of the GI algorithm. Finally, the IT of the GI algorithm with $\mu_{GI} = 4.5503e-06$ is the least.

In Figures 2–3, we present that the convergence curves of the RGI and GI algorithms with different δ. In this experiment, we compare the convergence curves of two algorithms under the optimal experimental parameters. It follows from Figures 2–3 that the ERR decreases as the IT increases and gradually approaches 0, which indicates that the tested algorithms are effective and convergent. In addition, Figures 2–3 clearly show that the IT of the RGI and GI algorithms are decreasing as the increasing of δ. Besides, we can find that the convergence speed of the RGI algorithm is always faster than the GI one in the above four situations.

In order to more specifically verify the advantages of the RGI algorithm, we list detailed numerical results of the RGI and GI algorithms in Table 2, which includes IT, CPU and ERR. According to Table 2, the IT and CPU of the tested algorithms are gradually increase with the decreasing of the parameter δ. Moreover, it can be seen that the IT and CPU of the RGI method are always less than the GI one. Therefore, we can conclude that the convergence performance of the RGI algorithm proposed in this paper is better than the GI algorithm [23].

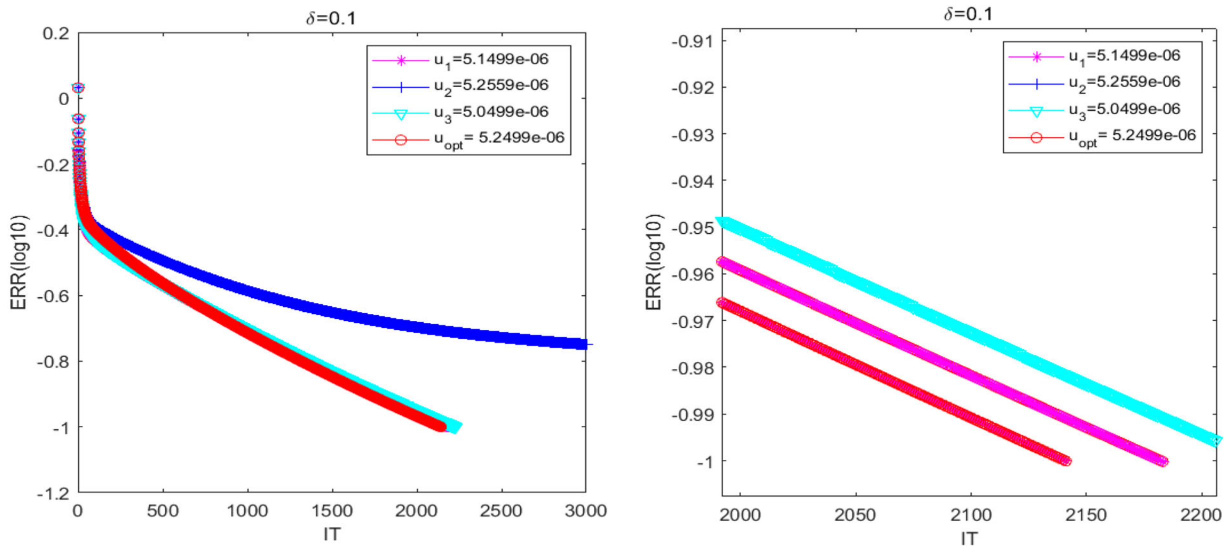

Figure 1. Comparison of convergence performance of RGI with different parameters μ for Example 5.1.

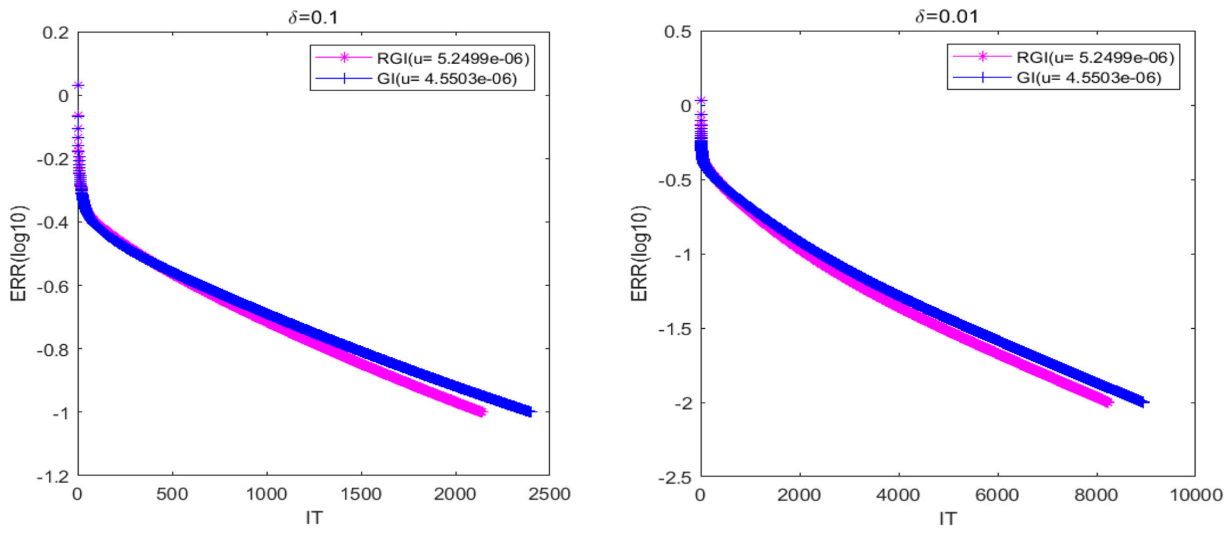

Figure 2. The convergence curves of the tested methods with $\delta = 0.1$ (left) and $\delta = 0.01$ (right) for Example 5.1.

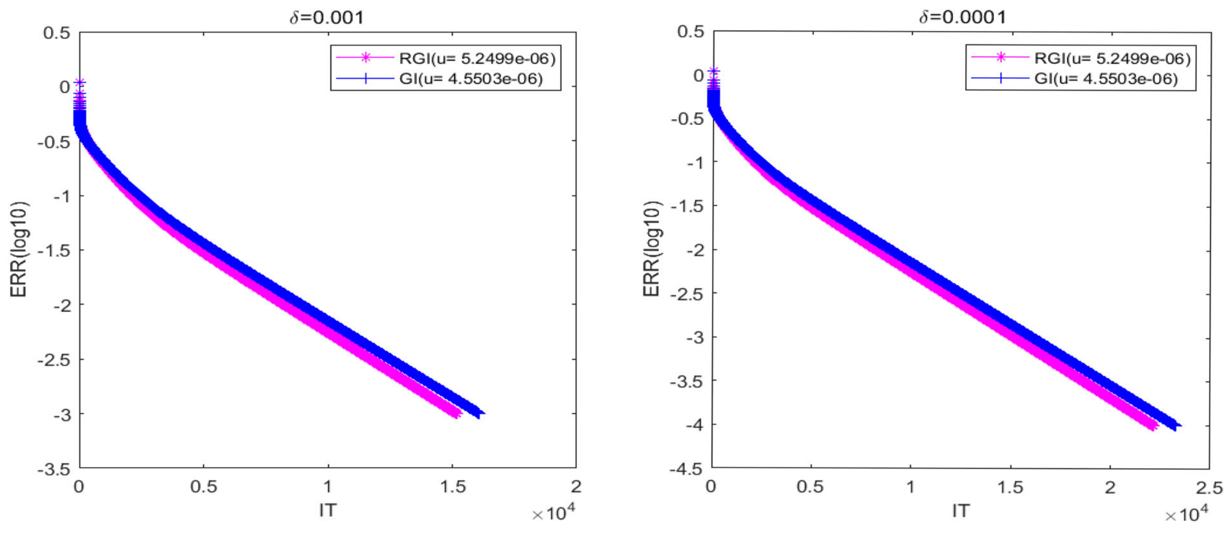

Figure 3. The convergence curves of the tested methods with $\delta = 0.001$ (left) and $\delta = 0.0001$ (right) for Example 5.1.

Table 2. Numerical results of the tested methods with different δ for Example 5.1.

Algorithm	δ	0.1	0.01	0.001	0.0001
GI [23]	IT	2403	8937	16093	23252
$\mu_{\text{GI}} = 4.5503e-06$	CPU	2.2139	8.4308	14.2259	20.4224
	ERR	0.0997	0.01	9.9968e-04	9.9996e-05
RGI	IT	2142	8238	15189	22151
$\mu_{\text{RGI}} = 5.2499e-06$	CPU	1.9771	7.5984	11.5423	18.4224
	ERR	0.0995	0.01	9.9979e-04	9.9999e-05

Example 5.2: We consider the generalized coupled complex conjugate and transpose Sylvester matrix equation (1) in the special case of $p = q = 2$, and it has

$$\begin{cases} A_{11}Y_1B_{11} + C_{11}\overline{Y_1}D_{11} \\ \quad +E_{12}Y_2^TF_{12} + G_{12}Y_2^HH_{12} = M_1, \\ A_{21}Y_1B_{21} + C_{21}\overline{Y_1}D_{21} \\ \quad +E_{22}Y_2^TF_{22} + G_{22}Y_2^HH_{22} = M_2, \end{cases} \tag{82}$$

with the following parametric matrices

$$A_{11} = \begin{bmatrix} 3 & -5 & -6 \\ -1+7i & -5+3i & 1 \\ -2+7i & -9+6i & 0 \end{bmatrix},$$

$$B_{11} = \begin{bmatrix} -3+8i & 2-9i & -3 \\ -1+4i & -3+11i & 10 \\ 1+8i & 2+6i & -3+1i \end{bmatrix},$$

$$A_{21} = \begin{bmatrix} 5-3i & 1 & 14-3i \\ 1-6i & -1+11i & 1+14i \\ 7 & 9-5i & 1+3i \end{bmatrix},$$

$$B_{21} = \begin{bmatrix} -1 & 3 & 5 \\ 1 & -3 & 1 \\ 1 & 2 & -3 \end{bmatrix},$$

$$C_{11} = \begin{bmatrix} 14+2i & 3+5i & 4-1i \\ -6-6i & -8-1i & 5+1i \\ 2+3i & -8+2i & 1-9i \end{bmatrix},$$

$$D_{11} = \begin{bmatrix} -13+1i & -11+6i & -11-2i \\ 2+7i & -10+4i & -10+1i \\ -2+5i & 12-18i & 1-9i \end{bmatrix},$$

$$C_{21} = \begin{bmatrix} 8-2i & -8-2i & -5i \\ 2+7i & -10+4i & -10+1i \\ -2+5i & 12-18i & 1-9i \end{bmatrix},$$

$$D_{21} = \begin{bmatrix} -12+1i & -10+6i & -20 \\ 2+7i & -10+4i & -10+1i \\ -2+5i & 12-18i & 1-9i \end{bmatrix},$$

$$E_{12} = \begin{bmatrix} 1+8i & 6-2i & -8+17i \\ 4+2i & 10+9i & 2+9i \\ -5-3i & 4+10i & 10+9i \end{bmatrix},$$

$$F_{12} = \begin{bmatrix} 11+2i & 2-8i & 2-3i \\ 7+7i & 9-1i & 1+5i \\ 7+10i & 5+8i & 7-8i \end{bmatrix},$$

$$E_{22} = \begin{bmatrix} 1 & -i & 1 \\ i & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix}, \quad F_{22} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & -3 \end{bmatrix},$$

$$G_{12} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 4 & 1 \\ -2 & 0 & -2 \end{bmatrix}, \quad H_{12} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

$$G_{22} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 4 & 1 \\ -2 & 0 & -2 \end{bmatrix}, \quad H_{12} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & -i & 1 \\ -1 & 2 & 1 \end{bmatrix},$$

$$M_1 = \begin{bmatrix} -4792+2166i & 2299+5490i & 3353-3607i \\ 4679+3574i & -3507-629i & 1933-1090i \\ -2720-2137i & -4833+2642i & 2164+163i \end{bmatrix},$$

$$M_2 = \begin{bmatrix} 562+1926i & 356+452i & 1636+492i \\ -1858+1250i & 1113+610i & 2678+3267i \\ -511+863i & 677-3228i & 346-3251i \end{bmatrix}.$$

It has the exact solution

$$Y_1 = \begin{bmatrix} 2+2i & 2-2i & 1 \\ 1 & -1 & 2i \\ 2-2i & 2+2i & -1 \end{bmatrix},$$

$$Y_2 = \begin{bmatrix} 16+16i & 20-28i & 12+4i \\ 29-4i & -7+5i & -4+17i \\ 10-19i & 9+11i & -1-2i \end{bmatrix}.$$

The initial iterative matrices are taken to be $Y_i(0) = 10^{-6} \times I_3, i \in \mathbb{I}[1,2]$. Then, we denote the relative iterative error by

$$ERR = \frac{\sqrt{\|Y_1 - Y_1(k)\|^2 + \|Y_2 - Y_2(k)\|^2}}{\sqrt{\|Y_1\|^2 + \|Y_2\|^2}}. \tag{83}$$

In this example, all runs are stopped once ERR is less than ξ or k reaches the maximal iterative steps $k_{\max} =$ 50,000. Here, ξ is a positive number.

For Example 5.2, we also compare the convergence performance of the RGI and GI algorithms. The optimal convergence parameters involved in two algorithms are determined by the following method. If relaxation factors are selected as $\omega_1 = 0.07, \omega_2 = 0.18$, the optimal convergence factor of the RGI algorithm is adopted as $\mu_{RGI} = 1.0821e-04$ by Theorem 4.1. Moreover, the RGI algorithm with $\omega_1 = \omega_2 = 0.5$ reduce to the GI algorithm. By some calculations, the best convergence parameter of the GI algorithm is $\mu_{GI} = 4.5361e-05$.

In Figures 4–5, we plot the graphs of ERR(log10) versus the IT of the RGI and GI algorithms with different ξ. According to the convergence curves, we observe

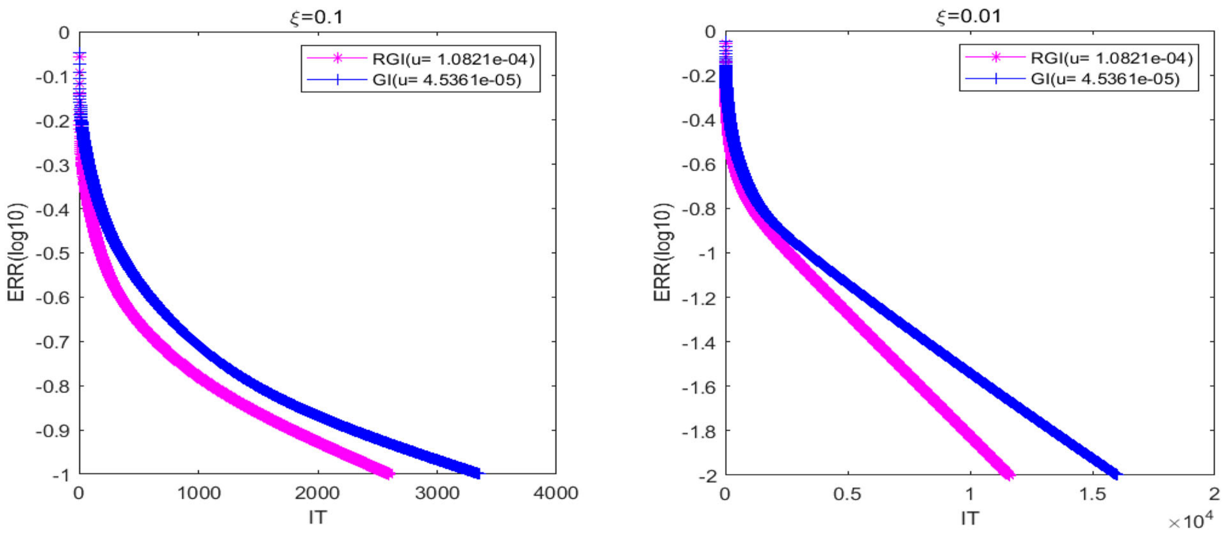

Figure 4. The convergence curves of the tested methods with $\xi = 0.1$ (left) and $\xi = 0.01$ (right) for Example 5.2.

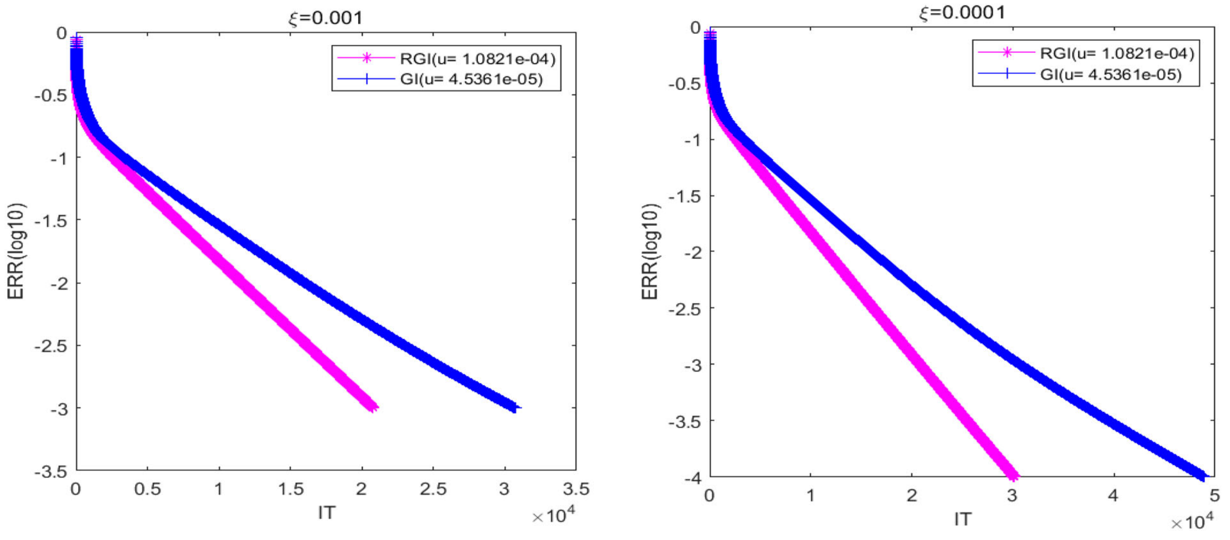

Figure 5. The convergence curves of the tested methods with $\xi = 0.001$ (left) and $\xi = 0.0001$ (right) for of Example 5.2.

that the two algorithms are both convergent and efficient. It is obvious that the convergence rate of the RGI method ($\omega_1 = 0.07, \omega_2 = 0.18$) is always faster than GI one ($\omega_1 = \omega_2 = 0.5$) for the four cases of ξ. In addition, it follows from Figures 3–4 that the IT and CPU of the tested algorithms are increasing with the decreasing of ξ. In particular, the convergence advantage of the RGI algorithm is more obvious when ξ is smaller. The results illustrate that the RGI algorithm is superior to the GI algorithm if the relaxation parameters are chosen appropriately.

In order to further verify the advantages of the proposed algorithm, we clearly report the numerical results of the RGI and GI methods for Example 5.2 in Table 3. From Table 3, it is easy to discover that the IT of the algorithms is increasing with the decreasing of relative error. Furthermore, the IT and CPU of the RGI method are less than those of the GI one. As a whole, the proposed algorithm has better convergence behaviours than the GI method. This means that the relaxation technique can effectively improve the convergence rate of the GI algorithm.

Table 3. Numerical results of the tested methods with different ξ for Example 5.2.

Algorithm	ξ	0.1	0.01	0.001	0.0001
GI [23]	IT	3360	16003	30706	48950
$\mu_{\text{GI}} = 4.5361e-05$	CPU	2.9696	13.7964	19.5881	41.3501
	ERR	0.1	0.01	9.9998e-04	9.9995e-05
RGI	IT	2594	11590	20780	30138
$\mu_{\text{RGI}} = 1.0821e-04$	CPU	2.2231	10.0329	15.4115	26.0794
	ERR	0.0995	0.01	9.9999e-04	9.9982e-05

6. Concluding remarks

In this paper, by adopting the relaxation technique into the GI algorithm, we establish the relaxed gradient-based iterative (RGI) algorithm to solve the generalized coupled complex conjugate and transpose Sylvester matrix equations. The main idea of the algorithm is introducing relaxation parameter to control the weights of iterative sequences. Applying straighten operation and real representation of complex matrices, we derive the necessary and sufficient condition for convergence of the RGI algorithm. Besides, the optimal convergence parameter and some related conclusions are given. To overcome high-dimensional computational problems, we propose sufficient condition for convergence with smaller computational complexity. Finally, numerical experiments verify that the RGI algorithm has more excellent convergence performance than the GI one.

Note that in our experiment, the relaxation factors ω_l ($l \in \mathbb{I}[1, q]$) are obtained through experimental debugging. The selection criteria for the optimal relaxation factors are not provided. The future research direction is to further develop the theory of selecting the optimal relaxation factor. Besides, the value of the convergence parameter μ in the RGI algorithm is fixed. To optimize the convergence performance of the RGI algorithm, we will consider to introduce different step size factors into the RGI algorithm.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is supported by the Guangxi Natural Science Foundations [No. 2021GXNSFBA196064, Guike AD21220129], the National Science Foundation of China [No. 11901123], the Guangxi Natural Science Foundations [2019GXNSFBA 185014, Guike AD20159056], the Natural Science Foundation of Guangxi University for Nationalities [No. 2019KJ QN001].

References

[1] Hajarian M. Computing symmetric solutions of general Sylvester matrix equations via Lanczos version of biconjugate residual algorithm. Comput Math Appl. 2018;76:686–700. doi: 10.1016/j.camwa.2018.05.010

[2] Hajarian M. Developing CGNE algorithm for the periodic discrete-time generalized coupled Sylvester matrix equations. Comput Appl Math. 2015;34:755–771. doi: 10.1007/s40314-014-0138-7

[3] Zhou Y-H, Zhang X, Ding F. Partially-coupled nonlinear parameter optimization algorithm for a class of multivariate hybrid models. Appl Math Comput. 2022;414:Article ID 126663.

[4] Dehghan M, Hajarian M. The generalised Sylvester matrix equations over the generalized bisymmetric and skew-symmetric matrices. Int J Syst Sci. 2012; 43:1580–1590. doi: 10.1080/00207721.2010.549584

[5] Zhou B, Wei X-Z, Duan G-R. Stability and stabilization of discrete-time periodic linear systems with actuator saturation. Automatica. 2011;47:1813–1820. doi: 10.1016/j.automatica.2011.04.015

[6] Zhou B, Duan G-R. Periodic Lyapunov equation based approaches to the stabilization of continuous-time periodic linear systems. IEEE Trans Automat Contr. 2011;57:2139–2146. doi: 10.1109/TAC.2011.2181796

[7] Ding F, Wang F. Decomposition based least squares iterative identification algorithm for multivariate pseudo-linear ARMA systems using the data filtering. J Franklin Inst. 2017;354:1321–1339. doi: 10.1016/j.jfranklin.2016.11.030

[8] Shen H-L, Peng C, Zhang T. Gradient based iterative solutions for Sylvester conjugate matrix equations. J Math Res Appl. 2017;03:103–118.

[9] Li X, Ding J, Ding F. Gradient based iterative solutions for general linear matrix equations. Comput Math Appl. 2009;58:1441–1448. doi: 10.1016/j.camwa.2009.06.047

[10] Li X, Liu Y-J, Yang H-Z. Gradient based and least squares based iterative algorithms for matrix equations $AXB + CX^TD = F$. Appl Math Comput. 2010;217: 2191–2199.

[11] Bai Z-Z, Guo X-X, Yin J-F. On two iteration methods for the quadratic matrix equations. Int J Numer Anal Model. 2005;2:114–122.

[12] Chen Z-B, Chen X-S. Modification on the convergence results of the Sylvester matrix equation $AX + XB = C$. J Franklin Inst. 2022;359:3126–3147. doi: 10.1016/j.jfranklin.2022.02.021

[13] Xu L, Ding F, Zhu Q-M. Separable synchronous multi-innovation gradient-based iterative signal modeling from on-line measurements. IEEE Trans Instrum Meas. 2022;71:1–13.

[14] Ding F. Least squares parameter estimation and multi-innovation least squares methods for linear fitting problems from noisy data. J Comput Appl Math. 2023;426: Article ID 115107. doi: 10.1016/j.cam.2023.115107

[15] Ding J, Liu Y-J, Ding F. Iterative solutions to matrix equations of the form $A_iXB_i = F_i$. Comput Math Appl. 2010;59:3500–3507. doi: 10.1016/j.camwa.2010.03.041

[16] Ding F, Ding J. Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle. Appl Math Comput. 2008;197:41–50.

[17] Ding F, Chen T-W. Gradient based iterative algorithms for solving a class of matrix equations. IEEE Trans Automat Contr. 2005;50:1216–1221. doi: 10.1109/TAC.2005.852558

[18] Ding F, Chen T-W. On iterative solutions of general coupled matrix equations. SIAM J Control Optim. 2006;44:2269–2284. doi: 10.1137/S0363012904441350

[19] Ding F, Chen T-W. Iterative least-squares solutions of coupled Sylvester matrix equations. Syst Control Lett. 2005;54:95–107. doi: 10.1016/j.sysconle.2004.06.008

[20] Wu A-G, Zeng X-L, Duan G-R, et al. Iterative solutions to the extended Sylvester-conjugate matrix equations. Appl Math Comput. 2010;217:130–142.

[21] Wu A-G, Feng G, Duan G-R, et al. Iterative solutions to coupled Sylvester-conjugate matrix equations. Comput Math Appl. 2010;60:54–66. doi: 10.1016/j.camwa.2010.04.029

[22] Song C-Q, Chen G-L, Zhao L-L. Iterative solutions to coupled Sylvester-transpose matrix equations. Appl Math Model. 2011;35:4675–4683. doi: 10.1016/j.apm.2011.03.038

[23] Beik FPA, Mahmoud MM. Gradient-based iterative algorithm for solving the generalized coupled Sylvester-transpose and conjugate matrix equations over reflexive (anti-reflexive) matrices. Trans Inst Meas Control. 2014;36:99–110. doi: 10.1177/0142331213482485

[24] Lv L-L, Chen J-B, Zhang L, et al. Gradient-based neural networks for solving periodic Sylvester matrix equations. J Franklin Inst. 2022;359:10849–10866. doi: 10.1016/j.jfranklin.2022.05.023

[25] Li S-H, Ma C-F. Factor gradient iterative algorithm for solving a class of discrete periodic Sylvester matrix

equations. J Franklin Inst. 2022;359:9952–9970. doi: 10.1016/j.jfranklin.2022.09.041

[26] Fan W, Gu C-Q, Tian Z-L. Jacobi-gradient iterative algorithms for Sylvester matrix equations. In: Linear algebra society topics. Shanghai, China: Shanghai University; 2007. p. 16–20.

[27] Niu Q, Wang X, Lu L-Z. A relaxed gradient based algorithm for solving Sylvester equations. Asian J Control. 2011;13:461–464. doi: 10.1002/asjc.v13.3

[28] Huang B-H, Ma C-F. The relaxed gradient-based iterative algorithms for a class of generalized coupled Sylvester-conjugate matrix equations. J Franklin Inst. 2018;355:3168–3195. doi: 10.1016/j.jfranklin.2018.02.014

[29] Huang B-H, Ma C-F. On the relaxed gradient-based iterative methods for the generalized coupled Sylvester-transpose matrix equations. J Franklin Inst. 2022;359:10688–10725. doi: 10.1016/j.jfranklin.2022.07.051

[30] Wang W-L, Song C-Q, Ji S-P. Iterative solution to a class of complex matrix equations and its application in time-varying linear system. J Appl Math Comput. 2021;67:317–341. doi: 10.1007/s12190-020-01486-6

[31] Ding F. Hierarchical multi-innovation stochastic gradient algorithm for Hammerstein nonlinear system modeling. Appl Math Model. 2013;37:1694–1704. doi: 10.1016/j.apm.2012.04.039

[32] Wu A-G, Zhang Y, Qian Y-Y. Complex conjugate matrix equations. Beijing: Science Press; 2017.

[33] Ding F, Chen T-W. Hierarchical gradient-based identification of multivariable discrete-time systems. Automatica. 2005;41:315–325. doi: 10.1016/j.automatica.2004.10.010

[34] Zhang H-M. A finite iterative algorithm for solving the complex generalized coupled Sylvester matrix equations by using the linear operators. J Franklin Inst. 2017;354:1856–1874. doi: 10.1016/j.jfranklin.2016.12.011