The effective and modified Jacobi gradient based iterative algorithms for the discrete-time periodic Sylvester matrix equations ☆

Xiaowen Wu, Zhengge Huang*

School of Mathematical Sciences, Center for Applied Mathematics of Guangxi, Guangxi Minzu University, 530006, Nanning, PR China

Abstract

In this paper, we discuss the new convergence properties of some gradient based iterative (GI) algorithms and propose two new GI-like algorithms for solving the discrete-time periodic Sylvester (DTPS) matrix equations and its generalized version, which often arise in the fields of physics, medicine and so forth. We first review the Jacobi GI (JGI) and accelerated JGI (AJGI) algorithms (Appl. Numer. Math., 168 (2021) 251-273) for the DTPS matrix equations, and establish the new and correct convergence conditions of these two algorithms. Then we apply a new update strategy to the JGI algorithm and develop the effective Jacobi gradient based iterative (EJGI) algorithm for solving the DTPS matrix equations, which is different from the AJGI one. Furthermore, based on the ideas of the JGI and the Gauss-Seidel (G-S) algorithms, we construct the modified Jacobi gradient based iterative (MJGI) algorithm for the generalized discrete-time periodic Sylvester (GDTPS) matrix equations. Compared with the JGI algorithm, the MJGI algorithm can make full use of the latest information to compute the next result and lead to a faster convergence rate. By utilizing the properties of the matrix norms, Kronecker product and techniques of inequalities, we prove that two proposed iterative algorithms are convergent under proper restrictions. Finally, some numerical examples are given to validate the efficiencies and advantages of the proposed EJGI and MJGI algorithms for DTPS and GDTPS matrix equations.

Keywords: DTPS matrix equations, GDTPS matrix equations, EJGI algorithm, MJGI algorithm, convergence properties

1. Introduction

In this paper, we aim to compute the numerical solution of the following discrete-time periodic Sylvester (DTPS) matrix equations

$$A_jY_j + Y_{j+1}B_j = C_j, j = 1, 2, \dots, \gamma, \tag{1}$$

where the known matrices $A_j \in \mathbb{R}^{m\times m}, B_j \in \mathbb{R}^{n\times n}, C_j \in \mathbb{R}^{m\times n}$ and the unknown matrices $Y_j \in \mathbb{R}^{m\times n}$ are periodic with period γ, i.e., $A_{j+\gamma} = A_j, B_{j+\gamma} = B_j, C_{j+\gamma} = C_j$ and $Y_{j+\gamma} = Y_j$ for $j = 1, \dots, \gamma$.

Linear discrete periodic systems are widely used in the fields of physics, biology, medicine and many other engineering fields [1–7]. For example, the following forward and backward periodic Sylvester matrix equations (PSMEs)

$$A_iY_jB_j + C_jY_{j+1}D_j = F_j, \tag{2}$$

and

$$A_jY_{j+1}B_i + C_jY_jD_j = F_j, \tag{3}$$

with $j = 1, 2, \dots, \gamma$, and Y_j being the unknown matrices, are an indispensable part of pole assignment and the design of state observers for linear discrete periodic systems [8]. The forward PSME (2) is more general than (1), and contains the DTPS matrix equation (1) as a special case. Up to now, a lot of efficient methods have been proposed to solve various types of periodic matrix equations due to the universal existence and significance of this kind of matrix equations. For instance, Varga [9] designed some efficient and numerically reliable algorithms for solving periodic Lyapunov matrix equations based on the periodic Schur decomposition. Based on the conjugate gradient normal equation error (CGNE) method, Hajarian [10] presented an iterative algorithm for solving the general coupled discrete-time periodic matrix equations. And the same author proposed the matrix form of the biconjugate

☆This work was supported by the National Natural Science Foundation of China (No. 12361078), and the Guangxi Natural Science Foundation (No. 2024GXNSFAA010498).

*Corresponding author.

Email addresses: xiaowenwu1998@163.com (Xiaowen Wu), ZhenggeHuang@163.com (Zhengge Huang*)

residual (BCR) algorithm for solving the forward PSME (2) in [11]. Note that the general periodic matrix equation is also a kind of important matrix equation, which has important applications in many fields. Recently, Lv et al. [12] constructed a finite iterative method for solving it. Subsequently, Hajarian [13] presented three types of BCR method to find the generalized bisymmetric periodic solutions of general periodic matrix equations. And the same author in [14] designed four new iterative methods based on the CGNE, conjugate gradient normal equation residual (CGNR), and least-squares QR factorization (LSQR) algorithms to compute the reflexive periodic solutions of the general periodic matrix equations.

Apart from the periodic matrix equations, there are many other linear matrix equations arising from many fields of science and engineering, and playing a very significant role in various branches of them. Due to this fact, in the past few decades, many researchers have devoted themselves to deriving a great deal of different methods to solve these matrix equations, including the conjugate gradient iterative method [15], Newton method [16], parametric iterative algorithms [17, 18], gradient based iterative (GI) algorithms [19, 20] and so on. In addition, Li and Wu [21] extended the single-step HSS (SHSS) method for saddle point problems. Yan and Ma [22] designed an iterative algorithm to solve a class of generalized coupled Sylvester-transpose matrix equations over bisymmetric or skew-anti-symmetric matrices. And Wu and Zeng [23] proposed the ADMM-based methods to solve the nearness symmetric solution of the system of matrix equations $A_1XB_1 = C_1$ and $A_2XB_2 = C_2$ recently. Besides, Wang and Song [24] proposed a new BCR algorithm to compute the constraint solution of the coupled operator equations. In [25–27], Shirilord and Dehghan constructed the efficient iteration methods for three different matrix equations, and they also designed a stationary Landweber method with momentum acceleration in [28]. Also, Huang and Cui [29] developed the modified and accelerated relaxed gradient-based iterative algorithms for the complex conjugate and transpose matrix equations. What is more, Hajarian [30] established the matrix form of the BCR algorithm for computing the generalized reflexive and anti-reflexive solutions of the generalized Sylvester matrix equation, then the same author generalized the Lanczos version of BCR algorithm to compute the symmetric solutions of the general Sylvester matrix equations in [31]. In [32], Zhang established the GI algorithm for solving the extended coupled Sylvester matrix equations $A_1XB_1 + A_2YB_2 = F_1, C_1XD_1 + C_2YD_2 = F_2$ by using the hierarchical identification principle. And Xie and Ma [33] derived the accelerated GI (AGI) algorithm to solve the generalized Sylvester-transpose matrix equation $AXB + CX^TD = F$ by taking advantage of information generated in the previous half-step and introducing a relaxation factor.

For the generalized coupled Sylvester matrix equation

$$A_{l1}X_1B_{l1} + A_{l2}X_2B_{i2} + \cdots + A_{lq}X_qB_{lq} = G_l, l = 1, 2, \cdots, p, \tag{4}$$

with X_t $(t = 1, 2, \ldots, q)$ being the unknown matrices that need to be determined, Zhang [34] developed the residual norm steepest descent (RNSD), conjugate gradient normal equation (CGNE) and biconjugate gradient stabilized (Bi-CGSTAB) algorithms to solve (4). Subsequently, by constructing an objective function and using the gradient search, Zhang [35] constructed the full-rank and reduced-rank gradient-based algorithms for solving the matrix equation (4).

In addition, the generalized coupled Sylvester-conjugate matrix equation

$$E_{l1}X_1F_{l1} + G_{l1}\overline{X_1}H_{l1} + \cdots + E_{lq}X_qF_{lq} + G_{lq}\overline{X_q}H_{lq} = W_l, l = 1, 2, \ldots, p, \tag{5}$$

with X_t $(t = 1, 2, \ldots, q)$ being the indeterminate matrices, is the general version of (4). When $G_{lt} = 0$ and $H_{lt} = 0$ $(l = 1, \ldots, p; t = 1, \ldots, q)$, (5) reduces to (4). For the matrix equation (5), Huang and Ma [36] introduced l relaxation factors into the GI algorithm and derived two relaxed GI (RGI) algorithm. And they proved the convergence of the RGI algorithms by utilizing the properties of the real representation of a complex matrix. Very recently, Wang and Song [37] constructed a modified RGI (MRGI) algorithm to solve the coupled Sylvester-conjugate matrix equation (5). Then Wang et al. [38] developed a cyclic GI (CGI) algorithm by introducing the modular operator, and the most significant improvement of this algorithm is that less information is used during each iteration update.

As mentioned before, the DTPS and the GDTPS matrix equations arise widely in scientific and engineering fields. Thus it is meaningful to design efficient algorithms for solving these two kinds of matrix equations. Based on this fact, in this work, we aim to construct some new and efficient algorithms to compute the iterative solutions of the DTPS and the GDTPS matrix equations. We first review the Jacobi GI (JGI) and the accelerated JGI (AJGI) algorithms for the DTPS matrix equations in [39], and find that their convergence proofs are not correct and can be improved. Then we establish the new convergence theorems of the JGI and AJGI algorithm by using the properties of the vector stretching operator, matrix norm and Kronecker product of two matrices. Besides, to further improve the convergence rates of the JGI and AJGI algorithms in [39], we apply a new update strategy to the JGI algorithm [39], and then construct the effective Jacobi gradient based iterative (EJGI) algorithm for the DTPS matrix equations, which is different from the AJGI one in [39]. Numerical experiments show that the proposed EJGI algorithm is more efficient than the GI [40], JGI and AJGI ones [39]. Also, we consider the iterative solutions of the GDTPS matrix equations. Based on the JGI algorithm [39], we propose the modified Jacobi gradient based iterative (MJGI) algorithm for the GDTPS matrix equations by combining the idea of the Jacobi method with the update strategy. It is noteworthy that this idea stems from [41].

The main contributions of this paper are as follows:

- Propose the new convergence conditions of the JGI and the AJGI algorithms in [39], which correct and improve the existing ones in [39].
- Apply a new update technique to the JGI algorithm in [39] and establish the EJGI algorithm, which is different from the AJGI one and has higher computational efficiency than the AJGI one.
- By combining the idea of Jacobi algorithm and the update strategy, we design the MJGI algorithm for the GDTPS matrix equations, and derive the sufficient and necessary condition for the convergence of the MJGI algorithm. Compared with the JGI algorithm, the proposed MJGI algorithm can use the latest results to compute the next results and has higher computational efficiency. Additionally, the MJGI algorithm requires less computational complexity than the factor gradient iterative (FGI) one in [41].

The remainder of this paper is organized as follows. In Section 2, we list some useful notations, definitions and lemmas that will be used throughout this paper. In Section 3, we review the JGI and AJGI algorithms proposed in [39] for the DTPS matrix equations (1) and establish their new convergence conditions. In Section 4, we construct the EJGI algorithm for the DTPS matrix equations (1) and analyze its convergence. Additionally, we derive a new algorithm referred to as the MJGI algorithm for the GDTPS matrix equations and investigate its convergence property in Section 5. In Section 6, several numerical examples are given to illustrate the effectivenesses and advantages of the proposed EJGI and MJGI algorithms. Lastly, some conclusions and outlooks are given to end this paper in Section 7.

2. Preliminaries

In this section, we list some notations, definitions and lemmas, which will be used in the subsequent sections.

Let $\mathbb{R}^{n\times n}$ and $\mathbb{C}^{n\times n}$ be the sets of all $n\times n$ real matrices and all $n\times n$ complex matrices, respectively. For a given matrix $B\in\mathbb{R}^{n\times n}$, the notations B^{-1}, B^T and $\rho(B)$ stand for the inverse, the transpose and the spectral radius of B, respectively. If B is a square matrix, then $\mathrm{tr}(B)$ stands for the trace of B. The 2-norm and Frobenius norm of B are denoted by $\|B\|_2=\sqrt{\rho(B^TB)}$ and $\|B\|=\sqrt{\mathrm{tr}(B^TB)}$, respectively. Let $B=D+R$, with D and R being the diagonal and non-diagonal parts of the matrix B, respectively.

In addition, we present several useful definitions below.

Definition 2.1. *[42] For two matrices $F=(f_{ij})\in\mathbb{C}^{m\times n}$ and $G=(g_{ij})\in\mathbb{C}^{k\times l}$, the Kronecker product of F and G is defined as*

$$F\otimes G=\begin{pmatrix} f_{11}G & f_{12}G & \cdots & f_{1n}G\\ f_{21}G & f_{22}G & \cdots & f_{2n}G\\ \vdots & \vdots & & \vdots\\ f_{m1}G & f_{m2}G & \cdots & f_{mn}G\end{pmatrix}=[f_{ij}G]_{m\times n}\in\mathbb{C}^{mk\times nl}. \tag{6}$$

Definition 2.2. *[43] Let e_{ks} be the s-dimensional column vector whose k-th element of e_{ks} is 1 and other elements are 0. Then the vec-permutation matrix $P(m,n)$ is defined as*

$$P(t,s):=\begin{pmatrix} I_t\otimes e_{1s}^T\\ I_t\otimes e_{2s}^T\\ \vdots\\ I_t\otimes e_{ss}^T\end{pmatrix}. \tag{7}$$

Definition 2.3. *[42] Let $G=[g_1,g_2,\cdots,g_s]\in\mathbb{C}^{t\times s}$ with g_k being the k-th column of G. The vector stretching function of G is defined as*

$$\mathrm{vec}(G)=[g_1^T,g_2^T,\cdots,g_s^T]^T\in\mathbb{C}^{ts}. \tag{8}$$

Next, some significant lemmas are reviewed in the following.

Lemma 2.1. *[42] Let $F\in\mathbb{C}^{m\times q}$, $G\in\mathbb{C}^{s\times t}$ and $Y\in\mathbb{C}^{q\times s}$, then*
(1) $\mathrm{vec}(FYG)=(G^T\otimes F)\mathrm{vec}(Y)$;
(2) $\mathrm{vec}(Y^T)=P(q,s)\mathrm{vec}(Y)$.

Lemma 2.2. *[44] Consider the matrix equation $AYB = F$, where $A \in \mathbb{R}^{m\times r}$, $B \in \mathbb{R}^{s\times n}$ and $F \in \mathbb{R}^{m\times n}$ are known matrices, and $Y \in \mathbb{R}^{r\times s}$ needs to be determined. For this matrix equation, an iterative algorithm is constructed as*

$$Y(l+1) = Y(l) + \mu A^T(F - AY(l)B)B^T, \tag{9}$$

with

$$0 < \mu < \frac{2}{\|A\|_2^2\|B\|_2^2}. \tag{10}$$

If this matrix equation has a unique solution Y_*, then the iterative solution $Y(l)$ converges to the unique solution Y_*, that is $\lim_{l\to\infty} Y(l) = Y_*$.

3. New convergence analyses of the JGI and the AJGI algorithms

In this section, we first review the Jacobi gradient based iterative (JGI) and the accelerated Jacobi gradient based iterative (AJGI) algorithms established in [39] for the DTPS matrix equation (1), then some errors in the proofs of Theorem 3.2 and Theorem 3.3 in [39] are pointed out. At last, we deduce the new convergent properties of the JGI and the AJGI algorithms, which correct and improve those in [39].

Based on the Jacobi iterative algorithm and the hierarchical identification principle, the JGI and the AJGI algorithms have been proposed for solving the DTPS matrix equation (1) in [39].

In [39], the coefficient matrices A_j, B_j $(j = 1, \ldots, \gamma)$ of the DTPS matrix equation (1) are decomposed into the following forms:

$$A_j = D_{1,j} + R_{1,j}, \tag{11}$$

$$B_j = D_{2,j} + R_{2,j}, \tag{12}$$

where $D_{1,j}$ and $D_{2,j}$ are the diagonal parts of A_j and B_j, respectively.

Define

$$A_0 = A_\gamma, B_0 = B_\gamma, D_{1,0} = D_{1,\gamma}, D_{2,0} = D_{2,\gamma},$$

then the frameworks of the JGI and the AJGI algorithms are as follows.

Algorithm 3.1. *The Jacobi gradient based iterative (JGI) algorithm [39]:*

Step 1: Input matrices $A_j \in \mathbb{R}^{m\times m}, B_j \in \mathbb{R}^{n\times n}, C_j \in \mathbb{R}^{m\times n}$ for $j = 1, \ldots, \gamma$, and two constants $\mu, \eta > 0$. Choose the initial matrices $Y_j(0) \in \mathbb{R}^{m\times m}$ $(j = 1, \ldots, \gamma)$, and set $l = 0$;

Step 2: Take $Y_{j+\gamma}(0) = Y_j(0), A_{j+\gamma} = A_j, B_{j+\gamma} = B_j, C_{j+\gamma} = C_j, D_{1,j+\gamma} = D_{1,j}$ and $D_{2,j+\gamma} = D_{2,j}$;

Step 3: If $\xi_l = \sqrt{\frac{\sum_{j=1}^{\gamma}\|C_j - A_jY_j(l) - Y_{j+1}(l)B_j\|^2}{\sum_{j=1}^{\gamma}\|C_j - A_jY_j(0) - Y_{j+1}(0)B_j\|^2}} < \eta$, then stop; otherwise, go to Step 4;

Step 4: For $l = 0, 1, 2, \cdots$, and $j = 1, \ldots, \gamma$, calculate

$Y_{1,j}(l+1) = Y_j(l) + \mu D_{1,j}(C_j - A_jY_j(l) - Y_{j+1}(l)B_j),$

$Y_{2,j}(l+1) = Y_j(l) + \mu(C_{j-1} - A_{j-1}Y_{j-1}(l) - Y_j(l)B_{j-1})D_{2,j-1},$

$Y_j(l+1) = \frac{Y_{1,j}(l+1) + Y_{2,j}(l+1)}{2},$

$Y_{j+\gamma}(l+1) = Y_j(l+1).$

Step 5: Set $l := l + 1$ and return to Step 3.

Algorithm 3.2. *The accelerated Jacobi gradient based iterative (AJGI) algorithm [39] :*

Step 1: Input matrices $A_j \in \mathbb{R}^{m\times m}, B_j \in \mathbb{R}^{n\times n}, C_j \in \mathbb{R}^{m\times n}$ for $i = 1, \ldots, \gamma$, and three constants $\mu, \eta > 0$ and $0 < \omega < 1$. Choose the initial matrices $Y_j(0), Y_{2,j}(0) \in \mathbb{R}^{m\times n}$ $(j = 1, \ldots, \gamma)$, and set $l = 0$;

Step 2: Take $Y_{j+\gamma}(0) = Y_j(0), A_{j+\gamma} = A_j, B_{j+\gamma} = B_j, C_{j+\gamma} = C_j, D_{1,j+\gamma} = D_{1,j}$ and $D_{2,j+\gamma} = D_{2,j}$;

Step 3: If $\xi_l = \sqrt{\frac{\sum_{j=1}^{\gamma}\|C_j - A_jY_j(l) - Y_{j+1}(l)B_j\|^2}{\sum_{j=1}^{\gamma}\|C_j - A_jY_j(0) - Y_{j+1}(0)B_j\|^2}} < \eta$, stop; otherwise, go to Step 4;

Step 4: For $l = 0, 1, 2, \cdots$, and $j = 1, \ldots, \gamma$, calculate

$$Y_{1,j}(l+1) = Y_j(l) + \mu\omega D_{1,j}\left(C_j - A_jY_j(l) - Y_{j+1}(l)B_j\right),$$

$$\hat{Y}_j(l) = (1-\omega)Y_{1,j}(l+1) + \omega Y_{2,j}(l),$$

$$\hat{Y}_{j+\gamma}(l) = \hat{Y}_j(l),$$

$$Y_{2,j}(l+1) = \hat{Y}_j(l) + \mu(1-\omega)\left(C_{j-1} - A_{j-1}\hat{Y}_{j-1}(l) - \hat{Y}_j(l)B_{j-1}\right)D_{2,j-1},$$

$$Y_j(l+1) = (1-\omega)Y_{1,j}(l+1) + \omega Y_{2,j}(l+1),$$

$$Y_{j+\gamma}(l+1) = Y_j(l+1).$$

Step 5: Set $l := l+1$ and return to Step 3.

Here, we re-present the function $Z(k+1)$ from the proof of Theorem 3.2 in [39] as follows

$$\begin{aligned} Z(l+1) &\le \sum_{j=1}^{\gamma}\left(\frac{1}{2}\left\|\tilde{Y}_{1,j}(l+1)\right\|^2 + \frac{1}{2}\left\|\tilde{Y}_{2,j}(l+1)\right\|^2\right) \\ &= \sum_{i=1}^{\gamma}\left[\left\|\tilde{Y}_j(l)\right\|^2 - \mu tr\left(\tilde{Y}_j^T(l)D_{1,j}\tilde{\delta}_j(l) + D_{2,j}\tilde{Y}_{j+1}^T(l)\tilde{\delta}_j(l)\right) + \frac{1}{2}\mu^2\left\|D_{1,j}\tilde{\delta}_j(l)\right\|^2 + \frac{1}{2}\mu^2\left\|\tilde{\delta}_j(l)D_{2,j}\right\|^2\right] \\ &\le Z(l) - \mu\sum_{j=1}^{\gamma}\left\|\tilde{\delta}_j(l)\right\|^2 + \frac{1}{2}\mu^2\sum_{j=1}^{\gamma}\left(\|D_{1,j}\|^2 + \|D_{2,j}\|^2\right)\left\|\tilde{\delta}_j(l)\right\|^2, \end{aligned} \tag{13}$$

where $\tilde{\delta}_j = A_j\tilde{Y}_j(l) + \tilde{Y}_{j+1}(l)B_j$.

Based on Equation (3.10) in [39] and Equations (11)-(12), we can get

$$\begin{aligned} \left\|\tilde{\delta}_j(l)\right\|^2 &= tr\left(\tilde{\delta}_j^T(l)\tilde{\delta}_j(l)\right) \\ &= tr\left[\left(\tilde{Y}_j^T(l)A_j^T + B_j^T\tilde{Y}_{j+1}^T(l)\right)\tilde{\delta}_j(l)\right] \\ &= tr\left\{\left[\tilde{Y}_j^T(l)\left(D_{1,j} + R_{1,j}^T\right) + \left(D_{2,j} + R_{2,j}^T\right)\tilde{Y}_{j+1}^T(l)\right]\tilde{\delta}_j(l)\right\} \\ &= tr\left[\left(\tilde{Y}_j^T(l)D_{1,j} + D_{2,j}\tilde{Y}_{j+1}^T(l)\right)\tilde{\delta}_j(l)\right] + tr\left[\left(\tilde{Y}_j^T(l)R_{1,j}^T + R_{2,j}^T\tilde{Y}_{j+1}^T(l)\right)\tilde{\delta}_j(l)\right], \end{aligned} \tag{14}$$

which implies that $\left\|\tilde{\delta}_j(l)\right\|^2 \le tr\left[\left(\tilde{Y}_j^T(l)D_{1,j} + D_{2,j}\tilde{Y}_{j+1}^T(l)\right)\tilde{\delta}_j(l)\right]$ may not be true. The reason is that the sign of $tr\left[\left(\tilde{Y}_j^T(l)R_{1,j}^T + R_{2,j}^T\tilde{Y}_{j+1}^T(l)\right)\tilde{\delta}_j(l)\right]$ is uncertain.

Thus, the derivation of Inequality (13) is not correct. Next we investigate the new convergence condition of the JGI algorithm.

Theorem 3.1. *Assume that the DTPS matrix equation* (1) *is consistent, i.e., the solution of the DTPS matrix equation* (1) *exists. Then the iterative sequences $\{Y_j(l)\}$ $(j = 1, \ldots, \gamma)$ generated by Algorithm 3.1 converge to the unique solution $Y^* = (Y_1^*, Y_2^*, \cdots, Y_\gamma^*)$ for any initial matrices $Y_j(0)$ $(j = 1, \ldots, \gamma)$, if the parameter μ satisfies*

$$\sum_{j=1}^{\gamma}\left(\|I - \mu D_{1,j}A_j\|_2 + \|I - \mu B_{j-1}D_{2,j-1}\|_2 + \mu\|D_{1,j-1}\|_2\|B_{j-1}\|_2 + \mu\|A_j\|_2\|D_{2,j}\|_2\right) < 2. \tag{15}$$

Proof. We first prove that the solution of the DTPS matrix equation (1) is unique. Assume that $\tilde{Y}^* = \left(\tilde{Y}_1^*, \tilde{Y}_2^*, \cdots, \tilde{Y}_\gamma^*\right)$ and $\hat{Y}^* = \left(\hat{Y}_1^*, \hat{Y}_2^*, \cdots, \hat{Y}_\gamma^*\right)$ are two solutions of the DTPS matrix equation (1), then it holds that

$$A_j\tilde{Y}_j^* + \tilde{Y}_{j+1}^*B_j = C_j,\ A_j\hat{Y}_j^* + \hat{Y}_{j+1}^*B_j = C_j,\ j = 1, 2, \ldots, \gamma.$$

It follows from $A_j\tilde{Y}_j^* + \tilde{Y}_{j+1}^*B_j = C_j$ $(j = 1, 2, \ldots, \gamma)$ that

$$\tilde{Y}_j^* = \tilde{Y}_j^* + \mu D_{1,j}\left(C_j - A_j\tilde{Y}_j^* - \tilde{Y}_{j+1}^*B_j\right),\ \tilde{Y}_j^* = \tilde{Y}_j^* + \mu\left(C_{j-1} - A_{j-1}\tilde{Y}_{j-1}^* - \tilde{Y}_j^*B_{j-1}\right)D_{2,j-1},$$

from which one can deduce that

$$\tilde{Y}_j^* = \tilde{Y}_j^* + \frac{\mu}{2} D_{1,j} \left(C_j - A_j \tilde{Y}_j^* - \tilde{Y}_{j+1}^* B_j \right) + \frac{\mu}{2} \left(C_{j-1} - A_{j-1} \tilde{Y}_{j-1}^* - \tilde{Y}_j^* B_{j-1} \right) D_{2,j-1}. \tag{16}$$

In a manner similar to that done for (16), from $A_j \hat{Y}_j^* + \hat{Y}_{j+1}^* B_j = C_j$ $(j = 1, 2, \ldots, \gamma)$, we can derive

$$\hat{Y}_j^* = \hat{Y}_j^* + \frac{\mu}{2} D_{1,j} \left(C_j - A_j \hat{Y}_j^* - \hat{Y}_{j+1}^* B_j \right) + \frac{\mu}{2} \left(C_{j-1} - A_{j-1} \hat{Y}_{j-1}^* - \hat{Y}_j^* B_{j-1} \right) D_{2,j-1}. \tag{17}$$

Subtracting (17) from (16) yields that

$$\begin{aligned} \tilde{Y}_j^* - \hat{Y}_j^* &= \tilde{Y}_j^* - \hat{Y}_j^* - \frac{\mu}{2} D_{1,j} \left[A_j (\tilde{Y}_j^* - \hat{Y}_j^*) + (\tilde{Y}_{j+1}^* - \hat{Y}_{j+1}^*) B_j \right] \\ &\quad - \frac{\mu}{2} \left[A_{j-1} (\tilde{Y}_{j-1}^* - \hat{Y}_{j-1}^*) + (\tilde{Y}_j^* - \hat{Y}_j^*) B_{j-1} \right] D_{2,j-1}, \ j = 1, \ldots, \gamma. \end{aligned} \tag{18}$$

Let $\bar{Y}_j^* = \tilde{Y}_j^* - \hat{Y}_j^*$ $(j = 1, \ldots, \gamma)$, then (18) can be written as

$$\begin{aligned} \bar{Y}_j^* &= \bar{Y}_j^* - \frac{\mu}{2} D_{1,j} \left(A_j \bar{Y}_j^* + \bar{Y}_{j+1}^* B_j \right) - \frac{\mu}{2} \left(A_{j-1} \bar{Y}_{j-1}^* + \bar{Y}_j^* B_{j-1} \right) D_{2,j-1} \\ &= \frac{1}{2} \bar{Y}_j^* - \frac{\mu}{2} D_{1,j} A_j \bar{Y}_j^* - \frac{\mu}{2} D_{1,j} \bar{Y}_{j+1}^* B_j + \frac{1}{2} \bar{Y}_j^* - \frac{\mu}{2} A_{j-1} \bar{Y}_{j-1}^* D_{2,j-1} - \frac{\mu}{2} \bar{Y}_j^* B_{j-1} D_{2,j-1} \\ &= \frac{1}{2} \left(I - \mu D_{1,j} A_j \right) \bar{Y}_j^* + \frac{1}{2} \bar{Y}_j^* \left(I - \mu B_{j-1} D_{2,j-1} \right) - \frac{\mu}{2} D_{1,j} \bar{Y}_{j+1}^* B_j - \frac{\mu}{2} A_{j-1} \bar{Y}_{j-1}^* D_{2,j-1}. \end{aligned} \tag{19}$$

By taking the 2-norm in (19) and using the properties of the matrix norm, we have

$$\begin{aligned} \left\| \bar{Y}_j^* \right\|_2 &= \left\| \frac{1}{2} \left(I - \mu D_{1,j} A_j \right) \bar{Y}_j^* + \frac{1}{2} \bar{Y}_j^* \left(I - \mu B_{j-1} D_{2,j-1} \right) - \frac{1}{2} \mu D_{1,j} \bar{Y}_{j+1}^* B_j - \frac{1}{2} \mu A_{j-1} \bar{Y}_{j-1}^* D_{2,j-1} \right\|_2 \\ &\le \frac{1}{2} \| I - \mu D_{1,j} A_j \|_2 \left\| \bar{Y}_j^* \right\|_2 + \frac{1}{2} \| I - \mu B_{j-1} D_{2,j-1} \|_2 \left\| \bar{Y}_j^* \right\|_2 \\ &\quad + \frac{1}{2} \mu \| D_{1,j} \|_2 \| B_j \|_2 \left\| \bar{Y}_{j+1}^* \right\|_2 + \frac{1}{2} \mu \| A_{j-1} \|_2 \| D_{2,j-1} \|_2 \left\| \bar{Y}_{j-1}^* \right\|_2. \end{aligned} \tag{20}$$

Define $\bar{U}^* = \sum_{j=1}^{\gamma} \left\| \bar{Y}_j^* \right\|_2$, then in view of (20) we deduce that

$$\begin{aligned} \bar{U}^* &= \sum_{j=1}^{\gamma} \left\| \bar{Y}_j^* \right\|_2 \le \sum_{j=1}^{\gamma} \left[\frac{1}{2} \left(\| I - \mu D_{1,j} A_j \|_2 + \| I - \mu B_{j-1} D_{2,j-1} \|_2 \right) \left\| \bar{Y}_j^* \right\|_2 \right] \\ &\quad + \frac{1}{2} \sum_{j=1}^{\gamma} \mu \| D_{1,j} \|_2 \| B_j \|_2 \left\| \bar{Y}_{j+1}^* \right\|_2 + \frac{1}{2} \sum_{j=1}^{\gamma} \mu \| A_{j-1} \|_2 \| D_{2,j-1} \|_2 \left\| \bar{Y}_{j-1}^* \right\|_2 \\ &= \sum_{j=1}^{\gamma} \left[\frac{1}{2} \left(\| I - \mu D_{1,j} A_j \|_2 + \| I - \mu B_{j-1} D_{2,j-1} \|_2 \right) \left\| \bar{Y}_j^* \right\|_2 \right] \\ &\quad + \frac{1}{2} \sum_{j=1}^{\gamma} \mu \| D_{1,j-1} \|_2 \| B_{j-1} \|_2 \left\| \bar{Y}_j^* \right\|_2 + \frac{1}{2} \sum_{j=1}^{\gamma} \mu \| A_j \|_2 \| D_{2,j} \|_2 \left\| \bar{Y}_j^* \right\|_2 \\ &= \frac{1}{2} \sum_{j=1}^{\gamma} \left(\| I - \mu D_{1,j} A_j \|_2 + \| I - \mu B_{j-1} D_{2,j-1} \|_2 + \mu \| D_{1,j-1} \|_2 \| B_{j-1} \|_2 + \mu \| A_j \|_2 \| D_{2,j} \|_2 \right) \left\| \bar{Y}_j^* \right\|_2 \\ &\le \frac{1}{2} \sum_{j=1}^{\gamma} \left(\| I - \mu D_{1,j} A_j \|_2 + \| I - \mu B_{j-1} D_{2,j-1} \|_2 + \mu \| D_{1,j-1} \|_2 \| B_{j-1} \|_2 + \mu \| A_j \|_2 \| D_{2,j} \|_2 \right) \sum_{j=1}^{\gamma} \left\| \bar{Y}_j^* \right\|_2 \end{aligned} \tag{21}$$

Denote

$$q = \frac{1}{2} \sum_{j=1}^{\gamma} \left(\| I - \mu D_{1,j} A_j \|_2 + \| I - \mu B_{j-1} D_{2,j-1} \|_2 + \mu \| D_{1,j-1} \|_2 \| B_{j-1} \|_2 + \mu \| A_j \|_2 \| D_{2,j} \|_2 \right),$$

then (21) leads to $\bar{U}^* \le q \bar{U}^*$, and for any positive integer t, it holds that

$$0 \le \bar{U}^* \le q \bar{U}^* \le q^2 \bar{U}^* \le \cdots \le q^t \bar{U}^*. \tag{22}$$

Under the condition $\sum_{j=1}^{\gamma}\left(\|I-\mu D_{1,j}A_j\|_2+\|I-\mu B_{j-1}D_{2,j-1}\|_2+\mu\|D_{1,j-1}\|_2\|B_{j-1}\|_2+\mu\|A_j\|_2\|D_{2,j}\|_2\right)<2$, i.e., $q<1$, we have $\lim_{t\to+\infty} q^t=0$. Let $t\to+\infty$ in (22), then $0\le \bar{U}^*=\sum_{j=1}^{\gamma}\left\|\bar{Y}_j^*\right\|_2\to 0$, and hence $\bar{Y}_j^*=0$ $(j=1,\ldots,\gamma)$, i.e., $\tilde{Y}_j^*=\hat{Y}_j^*$ $(j=1,\ldots,\gamma)$, which leads to $\tilde{Y}^*=\left(\tilde{Y}_1^*,\tilde{Y}_2^*,\cdots,\tilde{Y}_\gamma^*\right)=\left(\hat{Y}_1^*,\hat{Y}_2^*,\cdots,\hat{Y}_\gamma^*\right)=\hat{Y}^*$, thus we conclude that the solution of the DTPS matrix equation (1) is unique.

Let $Y^*=\left(Y_1^*,Y_2^*,\cdots,Y_\gamma^*\right)$ be the unique solution of the DTPS matrix equation (1). We define the following error matrices

$$\tilde{Y}_j\left(l\right)=Y_j\left(l\right)-Y_j^*,\ \tilde{Y}_{1,j}\left(l\right)=Y_{1,j}\left(l\right)-Y_j^*,\ \tilde{Y}_{2,j}\left(l\right)=Y_{2,j}\left(l\right)-Y_j^*,\ j=1,\ldots,\gamma.$$

According to Algorithm 3.1, we obtain

$$\begin{aligned}\tilde{Y}_{1,j}\left(l+1\right)&=\tilde{Y}_j\left(l\right)-\mu D_{1,j}\left(A_j\tilde{Y}_j\left(l\right)+\tilde{Y}_{j+1}\left(l\right)B_i\right)\\&=\tilde{Y}_j\left(l\right)-\mu D_{1,j}A_j\tilde{Y}_j\left(l\right)-\mu D_{1,j}\tilde{Y}_{j+1}\left(l\right)B_j, \qquad (23)\\\tilde{Y}_{2,j}\left(l+1\right)&=\tilde{Y}_j\left(l\right)-\mu\left(A_{j-1}\tilde{Y}_{j-1}\left(l\right)+\tilde{Y}_j\left(l\right)B_{j-1}\right)D_{2,j-1}\\&=\tilde{Y}_j\left(l\right)-\mu A_{j-1}\tilde{Y}_{j-1}\left(l\right)D_{2,j-1}-\mu\tilde{Y}_j\left(l\right)B_{j-1}D_{2,j-1}, \qquad (24)\\\tilde{Y}_j\left(l+1\right)&=\frac{1}{2}\tilde{Y}_{1,j}\left(l+1\right)+\frac{1}{2}\tilde{Y}_{2,j}\left(l+1\right). \qquad (25)\end{aligned}$$

Then substituting (23)-(24) into (25) leads to

$$\begin{aligned}\tilde{Y}_j\left(l+1\right)\ &=\ \frac{1}{2}\tilde{Y}_j\left(l\right)-\frac{1}{2}\mu D_{1,j}A_j\tilde{Y}_j\left(l\right)+\frac{1}{2}\tilde{Y}_j\left(l\right)-\frac{1}{2}\mu\tilde{Y}_j\left(l\right)B_{j-1}D_{2,j-1}\\&\quad-\frac{1}{2}\mu D_{1,j}\tilde{Y}_{j+1}\left(l\right)B_j-\frac{1}{2}\mu A_{j-1}\tilde{Y}_{j-1}\left(l\right)D_{2,j-1}\\&=\ \frac{1}{2}\left(I-\mu D_{1,j}A_j\right)\tilde{Y}_j\left(l\right)+\frac{1}{2}\tilde{Y}_j\left(l\right)\left(I-\mu B_{j-1}D_{2,j-1}\right)\\&\quad-\frac{1}{2}\mu D_{1,j}\tilde{Y}_{j+1}\left(l\right)B_j-\frac{1}{2}\mu A_{j-1}\tilde{Y}_{j-1}\left(l\right)D_{2,j-1}. \qquad (26)\end{aligned}$$

By taking the 2-norm in (26) and using the properties of the matrix norm, we deduce that

$$\begin{aligned}\left\|\tilde{Y}_j\left(l+1\right)\right\|_2&=\left\|\frac{1}{2}\left(I-\mu D_{1,j}A_j\right)\tilde{Y}_j\left(l\right)+\frac{1}{2}\tilde{Y}_j\left(l\right)\left(I-\mu B_{j-1}D_{2,j-1}\right)\right.\\&\quad\left.-\frac{1}{2}\mu D_{1,j}\tilde{Y}_{j+1}\left(l\right)B_j-\frac{1}{2}\mu A_{j-1}\tilde{Y}_{j-1}\left(l\right)D_{2,j-1}\right\|_2\\&\le\frac{1}{2}\|I-\mu D_{1,j}A_j\|_2\left\|\tilde{Y}_j\left(l\right)\right\|_2+\frac{1}{2}\|I-\mu B_{j-1}D_{2,j-1}\|_2\left\|\tilde{Y}_j\left(l\right)\right\|_2\\&\quad+\frac{1}{2}\mu\|D_{1,j}\|_2\|B_j\|_2\left\|\tilde{Y}_{j+1}\left(l\right)\right\|_2+\frac{1}{2}\mu\|A_{j-1}\|_2\|D_{2,j-1}\|_2\left\|\tilde{Y}_{j-1}\left(l\right)\right\|_2. \qquad (27)\end{aligned}$$

Next we define the following non-negative matrix norm function $H(l)$

$$H\left(l\right)=\sum_{j=1}^{\gamma}\left\|\tilde{Y}_j\left(l\right)\right\|_2,$$

which together with (27) gives

$$\begin{aligned}H\left(l+1\right)&=\sum_{j=1}^{\gamma}\left\|\tilde{Y}_j\left(l+1\right)\right\|_2\\&\le\sum_{j=1}^{\gamma}\left[\frac{1}{2}\left(\|I-\mu D_{1,j}A_j\|_2+\|I-\mu B_{j-1}D_{2,j-1}\|_2\right)\left\|\tilde{Y}_j\left(l\right)\right\|_2\right]\\&\quad+\frac{1}{2}\sum_{j=1}^{\gamma}\mu\|D_{1,j}\|_2\|B_j\|_2\left\|\tilde{Y}_{j+1}\left(l\right)\right\|_2+\frac{1}{2}\sum_{j=1}^{\gamma}\mu\|A_{j-1}\|_2\|D_{2,j-1}\|_2\left\|\tilde{Y}_{j-1}\left(l\right)\right\|_2\end{aligned}$$

$$= \sum_{j=1}^{\gamma} \left[\frac{1}{2} \left(\|I - \mu D_{1,j} A_j\|_2 + \|I - \mu B_{j-1} D_{2,j-1}\|_2 \right) \left\| \tilde{Y}_j (l) \right\|_2 \right]$$
$$+ \frac{1}{2} \sum_{j=1}^{\gamma} \mu \|D_{1,j-1}\|_2 \|B_{j-1}\|_2 \left\| \tilde{Y}_j (l) \right\|_2 + \frac{1}{2} \sum_{j=1}^{\gamma} \mu \|A_j\|_2 \|D_{2,j}\|_2 \left\| \tilde{Y}_j (l) \right\|_2$$
$$= \frac{1}{2} \sum_{j=1}^{\gamma} \left(\|I - \mu D_{1,j} A_j\|_2 + \|I - \mu B_{j-1} D_{2,j-1}\|_2 + \mu \|D_{1,j-1}\|_2 \|B_{j-1}\|_2 + \mu \|A_j\|_2 \|D_{2,j}\|_2 \right) \left\| \tilde{Y}_j (l) \right\|_2$$
$$\leq \frac{1}{2} \sum_{j=1}^{\gamma} \left(\|I - \mu D_{1,j} A_j\|_2 + \|I - \mu B_{j-1} D_{2,j-1}\|_2 + \mu \|D_{1,j-1}\|_2 \|B_{j-1}\|_2 + \mu \|A_j\|_2 \|D_{2,j}\|_2 \right) \sum_{j=1}^{\gamma} \left\| \tilde{Y}_j (l) \right\|_2$$
$$= qH (l) .$$

This leads to the following result

$$H (l + 1) \leq qH (l) \leq q^2 H (l - 1) \leq \cdots \leq q^{l+1} H (0) .$$

Hence, if $q < 1$, that is

$$\sum_{j=1}^{\gamma} \left(\|I - \mu D_{1,j} A_j\|_2 + \|I - \mu B_{j-1} D_{2,j-1}\|_2 + \mu \|D_{1,j-1}\|_2 \|B_{j-1}\|_2 + \mu \|A_j\|_2 \|D_{2,j}\|_2 \right) < 2,$$

then $\lim_{l \to +\infty} \sum_{j=1}^{\gamma} \left\| \tilde{Y}_j (l + 1) \right\|_2 = 0$ and therefore $\lim_{l \to +\infty} \tilde{Y}_j(l + 1) = 0$ $(j = 1, \ldots, \gamma)$. This shows that

$$\lim_{l \to +\infty} Y_j (l + 1) = Y_j^*, j = 1, 2, \ldots, \gamma.$$

The proof is completed. ■

Now we turn to review the function $Z(l + 1)$ in the proof of Theorem 3.3 in [39]:

$$Z (l + 1) = \sum_{j=1}^{\gamma} \left\| (1 - \omega) \tilde{Y}_{1,j} (l + 1) + \omega \tilde{Y}_{2,j} (l + 1) \right\|^2 \tag{28}$$
$$\leq 2 \sum_{j=1}^{\gamma} \left[(1 - \omega)^2 \left\| \tilde{Y}_{1,j} (l + 1) \right\|^2 + \omega^2 \left\| \tilde{Y}_{2,j} (l + 1) \right\|^2 \right] \tag{29}$$
$$= 2 \sum_{j=1}^{\gamma} \Big[(1 - \omega)^2 \left\| \tilde{Y}_j (l) \right\|^2 - 2\mu\omega(1 - \omega)^2 tr \left(\tilde{Y}_j^T (l) D_{1,j} \tilde{\delta}_j (l) \right) + \mu^2 \omega^2 (1 - \omega)^2 \left\| D_{1,j} \tilde{\delta}_j (l) \right\|^2 \tag{30}$$
$$+ \omega^2 \left\| \tilde{\hat{Y}}_j (l) \right\|^2 - 2\mu\omega^2 (1 - \omega) tr \left(\tilde{\hat{Y}}_{j+1}^T (l) \tilde{\psi}_j (l) D_{2,j} \right) + \mu^2 \omega^2 (1 - \omega)^2 \left\| \tilde{\psi}_j (l) D_{2,j} \right\|^2 \Big]$$
$$\leq 2 \sum_{j=1}^{\gamma} \Big[(1 - \omega)^2 \left\| \tilde{Y}_j (l) \right\|^2 - 2\mu\omega(1 - \omega)^2 \left\| \tilde{\delta}_j (l) \right\|^2 + \mu^2 \omega^2 (1 - \omega)^2 \|D_{1,j}\|^2 \left\| \tilde{\delta}_j (l) \right\|^2 \tag{31}$$
$$+ \omega^2 \left\| \tilde{\hat{Y}}_j (l) \right\|^2 - 2\mu\omega^2 (1 - \omega) \left\| \tilde{\psi}_j (l) \right\|^2 + \mu^2 \omega^2 (1 - \omega)^2 \left\| \tilde{\psi}_j (l) \right\|^2 \|D_{2,j}\|^2 \Big],$$

where $\tilde{\delta}_j(l) = A_j \tilde{Y}_j(l) + \tilde{Y}_{j+1}(l) B_j$ and $\tilde{\psi}_j(l) = A_j \tilde{\hat{Y}}_j(l) + \tilde{\hat{Y}}_{j+1}(l) B_j$.

By using the same analytical method applied in (14), we observe that

$$\left\| \tilde{\delta}_j (l) \right\|^2 = tr \left\{ \left[\tilde{Y}_j^T (l) \left(D_{1,j} + R_{1,j}^T \right) + \left(D_{2,j} + R_{2,j}^T \right) \tilde{Y}_{j+1}^T (l) \right] \tilde{\delta}_j (l) \right\} \leq tr \left[\tilde{Y}_j^T (l) D_{1,j} \tilde{\delta}_j (l) \right] \tag{32}$$

and

$$\left\| \tilde{\psi}_j (l) \right\|^2 = tr \left[\tilde{\hat{Y}}_j^T (l) A_j^T \tilde{\psi}_j (l) + (D_{2,j} + R_{2,j}^T) \tilde{\hat{Y}}_{j+1}^T (l) \tilde{\psi}_j (l) \right] \leq tr \left[D_{2,j} \tilde{\hat{Y}}_{j+1}^T (l) \tilde{\psi}_j (l) \right] = tr \left[\tilde{\hat{Y}}_{j+1}^T (l) \tilde{\psi}_j (l) D_{2,j} \right]$$

are not always true. Therefore, the derivation of (25) from (24) is not correct. In what follows, we establish the correct convergence theorem of the AJGI algorithm. To this end, we first define the following matrix

$$\mathcal{F}=\begin{bmatrix}
(1-\omega)M_1 & \omega M_1 & (1-\omega)N_1 & \omega N_1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
(1-\omega)^2V_1 & \omega Z_1 & (1-\omega)^2P_1 & \omega(1-\omega)P_1 & 0 & 0 & \cdots & 0 & 0 & (1-\omega)^2W_\gamma & \omega U_\gamma\\
0 & 0 & (1-\omega)M_2 & \omega M_2 & (1-\omega)N_2 & \omega N_2 & \cdots & 0 & 0 & 0 & 0\\
(1-\omega)^2W_1 & \omega U_1 & (1-\omega)^2V_2 & \omega Z_2 & (1-\omega)^2P_2 & \omega(1-\omega)P_2 & \cdots & 0 & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
(1-\omega)N_\gamma & \omega N_\gamma & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & (1-\omega)M_\gamma & \omega M_\gamma\\
(1-\omega)^2P_\gamma & \omega(1-\omega)P_\gamma & 0 & 0 & 0 & 0 & \cdots & (1-\omega)^2W_{\gamma-1} & \omega U_{\gamma-1} & (1-\omega)^2V_\gamma & \omega Z_\gamma
\end{bmatrix}, \tag{33}$$

with

$$\begin{aligned}
&M_j = I\otimes(I-\mu\omega D_{1,j}A_j),\ N_j=-\mu\omega B_j^T\otimes D_{1,j},\\
&G_j=[I-\mu(1-\omega)B_jD_{2,j}]^T\otimes I,\ H_j=-\mu(1-\omega)D_{2,j}\otimes A_j,\\
&P_j=G_{j-1}N_j,\ W_j=H_jM_j,\ U_j=H_j[(1-\omega)M_j+I],\\
&V_j=G_{j-1}M_j+H_{j-1}N_{j-1},\ Z_j=G_{j-1}[(1-\omega)M_j+I]+(1-\omega)H_{j-1}N_{j-1},\ j=1,\dots,\gamma.
\end{aligned}$$

Theorem 3.2. *Assume that the DTPS matrix equation* (1) *is consistent, i.e., the solution of the DTPS matrix equation* (1) *exists. Then the iterative sequences* $\{Y_j(l)\}$ $(j=1,\dots,\gamma)$ *generated by the AJGI algorithm converge to the unique solution* $Y^*=\left(Y_1^*,Y_2^*,\cdots,Y_\gamma^*\right)$ *for any initial matrices* $Y_j(0)$ $(j=1,\dots,\gamma)$ *if the parameters* μ *and* ω *are selected to satisfy*

$$\rho(\mathcal{F})<1,$$

where the matrix $\mathcal{F}$ *is defined as in* (33).

Proof. First of all, we prove that the solution of the DTPS matrix equation (1) is unique. Assume that $\tilde{Y}^*=\left(\tilde{Y}_1^*,\tilde{Y}_2^*,\cdots,\tilde{Y}_\gamma^*\right)$ and $\hat{Y}^*=\left(\hat{Y}_1^*,\hat{Y}_2^*,\cdots,\hat{Y}_\gamma^*\right)$ are two solutions of the DTPS matrix equation (1), then we have

$$A_j\tilde{Y}_j^*+\tilde{Y}_{j+1}^*B_j=C_j,\ A_j\hat{Y}_j^*+\hat{Y}_{j+1}^*B_j=C_j,\ j=1,2,\dots,\gamma.$$

It follows from $A_j\tilde{Y}_j^*+\tilde{Y}_{j+1}^*B_j=C_j$ $(j=1,2,\dots,\gamma)$ that

$$\begin{aligned}
&\tilde{Y}_j^*=\tilde{Y}_j^*+\mu\omega D_{1,j}\left(C_j-A_j\tilde{Y}_j^*-\tilde{Y}_{j+1}^*B_j\right),\ \tilde{Y}_j^*=\tilde{Y}_j^*+\mu(1-\omega)\left(C_{j-1}-A_{j-1}\tilde{Y}_{j-1}^*-\tilde{Y}_j^*B_{j-1}\right)D_{2,j-1},\\
&\tilde{Y}_j^*=(1-\omega)\tilde{Y}_j^*+\omega\tilde{Y}_j^*,\ \tilde{Y}_j^*=(1-\omega)\tilde{Y}_j^*+\omega\tilde{Y}_j^*.
\end{aligned}\tag{34}$$

Similarly, from $A_j\hat{Y}_j^*+\hat{Y}_{j+1}^*B_j=C_j$ $(j=1,2,\dots,\gamma)$, we can deduce that

$$\begin{aligned}
&\hat{Y}_j^*=\hat{Y}_j^*+\mu\omega D_{1,j}\left(C_j-A_j\hat{Y}_j^*-\hat{Y}_{j+1}^*B_j\right),\ \hat{Y}_j^*=\hat{Y}_j^*+\mu(1-\omega)\left(C_{j-1}-A_{j-1}\hat{Y}_{j-1}^*-\hat{Y}_j^*B_{j-1}\right)D_{2,j-1},\\
&\hat{Y}_j^*=(1-\omega)\hat{Y}_j^*+\omega\hat{Y}_j^*,\ \hat{Y}_j^*=(1-\omega)\hat{Y}_j^*+\omega\hat{Y}_j^*.
\end{aligned}\tag{35}$$

Let $\bar{Y}_j^*=\tilde{Y}_j^*-\hat{Y}_j^*$ $(j=1,\dots,\gamma)$. The combination of (34) and (35) gives

$$\begin{aligned}
\bar{Y}_j^* &= \bar{Y}_j^*-\mu\omega D_{1,j}\left(A_j\bar{Y}_j^*+\bar{Y}_{j+1}^*B_j\right)\\
&= \bar{Y}_j^*-\mu\omega D_{1,j}A_j\bar{Y}_j^*-\mu\omega D_{1,j}\bar{Y}_{j+1}^*B_j=(I-\mu\omega D_{1,j}A_j)\bar{Y}_j^*-\mu\omega D_{1,j}\bar{Y}_{j+1}^*B_j, &(36)\\
\bar{Y}_j^* &= \bar{Y}_j^*-\mu(1-\omega)\left(A_{j-1}\bar{Y}_{j-1}^*+\bar{Y}_j^*B_{j-1}\right)D_{2,j-1}\\
&= \bar{Y}_j^*-\mu(1-\omega)A_{j-1}\bar{Y}_{j-1}^*D_{2,j-1}-\mu(1-\omega)\bar{Y}_j^*B_{j-1}D_{2,j-1}\\
&= \bar{Y}_j^*[I-\mu(1-\omega)B_{j-1}D_{2,j-1}]-\mu(1-\omega)A_{j-1}\bar{Y}_{j-1}^*D_{2,j-1}, &(37)\\
\bar{Y}_j^* &= (1-\omega)\bar{Y}_j^*+\omega\bar{Y}_j^*, &(38)\\
\bar{Y}_j^* &= (1-\omega)\bar{Y}_j^*+\omega\bar{Y}_j^*. &(39)
\end{aligned}$$

Taking the vec-operator on both sides of (36)-(39) results in

$$\begin{aligned}
\mathrm{vec}\left(\bar{Y}_j^*\right) &= [I\otimes(I-\mu\omega D_{1,j}A_j)]\,\mathrm{vec}\left(\bar{Y}_j^*\right)-\left(\mu\omega B_j^T\otimes D_{1,j}\right)\mathrm{vec}\left(\bar{Y}_{j+1}^*\right)=M_j\mathrm{vec}\left(\bar{Y}_j^*\right)+N_j\mathrm{vec}\left(\bar{Y}_{j+1}^*\right), &(40)\\
\mathrm{vec}\left(\bar{Y}_j^*\right) &= \left\{[I-\mu(1-\omega)B_{j-1}D_{2,j-1}]^T\otimes I\right\}\mathrm{vec}\left(\bar{Y}_j^*\right)-[\mu(1-\omega)D_{2,j-1}\otimes A_{j-1}]\,\mathrm{vec}\left(\bar{Y}_{j-1}^*\right)\\
&= G_{j-1}\mathrm{vec}\left(\bar{Y}_j^*\right)+H_{j-1}\mathrm{vec}\left(\bar{Y}_{j-1}^*\right), &(41)
\end{aligned}$$

$$
\begin{aligned}
\operatorname{vec}\left(\bar{Y}_j^*\right) &= (1-\omega)\operatorname{vec}\left(\bar{Y}_j^*\right) + \omega\operatorname{vec}\left(\bar{Y}_j^*\right), \qquad (42)\\
\operatorname{vec}\left(\bar{Y}_j^*\right) &= (1-\omega)\operatorname{vec}\left(\bar{Y}_j^*\right) + \omega\operatorname{vec}\left(\bar{Y}_j^*\right), \qquad (43)
\end{aligned}
$$

in terms of Lemma 2.1. By substituting (43) into (40), it holds that

$$
\begin{aligned}
\operatorname{vec}\left(\bar{Y}_j^*\right) &= (1-\omega) M_j\operatorname{vec}\left(\bar{Y}_j^*\right) + \omega M_j\operatorname{vec}\left(\bar{Y}_j^*\right)\\
&\quad + (1-\omega) N_j\operatorname{vec}\left(\bar{Y}_{j+1}^*\right) + \omega N_j\operatorname{vec}\left(\bar{Y}_{j+1}^*\right). \qquad (44)
\end{aligned}
$$

By combining (44) with (42), we have

$$
\begin{aligned}
\operatorname{vec}\left(\bar{Y}_j^*\right) &= (1-\omega)\left\{(1-\omega) M_j\operatorname{vec}\left(\bar{Y}_j^*\right) + \omega M_j\operatorname{vec}\left(\bar{Y}_j^*\right)\right.\\
&\quad \left.+ (1-\omega) N_j\operatorname{vec}\left(\bar{Y}_{j+1}^*\right) + \omega N_j\operatorname{vec}\left(\bar{Y}_{j+1}^*\right)\right\} + \omega\operatorname{vec}\left(\bar{Y}_j^*\right)\\
&= (1-\omega)^2 M_j\operatorname{vec}\left(\bar{Y}_j^*\right) + (1-\omega)^2 N_j\operatorname{vec}\left(\bar{Y}_{j+1}^*\right)\\
&\quad + \omega\left[(1-\omega) M_j + I\right]\operatorname{vec}\left(\bar{Y}_j^*\right) + \omega(1-\omega) N_j\operatorname{vec}\left(\bar{Y}_{j+1}^*\right). \qquad (45)
\end{aligned}
$$

In addition, substituting (45) into (41) results in

$$
\begin{aligned}
\operatorname{vec}\left(\bar{Y}_j^*\right) &= (1-\omega)^2 H_{j-1} M_{j-1}\operatorname{vec}\left(\bar{Y}_{j-1}^*\right) + \omega H_{j-1}\left[(1-\omega) M_{j-1} + I\right]\operatorname{vec}\left(\bar{Y}_{j-1}^*\right)\\
&\quad + (1-\omega)^2\left(G_{j-1} M_j + H_{j-1} N_{j-1}\right)\operatorname{vec}\left(\bar{Y}_j^*\right) + \omega\left\{G_{j-1}\left[(1-\omega) M_j + I\right] + (1-\omega) H_{j-1} N_{j-1}\right\}\operatorname{vec}\left(\bar{Y}_j^*\right)\\
&\quad + (1-\omega)^2 G_{j-1} N_j\operatorname{vec}\left(\bar{Y}_{j+1}^*\right) + \omega(1-\omega) G_{j-1} N_j\operatorname{vec}\left(\bar{Y}_{j+1}^*\right)\\
&= (1-\omega)^2 W_{j-1}\operatorname{vec}\left(\bar{Y}_{j-1}^*\right) + \omega U_{j-1}\operatorname{vec}\left(\bar{Y}_{j-1}^*\right) + (1-\omega)^2 V_j\operatorname{vec}\left(\bar{Y}_j^*\right) + \omega Z_j\operatorname{vec}\left(\bar{Y}_j^*\right)\\
&\quad + (1-\omega)^2 P_j\operatorname{vec}\left(\bar{Y}_{j+1}^*\right) + \omega(1-\omega) P_j\operatorname{vec}\left(\bar{Y}_{j+1}^*\right). \qquad (46)
\end{aligned}
$$

Then from (33), (44) and (46), we conclude that for any positive integer t, it has

$$
\begin{pmatrix}
\operatorname{vec}\left(\bar{Y}_1^*\right)\\
\operatorname{vec}\left(\bar{Y}_1^*\right)\\
\operatorname{vec}\left(\bar{Y}_2^*\right)\\
\operatorname{vec}\left(\bar{Y}_2^*\right)\\
\vdots\\
\operatorname{vec}\left(\bar{Y}_\gamma^*\right)\\
\operatorname{vec}\left(\bar{Y}_\gamma^*\right)
\end{pmatrix}
= \mathcal{F}
\begin{pmatrix}
\operatorname{vec}\left(\bar{Y}_1^*\right)\\
\operatorname{vec}\left(\bar{Y}_1^*\right)\\
\operatorname{vec}\left(\bar{Y}_2^*\right)\\
\operatorname{vec}\left(\bar{Y}_2^*\right)\\
\vdots\\
\operatorname{vec}\left(\bar{Y}_\gamma^*\right)\\
\operatorname{vec}\left(\bar{Y}_\gamma^*\right)
\end{pmatrix}
= \mathcal{F}^t
\begin{pmatrix}
\operatorname{vec}\left(\bar{Y}_1^*\right)\\
\operatorname{vec}\left(\bar{Y}_1^*\right)\\
\operatorname{vec}\left(\bar{Y}_2^*\right)\\
\operatorname{vec}\left(\bar{Y}_2^*\right)\\
\vdots\\
\operatorname{vec}\left(\bar{Y}_\gamma^*\right)\\
\operatorname{vec}\left(\bar{Y}_\gamma^*\right)
\end{pmatrix}. \qquad (47)
$$

If $\rho(\mathcal{F}) < 1$, then it follows that $\lim\limits_{t\to+\infty}\mathcal{F}^t = 0$. Let $t\to+\infty$ in (47), we obtain $\operatorname{vec}\left(\bar{Y}_j^*\right) = 0$ $(j = 1,\ldots,\gamma)$, thus $\tilde{Y}_j^* = \hat{Y}_j^*$ $(j = 1,\ldots,\gamma)$. This implies that $\tilde{Y}^* = \left(\tilde{Y}_1^*, \tilde{Y}_2^*, \cdots, \tilde{Y}_\gamma^*\right) = \left(\hat{Y}_1^*, \hat{Y}_2^*, \cdots, \hat{Y}_\gamma^*\right) = \hat{Y}^*$, and therefore the solution of the DTPS matrix equation (1) is unique.

Let $Y^* = \left(Y_1^*, Y_2^*, \cdots, Y_\gamma^*\right)$ be the unique solution of the DTPS matrix equation (1). Similar to Theorem 3.1, we define the error matrices

$$
\begin{aligned}
&\tilde{Y}_{1,j}(l) = Y_{1,j}(l) - Y_j^*,\ \tilde{Y}_{2,j}(l) = Y_{2,j}(l) - Y_j^*,\\
&\tilde{\hat{Y}}_j(l) = \hat{Y}_j(l) - Y_j^*,\ \tilde{Y}_j(l) = Y_j(l) - Y_j^*, j = 1,\ldots,\gamma. \qquad (48)
\end{aligned}
$$

According to the iteration scheme of Algorithm 3.2, it holds that

$$
\begin{aligned}
\tilde{Y}_{1,j}(l+1) &= \tilde{Y}_j(l) - \mu\omega D_{1,j}\left(A_j\tilde{Y}_j(l) + \tilde{Y}_{j+1}(l) B_j\right)\\
&= \tilde{Y}_j(l) - \mu\omega D_{1,j} A_j\tilde{Y}_j(l) - \mu\omega D_{1,j}\tilde{Y}_{j+1}(l) B_j\\
&= \left(I - \mu\omega D_{1,j} A_j\right)\tilde{Y}_j(l) - \mu\omega D_{1,j}\tilde{Y}_{j+1}(l) B_j, \qquad (49)\\
\tilde{Y}_{2,j}(l+1) &= \tilde{\hat{Y}}_j(l) - \mu(1-\omega)\left(A_{j-1}\tilde{\hat{Y}}_{j-1}(l) + \tilde{\hat{Y}}_j(l) B_{j-1}\right) D_{2,j-1}\\
&= \tilde{\hat{Y}}_j(l) - \mu(1-\omega) A_{j-1}\tilde{\hat{Y}}_{j-1}(l) D_{2,j-1} - \mu(1-\omega)\tilde{\hat{Y}}_j(l) B_{j-1} D_{2,j-1}\\
&= \tilde{\hat{Y}}_j(l)\left[I - \mu(1-\omega) B_{j-1} D_{2,j-1}\right] - \mu(1-\omega) A_{j-1}\tilde{\hat{Y}}_{j-1}(l) D_{2,j-1}, \qquad (50)
\end{aligned}
$$

$$\widetilde{\tilde{Y}}_j(l) = (1-\omega)\,\tilde{Y}_{1,j}\,(l+1) + \omega\tilde{Y}_{2,j}\,(l)\,, \tag{51}$$

$$\tilde{Y}_j\,(l+1) = (1-\omega)\,\tilde{Y}_{1,j}\,(l+1) + \omega\tilde{Y}_{2,j}\,(l+1)\,. \tag{52}$$

By taking the vec-operator on both sides of (49)-(52) and using Lemma 2.1, we have

$$\begin{aligned}
\mathrm{vec}\left[\tilde{Y}_{1,j}\,(l+1)\right] &= \left[I\otimes\left(I-\mu\omega D_{1,j}A_j\right)\right]\mathrm{vec}\left[\tilde{Y}_j\,(l)\right] - \left(\mu\omega B_j^T\otimes D_{1,j}\right)\mathrm{vec}\left[\tilde{Y}_{j+1}\,(l)\right]\\
&= M_j\mathrm{vec}\left[\tilde{Y}_j\,(l)\right] + N_j\mathrm{vec}\left[\tilde{Y}_{j+1}\,(l)\right], &(53)\\
\mathrm{vec}\left[\tilde{Y}_{2,j}\,(l+1)\right] &= \left\{\left[I-\mu\,(1-\omega)\,B_{j-1}D_{2,j-1}\right]^T\otimes I\right\}\mathrm{vec}\left[\widetilde{\tilde{Y}}_j(l)\right] - \left[\mu\,(1-\omega)\,D_{2,j-1}\otimes A_{j-1}\right]\mathrm{vec}\left[\widetilde{\tilde{Y}}_{j-1}(l)\right]\\
&= G_{j-1}\mathrm{vec}\left[\widetilde{\tilde{Y}}_j(l)\right] + H_{j-1}\mathrm{vec}\left[\widetilde{\tilde{Y}}_{j-1}(l)\right], &(54)\\
\mathrm{vec}\left[\widetilde{\tilde{Y}}_j(l)\right] &= (1-\omega)\,\mathrm{vec}\left[\tilde{Y}_{1,j}\,(l+1)\right] + \omega\mathrm{vec}\left[\tilde{Y}_{2,j}\,(l)\right], &(55)\\
\mathrm{vec}\left[\tilde{Y}_j(l+1)\right] &= (1-\omega)\,\mathrm{vec}\left[\tilde{Y}_{1,j}\,(l+1)\right] + \omega\mathrm{vec}\left[\tilde{Y}_{2,j}\,(l+1)\right]. &(56)
\end{aligned}$$

Substituting (56) into (53) yields that

$$\begin{aligned}
\mathrm{vec}\left[\tilde{Y}_{1,j}\,(l+1)\right] = {}&(1-\omega)\,M_j\mathrm{vec}\left[\tilde{Y}_{1,j}\,(l)\right] + \omega M_j\mathrm{vec}\left[\tilde{Y}_{2,j}\,(l)\right]\\
&+ (1-\omega)\,N_j\mathrm{vec}\left[\tilde{Y}_{1,j+1}\,(l)\right] + \omega N_j\mathrm{vec}\left[\tilde{Y}_{2,j+1}\,(l)\right]. &(57)
\end{aligned}$$

By combining (57) with (55), it has

$$\begin{aligned}
\mathrm{vec}\left[\widetilde{\tilde{Y}}_j(l)\right] &= (1-\omega)\left\{(1-\omega)\,M_j\mathrm{vec}\left[\tilde{Y}_{1,j}\,(l)\right] + \omega M_j\mathrm{vec}\left[\tilde{Y}_{2,j}\,(l)\right]\right.\\
&\quad\left. + (1-\omega)\,N_j\mathrm{vec}\left[\tilde{Y}_{1,j+1}\,(l)\right] + \omega N_j\mathrm{vec}\left[\tilde{Y}_{2,j+1}\,(l)\right]\right\} + \omega\mathrm{vec}\left[\tilde{Y}_{2,j}\,(l)\right]\\
&= (1-\omega)^2 M_j\mathrm{vec}\left[\tilde{Y}_{1,j}\,(l)\right] + (1-\omega)^2 N_j\mathrm{vec}\left[\tilde{Y}_{1,j+1}\,(l)\right]\\
&\quad + \omega\left[(1-\omega)\,M_j + I\right]\mathrm{vec}\left[\tilde{Y}_{2,j}\,(l)\right] + \omega\,(1-\omega)\,N_j\mathrm{vec}\left[\tilde{Y}_{2,j+1}\,(l)\right]. &(58)
\end{aligned}$$

Besides, substituting (58) into (54) results in

$$\begin{aligned}
&\mathrm{vec}\left[\tilde{Y}_{2,j}\,(l+1)\right]\\
&= (1-\omega)^2 H_{j-1}M_{j-1}\mathrm{vec}\left[\tilde{Y}_{1,j-1}\,(l)\right] + \omega H_{j-1}\left[(1-\omega)\,M_{j-1}+I\right]\mathrm{vec}\left[\tilde{Y}_{2,j-1}\,(l)\right]\\
&\quad + (1-\omega)^2\left(G_{j-1}M_j + H_{j-1}N_{j-1}\right)\mathrm{vec}\left[\tilde{Y}_{1,j}\,(l)\right] + \omega\left\{G_{j-1}\left[(1-\omega)\,M_j+I\right] + (1-\omega)\,H_{j-1}N_{j-1}\right\}\mathrm{vec}\left[\tilde{Y}_{2,j}\,(l)\right]\\
&\quad + (1-\omega)^2 G_{j-1}N_j\mathrm{vec}\left[\tilde{Y}_{1,j+1}\,(l)\right] + \omega\,(1-\omega)\,G_{j-1}N_j\mathrm{vec}\left[\tilde{Y}_{2,j+1}\,(l)\right]\\
&= (1-\omega)^2 W_{j-1}\mathrm{vec}\left[\tilde{Y}_{1,j-1}\,(l)\right] + \omega U_{j-1}\mathrm{vec}\left[\tilde{Y}_{2,j-1}\,(l)\right] + (1-\omega)^2 V_j\mathrm{vec}\left[\tilde{Y}_{1,j}\,(l)\right] + \omega Z_j\mathrm{vec}\left[\tilde{Y}_{2,j}\,(l)\right]\\
&\quad + (1-\omega)^2 P_j\mathrm{vec}\left[\tilde{Y}_{1,j+1}\,(l)\right] + \omega\,(1-\omega)\,P_j\mathrm{vec}\left[\tilde{Y}_{2,j+1}\,(l)\right]. &(59)
\end{aligned}$$

Then it follows from (33), (57) and (59) that

$$\begin{pmatrix}
\mathrm{vec}\left[\tilde{Y}_{1,1}\,(l+1)\right]\\
\mathrm{vec}\left[\tilde{Y}_{2,1}\,(l+1)\right]\\
\mathrm{vec}\left[\tilde{Y}_{1,2}\,(l+1)\right]\\
\mathrm{vec}\left[\tilde{Y}_{2,2}\,(l+1)\right]\\
\vdots\\
\mathrm{vec}\left[\tilde{Y}_{1,\gamma}\,(l+1)\right]\\
\mathrm{vec}\left[\tilde{Y}_{2,\gamma}\,(l+1)\right]
\end{pmatrix} = F\begin{pmatrix}
\mathrm{vec}\left[\tilde{Y}_{1,1}\,(l)\right]\\
\mathrm{vec}\left[\tilde{Y}_{2,1}\,(l)\right]\\
\mathrm{vec}\left[\tilde{Y}_{1,2}\,(l)\right]\\
\mathrm{vec}\left[\tilde{Y}_{2,2}\,(l)\right]\\
\vdots\\
\mathrm{vec}\left[\tilde{Y}_{1,\gamma}\,(l)\right]\\
\mathrm{vec}\left[\tilde{Y}_{2,\gamma}\,(l)\right]
\end{pmatrix}. \tag{60}$$

Therefore, the matrix $\digamma$ in (60) is the iteration matrix of the AJGI algorithm, then the necessary and sufficient condition for the convergence of the AJGI algorithm is

$$\rho(\digamma) < 1,$$

which completes the proof of this theorem. ■

Remarl 3.1. *Although the convergence condition of the AJGI algorithm is given in Theorem 3.2, intervals of the step size factor μ and the relaxation factor ω have not been determined. The reason is that the parameters ω, μ are contained in the matrices in $\digamma$, and they can not be separated from the matrices in $\digamma$. Thus it is difficult to derive the convergence intervals of ω, μ, and this problem needs to be investigated in our future work.*

4. The EJGI algorithm for the DTPS matrix equations

In this section, to further improve the efficiency of the AJGI algorithm proposed in [39], based on Lemma 2.2, we introduce a new update technique for the JGI algorithm [39], and construct an effective JGI (EJGI) algorithm for the DTPS matrix equations, which is different from the AJGI algorithm and has better numerical performance. Then we investigate the convergence property of the EJGI algorithm. The framework of the EJGI algorithm is as follows.

Algorithm 4.1. *The effective Jacobi gradient based iterative (EJGI) algorithm:*

Step 1: Input matrices $A_j \in \mathbb{R}^{m\times m}, B_j \in \mathbb{R}^{n\times n}, C_j \in \mathbb{R}^{m\times n}$ for $j = 1,\ldots,\gamma$, and three constants $\mu, \eta > 0$ and $0 < \omega < 1$. Choose the initial matrices $Y_j(0) \in \mathbb{R}^{m\times n}$ $(j = 1,\ldots,\gamma)$, and set $l = 0$;

Step 2: Take $Y_{j+\gamma}(0) = Y_j(0), A_{j+\gamma} = A_j, B_{j+\gamma} = B_j, C_{j+\gamma} = C_j, D_{1,j+\gamma} = D_{1,j}$ and $D_{2,j+\gamma} = D_{2,j}$;

Step 3: If $\eta_l = \sqrt{\dfrac{\sum_{j=1}^{\gamma}\|C_j - A_jY_j(l) - Y_{j+1}(l)B_j\|^2}{\sum_{j=1}^{\gamma}\|C_j - A_jY_j(0) - Y_{j+1}(0)B_j\|^2}} < \eta$, stop; otherwise, go to Step 4;

Step 4: For $l = 0, 1, 2, \ldots,$ and $j = 1,\ldots,\gamma$, calculate

$$Y_{1,j}(l+1) = Y_j(l) + \mu\omega D_{1,j}(C_j - A_jY_j(l) - Y_{j+1}(l)B_j),$$

$$\hat{Y}_j(l) = (1-\omega)Y_{1,j}(l+1) + \omega Y_j(l),$$

$$\hat{Y}_{j+\gamma}(l) = \hat{Y}_j(l),$$

$$Y_{2,j}(l+1) = \hat{Y}_j(l) + \mu(1-\omega)\left(C_{j-1} - A_{j-1}\hat{Y}_{j-1}(l) - \hat{Y}_j(l)B_{j-1}\right)D_{2,j-1},$$

$$Y_j(l+1) = (1-\omega)Y_{1,j}(l+1) + \omega Y_{2,j}(l+1),$$

$$Y_{j+\gamma}(l+1) = Y_j(l+1).$$

Step 5: Set $l := l+1$ and return to Step 3.

Remarl 4.1. *Compared with the AJGI algorithm, the proposed EJGI algorithm is obtained by using a new and different update technique to the JGI algorithm. In the AJGI algorithm, $\hat{Y}_j(l)$ are computed by $Y_{1,j}(l+1)$ and $Y_{2,j}(l)$ $(j = 1,\cdots,\gamma)$. While in the proposed EJGI algorithm, $Y_{2,j}(l)$ are replaced by $Y_j(l)$ to compte $\hat{Y}_j(l)$ $(j = 1,\cdots,\gamma)$. Although the frameworks of the AJGI algorithm [39] and the proposed EJGI algorithm are similar and the only differences between these two algorithms are the formulas for $\hat{Y}_j(l)$ $(j = 1,\cdots,\gamma)$, the latter one may perform better than the former one. The reason is that $Y_j(l) = (1-\omega)Y_{1,j}(l) + \omega Y_{2,j}(l)$ may be better than $Y_{2,j}(l)$ if the relaxation factor ω is chosen properly. And this fact will be illustrated by numerical experiments in Section 6.*

In what follows, we establish the convergence theorem of the proposed EJGI algorithm.

Theorem 4.1. *Suppose that the DTPS matrix equation (1) is consistent, i.e., the solution of the DTPS matrix equation (1) exists. Then the iterative sequences $\{Y_j(l)\}$ $(j = 1,\ldots,\gamma)$ generated by Algorithm 4.1 converge to the unique solution $Y^* = \left(Y_1^*, Y_2^*, \cdots, Y_\gamma^*\right)$ for any initial matrices $Y_j(0)$ $(j = 1,\ldots,\gamma)$, if the parameters μ and ω satisfy*

$$\begin{aligned}&(1-\omega)\sum_{j=1}^{\gamma}\left(\|I - \mu\omega D_{1,j}A_j\|_2 + \mu\omega\|D_{1,j-1}\|_2\|B_{j-1}\|_2\right)\\&+\omega\sum_{j=1}^{\gamma}\left[\|I - \mu(1-\omega)B_{j-1}D_{2,j-1}\|_2 + \mu(1-\omega)\|A_j\|_2\|D_{2,j}\|_2\right]p < 1,\end{aligned}$$

where

$$p = \sum_{j=1}^{\gamma} \left[\|I - \mu\omega\left(1-\omega\right) D_{1,j} A_j\|_2 + \mu\omega\left(1-\omega\right) \|D_{1,j-1}\|_2 \|B_{j-1}\|_2 \right].$$

Proof. By assumptions, we can prove that the solution of the DTPS matrix equation (1) is unique by applying the similar method utilized in Theorem 3.1. Let $Y^* = \left(Y_1^*, Y_2^*, \cdots, Y_\gamma^*\right)$ be the unique solution of the DTPS matrix equation (1). It follows from Algorithm 4.1 and the notations in (48) that

$$\begin{aligned} \tilde{Y}_{1,j}\left(l+1\right) &= \tilde{Y}_j\left(l\right) - \mu\omega D_{1,j}\left(A_j \tilde{Y}_j\left(l\right) + \tilde{Y}_{j+1}\left(l\right) B_j\right) \\ &= \tilde{Y}_j\left(l\right) - \mu\omega D_{1,j} A_j \tilde{Y}_j\left(l\right) - \mu\omega D_{1,j} \tilde{Y}_{j+1}\left(l\right) B_j, \end{aligned} \tag{61}$$

and

$$\begin{aligned} \tilde{Y}_{2,j}\left(l+1\right) &= \tilde{\hat{Y}}_j(l) - \mu(1-\omega)\left(A_{j-1}\tilde{\hat{Y}}_{j-1}(l) + \tilde{\hat{Y}}_j(l) B_{j-1}\right) D_{2,j-1} \\ &= \tilde{\hat{Y}}_j(l) - \mu(1-\omega) A_{j-1}\tilde{\hat{Y}}_{j-1}(l) D_{2,j-1} - \mu(1-\omega)\tilde{\hat{Y}}_j(l) B_{j-1} D_{2,j-1}. \end{aligned} \tag{62}$$

By combining (61) with (62), we have

$$\begin{aligned} \tilde{\hat{Y}}_j(l) &= \left(1-\omega\right)\tilde{Y}_{1,j}\left(l+1\right) + \omega\tilde{Y}_j\left(l\right) \\ &= \left(1-\omega\right)\left[\tilde{Y}_j\left(l\right) - \mu\omega D_{1,j} A_j \tilde{Y}_j\left(l\right) - \mu\omega D_{1,j}\tilde{Y}_{j+1}\left(l\right) B_j\right] + \omega\tilde{Y}_j\left(l\right) \\ &= \tilde{Y}_j\left(l\right) - \mu\omega\left(1-\omega\right) D_{1,j} A_j \tilde{Y}_j\left(l\right) - \mu\omega\left(1-\omega\right) D_{1,j}\tilde{Y}_{j+1}\left(l\right) B_j \\ &= \left[I - \mu\omega\left(1-\omega\right) D_{1,j} A_j\right]\tilde{Y}_j\left(l\right) - \mu\omega\left(1-\omega\right) D_{1,j}\tilde{Y}_{j+1}\left(l\right) B_j, \end{aligned} \tag{63}$$

and

$$\begin{aligned} \tilde{Y}_j\left(l+1\right) &= \left(1-\omega\right)\tilde{Y}_{1,j}\left(l+1\right) + \omega\tilde{Y}_{2,j}\left(l+1\right) \\ &= \left(1-\omega\right)\left[\tilde{Y}_j\left(l\right) - \mu\omega D_{1,j} A_j \tilde{Y}_j\left(l\right) - \mu\omega D_{1,j}\tilde{Y}_{j+1}\left(l\right) B_j\right] \\ &\quad + \omega[\tilde{\hat{Y}}_j(l) - \mu(1-\omega) A_{j-1}\tilde{\hat{Y}}_{j-1}(l) D_{2,j-1} - \mu(1-\omega)\tilde{\hat{Y}}_j(l) B_{j-1} D_{2,j-1}] \\ &= (1-\omega)\tilde{Y}_j\left(l\right) - \mu\omega(1-\omega) D_{1,j} A_j \tilde{Y}_j\left(l\right) - \mu\omega(1-\omega) D_{1,j}\tilde{Y}_{j+1}\left(l\right) B_j \\ &\quad + \omega\tilde{\hat{Y}}_j(l) - \mu\omega(1-\omega) A_{j-1}\tilde{\hat{Y}}_{j-1}(l) D_{2,j-1} - \mu\omega(1-\omega)\tilde{\hat{Y}}_j(l) B_{j-1} D_{2,j-1} \\ &= (1-\omega)[I - \mu\omega D_{1,j} A_j]\tilde{Y}_j(l) - \mu\omega(1-\omega) D_{1,j}\tilde{Y}_{j+1}(l) B_j \\ &\quad - \mu\omega(1-\omega) A_{j-1}\tilde{\hat{Y}}_{j-1}(l) D_{2,j-1} + \omega\tilde{\hat{Y}}_j(l)[I - \mu(1-\omega) B_{j-1} D_{2,j-1}]. \end{aligned} \tag{64}$$

By taking the 2-norm in (63) and (64), and using the properties of the matrix norm, it holds that

$$\begin{aligned} \left\|\tilde{\hat{Y}}_j(l)\right\|_2 &= \left\|\left[I - \mu\omega\left(1-\omega\right) D_{1,j} A_j\right]\tilde{Y}_j\left(l\right) - \mu\omega\left(1-\omega\right) D_{1,j}\tilde{Y}_{j+1}\left(l\right) B_j\right\|_2 \\ &\leq \|I - \mu\omega\left(1-\omega\right) D_{1,j} A_j\|_2 \left\|\tilde{Y}_j\left(l\right)\right\|_2 + \mu\omega\left(1-\omega\right)\|D_{1,j}\|_2 \|B_j\|_2 \left\|\tilde{Y}_{j+1}\left(l\right)\right\|_2, \end{aligned} \tag{65}$$

and

$$\begin{aligned} \left\|\tilde{Y}_j\left(l+1\right)\right\|_2 &= \Big\|(1-\omega)[I - \mu\omega D_{1,j} A_j]\tilde{Y}_j(l) - \mu\omega(1-\omega) D_{1,j}\tilde{Y}_{j+1}(l) B_j \\ &\quad - \mu\omega(1-\omega) A_{j-1}\tilde{\hat{Y}}_{j-1}(l) D_{2,j-1} + \omega\tilde{\hat{Y}}_j(l)[I - \mu(1-\omega) B_{j-1} D_{2,j-1}]\Big\|_2 \\ &\leq (1-\omega)\|I - \mu\omega D_{1,j} A_j\|_2 \left\|\tilde{Y}_j\left(l\right)\right\|_2 + \mu\omega(1-\omega)\|D_{1,j}\|_2 \|B_j\|_2 \left\|\tilde{Y}_{j+1}\left(l\right)\right\|_2 \\ &\quad + \omega\|I - \mu(1-\omega) B_{j-1} D_{2,j-1}\|_2 \left\|\tilde{\hat{Y}}_j(l)\right\|_2 + \mu\omega(1-\omega)\|A_{j-1}\|_2 \|D_{2,j-1}\|_2 \left\|\tilde{\hat{Y}}_{j-1}(l)\right\|_2. \end{aligned} \tag{66}$$

From (65), we can derive the following inequality

$$\sum_{j=1}^{\gamma}\left\|\tilde{\hat{Y}}_j(l)\right\|_2 \leq \sum_{j=1}^{\gamma}\|I - \mu\omega\left(1-\omega\right) D_{1,j} A_j\|_2 \left\|\tilde{Y}_j\left(l\right)\right\|_2 + \sum_{j=1}^{\gamma}\mu\omega\left(1-\omega\right)\|D_{1,j}\|_2 \|B_j\|_2 \left\|\tilde{Y}_{j+1}\left(l\right)\right\|_2$$

$$= \sum_{j=1}^{\gamma} \|I - \mu\omega (1-\omega) D_{1,j} A_j\|_2 \left\|\tilde{Y}_j (l)\right\|_2 + \sum_{j=1}^{\gamma} \mu\omega (1-\omega) \|D_{1,j-1}\|_2 \|B_{j-1}\|_2 \left\|\tilde{Y}_j (l)\right\|_2$$

$$= \sum_{j=1}^{\gamma} \left[\|I - \mu\omega (1-\omega) D_{1,j} A_j\|_2 + \mu\omega (1-\omega) \|D_{1,j-1}\|_2 \|B_{j-1}\|_2\right] \left\|\tilde{Y}_j (l)\right\|_2$$

$$\leq \sum_{j=1}^{\gamma} \left[\|I - \mu\omega (1-\omega) D_{1,j} A_j\|_2 + \mu\omega (1-\omega) \|D_{1,j-1}\|_2 \|B_{j-1}\|_2\right] \sum_{j=1}^{\gamma} \left\|\tilde{Y}_j (l)\right\|_2. \tag{67}$$

Let

$$p = \sum_{j=1}^{\gamma} \left[\|I - \mu\omega (1-\omega) D_{1,j} A_j\|_2 + \mu\omega (1-\omega) \|D_{1,j-1}\|_2 \|B_{j-1}\|_2\right],$$

then (67) can be written as

$$\sum_{j=1}^{\gamma} \left\|\tilde{\hat{Y}}_j(l)\right\|_2 \leq p \sum_{j=1}^{\gamma} \left\|\tilde{Y}_j (l)\right\|_2. \tag{68}$$

Below we construct a non-negative function $Z(l)$ as follows

$$Z(l) = \sum_{j=1}^{\gamma} \left\|\tilde{Y}_j (l)\right\|_2,$$

which together with (66) and (68) yields that

$$\begin{aligned}
Z(l+1) &= \sum_{j=1}^{\gamma} \left\|\tilde{Y}_j (l+1)\right\|_2 \\
&\leq \sum_{j=1}^{\gamma} (1-\omega) \|I - \mu\omega D_{1,j} A_j\|_2 \left\|\tilde{Y}_j (l)\right\|_2 + \sum_{j=1}^{\gamma} \mu\omega(1-\omega) \|D_{1,j}\|_2 \|B_j\|_2 \left\|\tilde{Y}_{j+1} (l)\right\|_2 \\
&\quad + \sum_{j=1}^{\gamma} \omega \|I - \mu(1-\omega) B_{j-1} D_{2,j-1}\|_2 \left\|\tilde{\hat{Y}}_j(l)\right\|_2 + \sum_{j=1}^{\gamma} \mu\omega(1-\omega) \|A_{j-1}\|_2 \|D_{2,j-1}\|_2 \left\|\tilde{\hat{Y}}_{j-1}(l)\right\|_2 \\
&= \sum_{j=1}^{\gamma} (1-\omega) \|I - \mu\omega D_{1,j} A_j\|_2 \left\|\tilde{Y}_j (l)\right\|_2 + \sum_{j=1}^{\gamma} \mu\omega(1-\omega) \|D_{1,j-1}\|_2 \|B_{j-1}\|_2 \left\|\tilde{Y}_j (l)\right\|_2 \\
&\quad + \sum_{j=1}^{\gamma} \omega \|I - \mu(1-\omega) B_{j-1} D_{2,j-1}\|_2 \left\|\tilde{\hat{Y}}_j(l)\right\|_2 + \sum_{j=1}^{\gamma} \mu\omega(1-\omega) \|A_j\|_2 \|D_{2,j}\|_2 \left\|\tilde{\hat{Y}}_j(l)\right\|_2 \\
&= (1-\omega) \sum_{j=1}^{\gamma} \left(\|I - \mu\omega D_{1,j} A_j\|_2 + \mu\omega \|D_{1,j-1}\|_2 \|B_{j-1}\|_2\right) \left\|\tilde{Y}_j (l)\right\|_2 \\
&\quad + \omega \sum_{j=1}^{\gamma} \left[\|I - \mu(1-\omega) B_{j-1} D_{2,j-1}\|_2 + \mu(1-\omega) \|A_j\|_2 \|D_{2,j}\|_2\right] \left\|\tilde{\hat{Y}}_j(l)\right\|_2 \\
&\leq (1-\omega) \sum_{j=1}^{\gamma} \left(\|I - \mu\omega D_{1,j} A_j\|_2 + \mu\omega \|D_{1,j-1}\|_2 \|B_{j-1}\|_2\right) \sum_{j=1}^{\gamma} \left\|\tilde{Y}_j (l)\right\|_2 \\
&\quad + \omega \sum_{j=1}^{\gamma} \left[\|I - \mu(1-\omega) B_{j-1} D_{2,j-1}\|_2 + \mu(1-\omega) \|A_j\|_2 \|D_{2,j}\|_2\right] \sum_{j=1}^{\gamma} \left\|\tilde{\hat{Y}}_j(l)\right\|_2 \\
&\leq (1-\omega) \sum_{j=1}^{\gamma} \left(\|I - \mu\omega D_{1,j} A_j\|_2 + \mu\omega \|D_{1,j-1}\|_2 \|B_{j-1}\|_2\right) \sum_{j=1}^{\gamma} \left\|\tilde{Y}_j (l)\right\|_2 \\
&\quad + \omega \sum_{j=1}^{\gamma} \left[\|I - \mu(1-\omega) B_{j-1} D_{2,j-1}\|_2 + \mu(1-\omega) \|A_j\|_2 \|D_{2,j}\|_2\right] p \sum_{j=1}^{\gamma} \left\|\tilde{Y}_j (l)\right\|_2 \\
&= \Bigg\{ (1-\omega) \sum_{j=1}^{\gamma} \left(\|I - \mu\omega D_{1,j} A_j\|_2 + \mu\omega \|D_{1,j-1}\|_2 \|B_{j-1}\|_2\right)
\end{aligned}$$

$$+\omega\sum_{j=1}^{\gamma}\left[\|I-\mu(1-\omega)B_{j-1}D_{2,j-1}\|_2+\mu(1-\omega)\|A_j\|_2\|D_{2,j}\|_2\right]p\Bigg\}\sum_{j=1}^{\gamma}\left\|\tilde{Y}_j(l)\right\|_2. \quad (69)$$

Let

$$Q=(1-\omega)\sum_{j=1}^{\gamma}\left(\|I-\mu\omega D_{1,j}A_j\|_2+\mu\omega\|D_{1,j-1}\|_2\|B_{j-1}\|_2\right)$$
$$+\,\omega\sum_{j=1}^{\gamma}\left[\|I-\mu(1-\omega)B_{j-1}D_{2,j-1}\|_2+\mu(1-\omega)\|A_j\|_2\|D_{2,j}\|_2\right]p. \quad (70)$$

Then (69) can be written as

$$Z(l+1)\le QZ(l),$$

which leads to

$$Z(l+1)\le QZ(l)\le Q^2Z(l-1)\le\cdots\le Q^{l+1}Z(0).$$

Thus if $Q<1$, then it holds that

$$\lim_{l\to+\infty}\sum_{j=1}^{\gamma}\left\|\tilde{Y}_j(l+1)\right\|_2=0,$$

and therefore $\lim_{l\to+\infty}\tilde{Y}_j(l+1)=0$ $(j=1,\cdots,\gamma)$, i.e.,

$$\lim_{l\to+\infty}Y_j(l+1)=Y_j^*,\ j=1,2,\cdots,\gamma.$$

The proof of this theorem is completed. ■

5. The MJGI algorithm for the GDTPS matrix equations

In this section, we consider the iterative solution of the following generalized DTPS (GDTPS) matrix equations

$$\sum_{s=1}^{p}A_{j,s}Y_jB_{j,s}+\sum_{v=1}^{q}E_{j,v}Y_{j+1}F_{j,v}=C_j,\ j=1,2,\cdots,\gamma, \quad (71)$$

where the known matrices $A_{j,s},E_{j,v}\in\mathbb{R}^{m\times m},B_{j,s},F_{j,v}\in\mathbb{R}^{n\times n},C_j\in\mathbb{R}^{m\times n}$ and the unknown matrices $Y_j\in\mathbb{R}^{m\times n}$ are periodic with period γ, i.e., $A_{j+\gamma,s}=A_{j,s},B_{j+\gamma,s}=B_{j,s},E_{j+\gamma,v}=E_{j,v},F_{j+\gamma,v}=F_{j,v},C_{j+\gamma}=C_j$ and $Y_{j+\gamma}=Y_j$.

First of all, we split the system matrices $A_{j,s},B_{j,s},E_{j,v},F_{j,v}$ $(j=1,\ldots,\gamma,s=1,\ldots,p,v=1,\ldots,q)$ of the GDTPS matrix equation (71) into the following forms:

$$A_{j,s}=D_{j,s}^{(1)}+R_{j,s}^{(1)},$$
$$B_{j,s}=D_{j,s}^{(2)}+R_{j,s}^{(2)},$$
$$E_{j,v}=D_{j,v}^{(3)}+R_{j,v}^{(3)},$$
$$F_{j,v}=D_{j,v}^{(4)}+R_{j,v}^{(4)},$$

where $D_{j,s}^{(1)},D_{j,s}^{(2)},D_{j,v}^{(3)},D_{j,v}^{(4)}$ are the diagonal parts of $A_{j,s},B_{j,s},E_{j,v},F_{j,v}$, respectively. In [39], Wang and Song also extended the JGI algorithm to solve the GDTPS matrix equations by replacing the coefficient matrices by their diagonal parts. Before developing a new algorithm for the GDTPS matrix equations, we review the JGI algorithm proposed in [39] for the GDTPS matrix equations as follows.

Algorithm 5.1. *The JGI algorithm for the GDTPS matrix equations* (71) *[39]:*

Step 1: Input matrices $A_{j,s},E_{j,v}\in\mathbb{R}^{m\times m},B_{j,s},F_{j,v}\in\mathbb{R}^{n\times n},C_j\in\mathbb{R}^{m\times n}$ *for* $j=1,\ldots,\gamma,s=1,\ldots,p,v=1,\ldots,q$*, and two constants* $\mu,\eta>0$*. Choose the initial matrices* $Y_j(0)\in\mathbb{R}^{m\times n}$ $(j=1,\ldots,\gamma)$*, and set* $l=0$*;*

Step 2: Take $Y_{j+\gamma}(0)=Y_j(0),A_{j+\gamma,s}=A_{j,s},B_{j+\gamma,s}=B_{j,s},E_{j+\gamma,v}=E_{j,v},F_{j+\gamma,v}=F_{j,v},C_{j+\gamma}=C_j,D_{j+\gamma,s}^{(1)}=D_{j,s}^{(1)},D_{j+\gamma,s}^{(2)}=D_{j,s}^{(2)},D_{j+\gamma,v}^{(3)}=D_{j,v}^{(3)}$ *and* $D_{j+\gamma,v}^{(4)}=D_{j,v}^{(4)}$*;*

Step 3: If $\xi_l = \sqrt{\frac{\sum_{j=1}^{\gamma}\|C_j-\sum_{s=1}^{p}A_{j,s}Y_j(l)B_{j,s}-\sum_{v=1}^{q}E_{j,v}Y_{j+1}(l)F_{j,v}\|^2}{\sum_{j=1}^{\gamma}\|C_j-\sum_{s=1}^{p}A_{j,s}Y_j(0)B_{j,s}-\sum_{v=1}^{q}E_{j,v}Y_{j+1}(0)F_{j,v}\|^2}} < \eta$, *then stop; otherwise, go to Step 4;*

Step 4: For $l = 0, 1, 2, \ldots$ *and* $j = 1, \ldots, \gamma$, *calculate*

$$Y_{1,j}(l+1) = Y_j(l) + \mu \sum_{s=1}^{p} D_{j,s}^{(1)} \left[C_j - \left(\sum_{k=1}^{p} A_{j,k}Y_j(l)B_{j,k} + \sum_{t=1}^{q} E_{j,t}Y_{j+1}(l)F_{j,t}\right)\right] D_{j,s}^{(2)},$$

$$Y_{2,j}(l+1) = Y_j(l) + \mu \sum_{v=1}^{q} D_{j-1,v}^{(3)} \left[C_{j-1} - \left(\sum_{k=1}^{p} A_{j-1,k}Y_{j-1}(l)B_{j-1,k} + \sum_{t=1}^{q} E_{j-1,t}Y_j(l)F_{j-1,t}\right)\right] D_{j-1,v}^{(4)},$$

$$Y_j(l+1) = \frac{Y_{1,j}(l+1)+Y_{2,j}(l+1)}{2},$$

$$Y_{j+\gamma}(l+1) = Y_j(l+1).$$

Step 5: Set $l := l+1$ *and return to Step 3.*

In the following, we apply the update strategy to Algorithm 5.1 and then propose the modified Jacobi gradient based iterative (MJGI) algorithm for the GDTPS matrix equations. The details are presented as follows.

It can be observed that $Y_{1,j}(l+1)$ is computed by $Y_j(l)$ and $Y_{j+1}(l)$ $(j = 1, \ldots, \gamma)$. Then for $j = \gamma$, we can calculate $Y_{1,\gamma}(l+1)$ by $Y_\gamma(l)$ and $Y_{\gamma+1}(l) = Y_1(l)$. Note that when we compute $Y_{1,\gamma}(l+1)$, the matrix $Y_1(l+1)$ has been determined. To improve the convergence rate of the JGI algorithm, motivated by the ideas of the FGI algorithm in [41] and the Gauss-Seidel iteration method, we replace $Y_1(l)$ by the latest information $Y_1(l+1)$ to compute $Y_{1,\gamma}(l+1)$. In addition, we see that $Y_{2,j}(l+1)$ are computed by $Y_{j-1}(l)$ and $Y_j(l)$ $(j = 1, \ldots, \gamma)$. Also, for $j = 2, 3, \cdots, \gamma$, when we compute $Y_{2,\gamma}(l+1)$, the matrices $Y_{j-1}(l+1)$ have been obtained. Similar to the above analysis, in second line of Step 4 of the JGI algorithm, $Y_{j-1}(l)$ are replaced by $Y_{j-1}(l+1)$ to compute $Y_{2,j}(l+1)$. By summarizing the above discussions, we can establish the following modified JGI (MJGI) algorithm for the GDTPS matrix equations (71).

Algorithm 5.2. *The modified Jacobi gradient based iterative (MJGI) algorithm for the GDTPS matrix equations* (71)*:*

Step 1: Input matrices $A_{j,s}, E_{j,v} \in \mathbb{R}^{m\times m}, B_{j,s}, F_{j,t} \in \mathbb{R}^{n\times n}, C_j \in \mathbb{R}^{m\times n}$ *for* $j = 1, \ldots, \gamma, s = 1, \ldots, p, v = 1, \ldots, q$, *and two constants* $\mu, \eta > 0$. *Choose the initial matrices* $Y_j(0) \in \mathbb{R}^{m\times n}$ $(j = 1, \ldots, \gamma)$, *and set* $l = 0$;

Step 2: Take $Y_{j+\gamma}(0) = Y_j(0), A_{j+\gamma,s} = A_{j,s}, B_{j+\gamma,s} = B_{j,s}, E_{j+\gamma,v} = E_{j,v}, F_{j+\gamma,v} = F_{j,v}, C_{j+\gamma} = C_j, D_{j+\gamma,s}^{(1)} = D_{j,s}^{(1)}, D_{j+\gamma,s}^{(2)} = D_{j,s}^{(2)}, D_{j+\gamma,v}^{(3)} = D_{j,v}^{(3)}$ *and* $D_{j+\gamma,v}^{(4)} = D_{j,v}^{(4)}$;

Step 3: If $\xi_l = \sqrt{\frac{\sum_{j=1}^{\gamma}\|C_j-\sum_{s=1}^{p}A_{j,s}Y_j(l)B_{j,s}-\sum_{v=1}^{q}E_{j,v}Y_{j+1}(l)F_{j,v}\|^2}{\sum_{j=1}^{\gamma}\|C_j-\sum_{s=1}^{p}A_{j,s}Y_j(0)B_{j,s}-\sum_{v=1}^{q}E_{j,v}Y_{j+1}(0)F_{j,v}\|^2}} < \eta$, *then stop; otherwise, go to Step 4;*

Step 4: For $l = 0, 1, 2, \ldots$ *and* $j = 1, \ldots, \gamma$, *calculate*

$$Y_{1,j}(l+1) = \begin{cases} Y_j(l) + \mu \sum_{s=1}^{p} D_{j,s}^{(1)} \left[C_j - \left(\sum_{k=1}^{p} A_{j,k}Y_j(l)B_{j,k} + \sum_{t=1}^{q} E_{j,t}Y_{j+1}(l)F_{j,t}\right)\right] D_{j,s}^{(2)}, j = 1, 2, \cdots, \gamma-1. \\ Y_j(l) + \mu \sum_{s=1}^{p} D_{j,s}^{(1)} \left[C_j - \left(\sum_{k=1}^{p} A_{j,k}Y_j(l)B_{j,k} + \sum_{t=1}^{q} E_{j,t}Y_{j+1}(l+1)F_{j,t}\right)\right] D_{j,s}^{(2)}, j = \gamma, \end{cases}$$

$$Y_{2,j}(l+1) = \begin{cases} Y_j(l) + \mu \sum_{v=1}^{q} D_{j-1,v}^{(3)} \left[C_{j-1} - \left(\sum_{k=1}^{p} A_{j-1,k}Y_{j-1}(l)B_{j-1,k} + \sum_{t=1}^{q} E_{j-1,t}Y_j(l)F_{j-1,t}\right)\right] D_{j-1,v}^{(4)}, j = 1. \\ Y_j(l) + \mu \sum_{v=1}^{q} D_{j-1,v}^{(3)} \left[C_{j-1} - \left(\sum_{k=1}^{p} A_{j-1,k}Y_{j-1}(l+1)B_{j-1,k} + \sum_{t=1}^{q} E_{j-1,t}Y_j(l)F_{j-1,t}\right)\right] D_{j-1,v}^{(4)}, j = 2, 3, \cdots, \gamma, \end{cases}$$

$$Y_j(l+1) = \frac{Y_{1,j}(l+1)+Y_{2,j}(l+1)}{2}, j = 1, 2, \cdots, \gamma,$$

$$Y_{j+\gamma}(l+1) = Y_j(l+1), j = 1, 2, \cdots, \gamma.$$

Step 5: Set $l := l+1$ *and return to Step 3.*

Below we derive the necessary and sufficient condition for the convergence of the MJGI algorithm by utilizing the properties of the vector stretching operator and the Kronecker product of two matrices. To this end, we define the following two matrices:

$$L=\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0\\ c_1 & 0 & 0 & \cdots & 0 & 0\\ 0 & c_2 & 0 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & c_{\gamma-2} & 0 & 0\\ b_\gamma & 0 & 0 & 0 & c_{\gamma-1} & 0 \end{bmatrix},\ H=\begin{bmatrix} a_1 & b_1 & 0 & 0 & \cdots & c_\gamma\\ 0 & a_2 & b_2 & 0 & \cdots & 0\\ 0 & 0 & a_3 & b_3 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & a_{\gamma-1} & b_{\gamma-1}\\ 0 & 0 & 0 & 0 & \cdots & a_\gamma \end{bmatrix} \tag{72}$$

with

$$\begin{aligned} a_j &= \sum_{s=1}^{p}\sum_{k=1}^{p} D_{j,s}^{(2)} B_{j,k}^T \otimes D_{j,s}^{(1)} A_{j,k} + \sum_{v=1}^{q}\sum_{t=1}^{q} D_{j-1,v}^{(4)} F_{j-1,t}^T \otimes D_{j-1,v}^{(3)} E_{j-1,t},\\ b_j &= \sum_{s=1}^{p}\sum_{t=1}^{q}\left(D_{j,s}^{(2)} F_{j,t}^T \otimes D_{j,s}^{(1)} E_{j,t}\right),\\ c_j &= \sum_{v=1}^{q}\sum_{k=1}^{p}\left(D_{j,v}^{(4)} B_{j,k}^T \otimes D_{j,v}^{(3)} A_{j,k}\right),\ j=1,\ldots,\gamma. \end{aligned}$$

Theorem 5.1. *Assume that the GDTPS matrix equation* (71) *is consistent, i.e., the solution of the GDTPS matrix equation* (71) *exists. Then the iterative sequences* $\{Y_j(l)\}(j=1,\ldots,\gamma)$ *generated by Algorithm 5.2 converge to the unique solution* $Y^*=\left(Y_1^*,Y_2^*,\cdots,Y_\gamma^*\right)$ *for any initial matrices* $Y_j(0)$ $(j=1,\ldots,\gamma)$ *if and only if the parameter* μ *satisfies*

$$\rho\left[\left(I+\frac{\mu}{2}L\right)^{-1}\left(I-\frac{\mu}{2}H\right)\right]<1.$$

Proof. By assumptions, we can prove that the solution of the GDTPS matrix equation (71) is unique by applying the similar method utilized in Theorem 3.2. Let $Y^*=\left(Y_1^*,Y_2^*,\cdots,Y_\gamma^*\right)$ be the unique solution of the GDTPS matrix equation (71). According to (48), we have

$$\tilde{Y}_{1,j}(l)=Y_{1,j}(l)-Y_j^*,\tilde{Y}_{2,j}(l)=Y_{2,j}(l)-Y_j^*,\tilde{Y}_j(l)=Y_j(l)-Y_j^*,\ j=1,\ldots,\gamma. \tag{73}$$

Based on Algorithm 5.2, we distinguish the following cases to discuss:

- when $j=1$, it has

$$Y_{1,1}(l+1)=Y_1(l)+\mu\sum_{s=1}^{p} D_{1,s}^{(1)}\left[C_1-\left(\sum_{k=1}^{p} A_{1,k} Y_1(l) B_{1,k}+\sum_{t=1}^{q} E_{1,t} Y_2(l) F_{1,t}\right)\right] D_{1,s}^{(2)}, \tag{74}$$

$$Y_{2,1}(l+1)=Y_1(l)+\mu\sum_{v=1}^{q} D_{\gamma,v}^{(3)}\left[C_\gamma-\left(\sum_{k=1}^{p} A_{\gamma,k} Y_\gamma(l) B_{\gamma,k}+\sum_{t=1}^{q} E_{\gamma,t} Y_1(l) F_{\gamma,t}\right)\right] D_{\gamma,v}^{(4)}, \tag{75}$$

$$Y_1(l+1)=\frac{Y_{1,1}(l+1)+Y_{2,1}(l+1)}{2}. \tag{76}$$

Then it follows from (73)–(76) that

$$\begin{aligned} \tilde{Y}_1(l+1) &= \frac{\tilde{Y}_{1,1}(l+1)+\tilde{Y}_{2,1}(l+1)}{2}\\ &= \tilde{Y}_1(l)-\frac{\mu}{2}\left[\sum_{s=1}^{p} D_{1,s}^{(1)}\left(\sum_{k=1}^{p} A_{1,k}\tilde{Y}_1(l) B_{1,k}+\sum_{t=1}^{q} E_{1,t}\tilde{Y}_2(l) F_{1,t}\right) D_{1,s}^{(2)}\right.\\ &\quad\left.+\sum_{v=1}^{q} D_{\gamma,v}^{(3)}\left(\sum_{k=1}^{p} A_{\gamma,k}\tilde{Y}_\gamma(l) B_{\gamma,k}+\sum_{t=1}^{q} E_{\gamma,t}\tilde{Y}_1(l) F_{\gamma,t}\right) D_{\gamma,v}^{(4)}\right]\\ &= \tilde{Y}_1(l)-\frac{\mu}{2}\sum_{s=1}^{p}\sum_{k=1}^{p} D_{1,s}^{(1)} A_{1,k}\tilde{Y}_1(l) B_{1,k} D_{1,s}^{(2)}-\frac{\mu}{2}\sum_{s=1}^{p}\sum_{t=1}^{q} D_{1,s}^{(1)} E_{1,t}\tilde{Y}_2(l) F_{1,t} D_{1,s}^{(2)}\\ &\quad-\frac{\mu}{2}\sum_{v=1}^{q}\sum_{k=1}^{p} D_{\gamma,v}^{(3)} A_{\gamma,k}\tilde{Y}_\gamma(l) B_{\gamma,k} D_{\gamma,v}^{(4)}-\frac{\mu}{2}\sum_{v=1}^{q}\sum_{t=1}^{q} D_{\gamma,v}^{(3)} E_{\gamma,t}\tilde{Y}_1(l) F_{\gamma,t} D_{\gamma,v}^{(4)}. \end{aligned} \tag{77}$$

By taking vector straightening operator on both sides of (77), we get

$$\operatorname{vec}\left[\tilde{Y}_1(l+1)\right]=\left[I_{mn}-\frac{\mu}{2}\left(\sum_{s=1}^{p}\sum_{k=1}^{p} D_{1,s}^{(2)} B_{1,k}^T\otimes D_{1,s}^{(1)} A_{1,k}+\sum_{v=1}^{q}\sum_{t=1}^{q} D_{\gamma,v}^{(4)} F_{\gamma,t}^T\otimes D_{\gamma,v}^{(3)} E_{\gamma,t}\right)\right]\operatorname{vec}\left[\tilde{Y}_1(l)\right]$$

$$-\frac{\mu}{2}\sum_{s=1}^{p}\sum_{t=1}^{q}\left(D_{1,s}^{(2)}F_{1,t}^{T}\otimes D_{1,s}^{(1)}E_{1,t}\right)\mathrm{vec}\left[\tilde{Y}_{2}\left(l\right)\right]-\frac{\mu}{2}\sum_{v=1}^{q}\sum_{k=1}^{p}\left(D_{\gamma,v}^{(4)}B_{\gamma,k}^{T}\otimes D_{\gamma,v}^{(3)}A_{\gamma,k}\right)\mathrm{vec}\left[\tilde{Y}_{\gamma}\left(l\right)\right], \tag{78}$$

in view of Lemma 2.1.

- when $j=2,\cdots,\gamma-1$, it holds that

$$Y_{1,j}\left(l+1\right)=Y_{j}\left(l\right)+\mu\sum_{s=1}^{p}D_{j,s}^{(1)}\left[C_{j}-\left(\sum_{k=1}^{p}A_{j,k}Y_{j}\left(l\right)B_{j,k}+\sum_{t=1}^{q}E_{j,t}Y_{j+1}\left(l\right)F_{j,t}\right)\right]D_{j,s}^{(2)}, \tag{79}$$

$$Y_{2,j}\left(l+1\right)=Y_{j}\left(l\right)+\mu\sum_{v=1}^{q}D_{j-1,v}^{(3)}\left[C_{j-1}-\left(\sum_{k=1}^{p}A_{j-1,k}Y_{j-1}\left(l+1\right)B_{j-1,k}+\sum_{t=1}^{q}E_{j-1,t}Y_{j}\left(l\right)F_{j-1,t}\right)\right]D_{j-1,v}^{(4)}, \tag{80}$$

$$Y_{j}\left(l+1\right)=\frac{Y_{1,j}\left(l+1\right)+Y_{2,j}\left(l+1\right)}{2}. \tag{81}$$

Then from (73) and (79)–(81), straightforward computations show that

$$\begin{aligned}
\tilde{Y}_{j}\left(l+1\right)&=\frac{\tilde{Y}_{1,j}\left(l+1\right)+\tilde{Y}_{2,j}\left(l+1\right)}{2}\\
&=\tilde{Y}_{j}\left(l\right)-\frac{\mu}{2}\left[\sum_{s=1}^{p}D_{j,s}^{(1)}\left(\sum_{k=1}^{p}A_{j,k}\tilde{Y}_{j}\left(l\right)B_{j,k}+\sum_{t=1}^{q}E_{j,t}\tilde{Y}_{j+1}\left(l\right)F_{j,t}\right)D_{j,s}^{(2)}\right.\\
&\quad\left.+\sum_{v=1}^{q}D_{j-1,v}^{(3)}\left(\sum_{k=1}^{p}A_{j-1,k}\tilde{Y}_{j-1}\left(l+1\right)B_{j-1,k}+\sum_{t=1}^{q}E_{j-1,t}\tilde{Y}_{j}\left(l\right)F_{j-1,t}\right)D_{j-1,v}^{(4)}\right]\\
&=\tilde{Y}_{j}\left(l\right)-\frac{\mu}{2}\sum_{s=1}^{p}\sum_{k=1}^{p}D_{j,s}^{(1)}A_{j,k}\tilde{Y}_{j}\left(l\right)B_{j,k}D_{j,s}^{(2)}-\frac{\mu}{2}\sum_{s=1}^{p}\sum_{t=1}^{q}D_{j,s}^{(1)}E_{j,t}\tilde{Y}_{j+1}\left(l\right)F_{j,t}D_{j,s}^{(2)}\\
&\quad-\frac{\mu}{2}\sum_{v=1}^{q}\sum_{k=1}^{p}D_{j-1,v}^{(3)}A_{j-1,k}\tilde{Y}_{j-1}\left(l+1\right)B_{j-1,k}D_{j-1,v}^{(4)}-\frac{\mu}{2}\sum_{v=1}^{q}\sum_{t=1}^{q}D_{j-1,v}^{(3)}E_{j-1,t}\tilde{Y}_{j}\left(l\right)F_{j-1,t}D_{j-1,v}^{(4)}.
\end{aligned} \tag{82}$$

Using vector straightening operator on both sides of relation (82) and applying Lemma 2.1 yield that

$$\begin{aligned}
&\mathrm{vec}\left[\tilde{Y}_{j}\left(l+1\right)\right]\\
&=\left[I-\frac{\mu}{2}\left(\sum_{s=1}^{p}\sum_{k=1}^{p}D_{j,s}^{(2)}B_{j,k}^{T}\otimes D_{j,s}^{(1)}A_{j,k}+\sum_{v=1}^{q}\sum_{v=1}^{q}D_{j-1,v}^{(4)}F_{j-1,t}^{T}\otimes D_{j-1,v}^{(3)}E_{j-1,t}\right)\right]\mathrm{vec}\left[\tilde{Y}_{j}\left(l\right)\right]\\
&\quad-\frac{\mu}{2}\sum_{s=1}^{p}\sum_{t=1}^{q}\left(D_{j,s}^{(2)}F_{j,t}^{T}\otimes D_{j,s}^{(1)}E_{j,t}\right)\mathrm{vec}\left[\tilde{Y}_{j+1}\left(l\right)\right]-\frac{\mu}{2}\sum_{v=1}^{q}\sum_{k=1}^{p}\left(D_{j-1,v}^{(4)}B_{j-1,k}^{T}\otimes D_{j-1,v}^{(3)}A_{j-1,k}\right)\mathrm{vec}\left[\tilde{Y}_{j-1}\left(l+1\right)\right],
\end{aligned}$$

which is equivalent to

$$\begin{aligned}
&\frac{\mu}{2}\sum_{v=1}^{q}\sum_{k=1}^{p}\left(D_{j-1,v}^{(4)}B_{j-1,k}^{T}\otimes D_{j-1,v}^{(3)}A_{j-1,k}\right)\mathrm{vec}\left[\tilde{Y}_{j-1}\left(l+1\right)\right]+\mathrm{vec}\left[\tilde{Y}_{j}\left(l+1\right)\right]\\
&=\left[I-\frac{\mu}{2}\left(\sum_{s=1}^{p}\sum_{k=1}^{p}D_{j,s}^{(2)}B_{j,k}^{T}\otimes D_{j,s}^{(1)}A_{j,k}+\sum_{v=1}^{q}\sum_{t=1}^{q}D_{j-1,v}^{(4)}F_{j-1,t}^{T}\otimes D_{j-1,v}^{(3)}E_{j-1,t}\right)\right]\mathrm{vec}\left[\tilde{Y}_{j}\left(l\right)\right]\\
&\quad-\frac{\mu}{2}\sum_{s=1}^{p}\sum_{t=1}^{q}\left(D_{j,s}^{(2)}F_{j,t}^{T}\otimes D_{j,s}^{(1)}E_{j,t}\right)\mathrm{vec}\left[\tilde{Y}_{j+1}\left(l\right)\right],\ j=2,\ldots,\gamma-1.
\end{aligned} \tag{83}$$

- when $j=\gamma$, it follows that

$$Y_{1,\gamma}\left(l+1\right)=Y_{\gamma}\left(l\right)+\mu\sum_{s=1}^{p}D_{\gamma,s}^{(1)}\left[C_{\gamma}-\left(\sum_{k=1}^{p}A_{\gamma,k}Y_{\gamma}\left(l\right)B_{\gamma,k}+\sum_{t=1}^{q}E_{\gamma,t}Y_{1}\left(l+1\right)F_{\gamma,t}\right)\right]D_{\gamma,s}^{(2)}, \tag{84}$$

$$Y_{2,\gamma}(l+1) = Y_\gamma(l) + \mu \sum_{v=1}^{q} D^{(3)}_{\gamma-1,v} \left[C_{\gamma-1} - \left(\sum_{k=1}^{p} A_{\gamma-1,k} Y_{\gamma-1}(l+1) B_{\gamma-1,k} + \sum_{t=1}^{q} E_{\gamma-1,t} Y_\gamma(l) F_{\gamma-1,t} \right) \right] D^{(4)}_{\gamma-1,v}, \tag{85}$$

$$Y_\gamma(l+1) = \frac{Y_{1,\gamma}(l+1) + Y_{2,\gamma}(l+1)}{2}. \tag{86}$$

Then the combination of (73) and (84)–(86) results in

$$\begin{aligned}
\tilde{Y}_\gamma(l+1) &= \frac{\tilde{Y}_{1,\gamma}(l+1) + \tilde{Y}_{2,\gamma}(l+1)}{2} \\
&= \tilde{Y}_\gamma(l) - \frac{\mu}{2} \left[\sum_{s=1}^{p} D^{(1)}_{\gamma,s} \left(\sum_{k=1}^{p} A_{\gamma,k} \tilde{Y}_\gamma(l) B_{\gamma,k} + \sum_{t=1}^{q} E_{\gamma,t} \tilde{Y}_1(l+1) F_{\gamma,t} \right) D^{(2)}_{\gamma,s} \right. \\
&\quad \left. + \sum_{v=1}^{q} D^{(3)}_{\gamma-1,v} \left(\sum_{k=1}^{p} A_{\gamma-1,k} \tilde{Y}_{\gamma-1}(l+1) B_{\gamma-1,k} + \sum_{t=1}^{q} E_{\gamma-1,t} \tilde{Y}_\gamma(l) F_{\gamma-1,t} \right) D^{(4)}_{\gamma-1,v} \right] \\
&= \tilde{Y}_\gamma(l) - \frac{\mu}{2} \sum_{s=1}^{p} \sum_{k=1}^{p} D^{(1)}_{\gamma,s} A_{\gamma,k} \tilde{Y}_\gamma(l) B_{\gamma,k} D^{(2)}_{\gamma,s} - \frac{\mu}{2} \sum_{s=1}^{p} \sum_{t=1}^{q} D^{(1)}_{\gamma,s} E_{\gamma,t} \tilde{Y}_1(l+1) F_{\gamma,t} D^{(2)}_{\gamma,s} \\
&\quad - \frac{\mu}{2} \sum_{v=1}^{q} \sum_{k=1}^{p} D^{(3)}_{\gamma-1,v} A_{\gamma-1,k} \tilde{Y}_{\gamma-1}(l+1) B_{\gamma-1,k} D^{(4)}_{\gamma-1,v} - \frac{\mu}{2} \sum_{v=1}^{q} \sum_{t=1}^{q} D^{(3)}_{\gamma-1,v} E_{\gamma-1,t} \tilde{Y}_\gamma(l) F_{\gamma-1,t} D^{(4)}_{\gamma-1,v}.
\end{aligned} \tag{87}$$

By applying the vector stretching function to (87) and according to Lemma 2.1, it has

$$\begin{aligned}
&\mathrm{vec}\left[\tilde{Y}_\gamma(l+1)\right] \\
&= \left[I - \frac{\mu}{2} \left(\sum_{s=1}^{p} \sum_{k=1}^{p} D^{(2)}_{\gamma,s} B^T_{\gamma,k} \otimes D^{(1)}_{\gamma,s} A_{\gamma,k} + \sum_{v=1}^{q} \sum_{t=1}^{q} D^{(4)}_{\gamma-1,v} F^T_{\gamma-1,t} \otimes D^{(3)}_{\gamma-1,v} E_{\gamma-1,t} \right) \right] \mathrm{vec}\left[\tilde{Y}_\gamma(l)\right] \\
&\quad - \frac{\mu}{2} \sum_{s=1}^{p} \sum_{t=1}^{q} \left(D^{(2)}_{\gamma,s} F^T_{\gamma,t} \otimes D^{(1)}_{\gamma,s} E_{\gamma,t} \right) \mathrm{vec}\left[\tilde{Y}_1(l+1)\right] - \frac{\mu}{2} \sum_{v=1}^{q} \sum_{k=1}^{p} \left(D^{(4)}_{\gamma-1,v} B^T_{\gamma-1,k} \otimes D^{(3)}_{\gamma-1,v} A_{\gamma-1,k} \right) \mathrm{vec}\left[\tilde{Y}_{\gamma-1}(l+1)\right],
\end{aligned} \tag{88}$$

which can be rewritten into the following equivalent form

$$\begin{aligned}
&\frac{\mu}{2} \sum_{s=1}^{p} \sum_{t=1}^{q} \left(D^{(2)}_{\gamma,s} F^T_{\gamma,t} \otimes D^{(1)}_{\gamma,s} E_{\gamma,t} \right) \mathrm{vec}\left[\tilde{Y}_1(l+1)\right] \\
&\quad + \frac{\mu}{2} \sum_{v=1}^{q} \sum_{k=1}^{p} \left(D^{(4)}_{\gamma-1,v} B^T_{\gamma-1,k} \otimes D^{(3)}_{\gamma-1,v} A_{\gamma-1,k} \right) \mathrm{vec}\left[\tilde{Y}_{\gamma-1}(l+1)\right] + \mathrm{vec}\left[\tilde{Y}_\gamma(l+1)\right] \\
&= \left[I - \frac{\mu}{2} \left(\sum_{s=1}^{p} \sum_{k=1}^{p} D^{(2)}_{\gamma,s} B^T_{\gamma,k} \otimes D^{(1)}_{\gamma,s} A_{\gamma,k} + \sum_{v=1}^{q} \sum_{t=1}^{q} D^{(4)}_{\gamma-1,v} F^T_{\gamma-1,t} \otimes D^{(3)}_{\gamma-1,v} E_{\gamma-1,t} \right) \right] \mathrm{vec}\left[\tilde{Y}_\gamma(l)\right].
\end{aligned} \tag{89}$$

In view of (78), (83) and (89), we deduce that

$$\left(I + \frac{\mu}{2} L \right) \begin{bmatrix} \mathrm{vec}\left[\tilde{Y}_1(l+1)\right] \\ \mathrm{vec}\left[\tilde{Y}_2(l+1)\right] \\ \vdots \\ \mathrm{vec}\left[\tilde{Y}_{\gamma-1}(l+1)\right] \\ \mathrm{vec}\left[\tilde{Y}_\gamma(l+1)\right] \end{bmatrix} = \left(I - \frac{\mu}{2} H \right) \begin{bmatrix} \mathrm{vec}\left[\tilde{Y}_1(l)\right] \\ \mathrm{vec}\left[\tilde{Y}_2(l)\right] \\ \vdots \\ \mathrm{vec}\left[\tilde{Y}_{\gamma-1}(l)\right] \\ \mathrm{vec}\left[\tilde{Y}_\gamma(l)\right] \end{bmatrix}, \tag{90}$$

where the matrices L and H are defined as in (72). It is evident that $I+\frac{\mu}{2}L$ is a nonsingular matrix, then it follows from (90) that

$$\begin{bmatrix} \operatorname{vec}\left[\tilde{Y}_1(l+1)\right] \\ \operatorname{vec}\left[\tilde{Y}_2(l+1)\right] \\ \vdots \\ \operatorname{vec}\left[\tilde{Y}_{\gamma-1}(l+1)\right] \\ \operatorname{vec}\left[\tilde{Y}_{\gamma}(l+1)\right] \end{bmatrix} = \left(I+\frac{\mu}{2}L\right)^{-1}\left(I-\frac{\mu}{2}H\right) \begin{bmatrix} \operatorname{vec}\left[\tilde{Y}_1(l)\right] \\ \operatorname{vec}\left[\tilde{Y}_2(l)\right] \\ \vdots \\ \operatorname{vec}\left[\tilde{Y}_{\gamma-1}(l)\right] \\ \operatorname{vec}\left[\tilde{Y}_{\gamma}(l)\right] \end{bmatrix}, \tag{91}$$

and the matrix $\left(I+\frac{\mu}{2}L\right)^{-1}\left(I-\frac{\mu}{2}H\right)$ is the iteration matrix of the MJGI algorithm. Therefore, the MJGI algorithm is convergent if and only if the parameter μ satisfies

$$\rho\left[\left(I+\frac{\mu}{2}L\right)^{-1}\left(I-\frac{\mu}{2}H\right)\right]<1,$$

which completes the proof of this theorem. ■

6. Numerical experiments

This section provides several numerical examples to validate the effectiveness and advantages of the proposed algorithms, and compare their numerical performances with those of the GI, JGI and AJGI ones, with respect to the number of iteration steps (IT) and the elapsed time in seconds (CPU). All numerical experiments are computed in MATLAB (version R2018b) on a personal computer with AMD Ryzen 7 5800H, CPU 3.20 GHz and 16.0 GB memory.

Example 6.1. *Consider the discrete-time periodic Sylvester (DTPS) matrix equations*

$$A_jY_j+Y_{j+1}B_j=C_j,\ \ j=1,2,3,$$

with the following coefficient matrices:

$$A_1=\begin{bmatrix} 2.7 & 0.9 \\ -1.1 & 2.3 \end{bmatrix}\otimes I_{200}+I_{200}\otimes\begin{bmatrix} 2.7 & 0.9 \\ -1.1 & 2.3 \end{bmatrix},\ A_2=\begin{bmatrix} 4.2 & 1.3 \\ -1.9 & 3.8 \end{bmatrix}\otimes I_{200}+I_{200}\otimes\begin{bmatrix} 4.2 & 1.3 \\ -1.9 & 3.8 \end{bmatrix},$$

$$A_3=\begin{bmatrix} 6.1 & 3.8 \\ -3.1 & 6.3 \end{bmatrix}\otimes I_{200}+I_{200}\otimes\begin{bmatrix} 6.1 & 3.8 \\ -3.1 & 6.3 \end{bmatrix},\ B_1=\begin{bmatrix} 1.5 & -0.2 \\ 0.4 & 1.0 \end{bmatrix}\otimes I_{200}+I_{200}\otimes\begin{bmatrix} 1.5 & -0.2 \\ 0.4 & 1.0 \end{bmatrix},$$

$$B_2=\begin{bmatrix} 2.1 & -0.4 \\ 0.4 & 2.0 \end{bmatrix}\otimes I_{200}+I_{200}\otimes\begin{bmatrix} 2.1 & -0.4 \\ 0.4 & 2.0 \end{bmatrix},\ B_3=\begin{bmatrix} 3.1 & -0.6 \\ 0.7 & 3.5 \end{bmatrix}\otimes I_{200}+I_{200}\otimes\begin{bmatrix} 3.1 & -0.6 \\ 0.7 & 3.5 \end{bmatrix},$$

$$C_1=\begin{bmatrix} 13.2 & 10.6 \\ 0.6 & 8.4 \end{bmatrix}\otimes I_{200}+I_{200}\otimes\begin{bmatrix} 13.2 & 10.6 \\ 0.6 & 8.4 \end{bmatrix},\ C_2=\begin{bmatrix} 26.4 & 21.2 \\ 1.2 & 16.8 \end{bmatrix}\otimes I_{200}+I_{200}\otimes\begin{bmatrix} 26.4 & 21.2 \\ 1.2 & 16.8 \end{bmatrix},$$

$$C_3=\begin{bmatrix} 38.6 & 32.1 \\ 1.6 & 24.2 \end{bmatrix}\otimes I_{200}+I_{200}\otimes\begin{bmatrix} 38.6 & 32.1 \\ 1.6 & 24.2 \end{bmatrix}.$$

In our computations, the initial matrices are taken to be

$$Y_j(0)=10^{-6}\times I_{400}, j=1,2,3,$$

and all iterations are terminated once

$$RES=\sqrt{\frac{\|C_1-A_1Y_1(l)-Y_2(l)B_1\|^2+\|C_2-A_2Y_2(l)-Y_3(l)B_2\|^2+\|C_3-A_3Y_3(l)-Y_1(l)B_3\|^2}{\|C_1\|^2+\|C_2\|^2+\|C_3\|^2}}\le\eta$$

with η being a positive number, or l reaches the maximal number of iteration steps $l_{\max}=10000$.

For all tested algorithms, their parameters are the experimentally found optimal ones which minimize their IT. And the experimental optimal parameters, IT, CPU time and RES of the GI, JGI, AJGI and EJGI algorithms for Example 6.1 with respect to five different values of η are listed in Table 1. Comparing the numerical results of Table 1, we see that all tested algorithms can successfully compute approximate solutions satisfying the prescribed stopping criterion, and their IT and CPU time increase gradually with decreasing of η. Meanwhile, the proposed

EJGI algorithm performs better than the GI, JGI and AJGI ones in terms of both the IT and CPU time. The IT and CPU time of the EJGI algorithms are less than half of those for the GI one, and are almost one half of those of the JGI one. Additionally, the proposed EJGI algorithm is more stable than the other ones in view of IT, due to the fact that the variational range of the IT of the former one is smaller than those of the latter ones. Finally, the numerical results in Table 1 show that the new updated technique applied in the EJGI algorithm can improve the convergence speed of the AJGI and JGI ones effectively, and the EJGI algorithm has higher computational efficiency than the AJGI and JGI ones.

To better show the superiority of the proposed EJGI algorithm, RES(log10) of four tested algorithms with respect to IT are depicted in Figure 1 for four different values of η. It follows from Figure 1 that all tested algorithms are convergent, and the EJGI algorithm has advantages over the other ones in view of IT, because it requires less IT to achieve the termination criterion. Additionally, the advantage of the EJGI algorithm becomes more pronounced as the value of η decreases. This further confirms that the superiority of the EJGI algorithm for solving the discrete-time periodic Sylvester matrix equations. These conclusions are in accordance with the results of Table 1, and indicate that the convergent speed of the EJGI algorithm is the fastest among the tested algorithms.

Table 1: IT, CPU and RES of four GI-like algorithms for Example 6.1 with five values of η

Algorithm		η				
		10^{-11}	10^{-12}	10^{-13}	10^{-14}	10^{-15}
GI	IT	193	213	233	254	274
$\mu = 1.32e-02$	CPU	4.2756	4.7080	5.1694	5.6498	6.1270
	RES	9.7007e-12	9.8186e-13	9.9392e-14	8.9720e-15	9.0839e-16
JGI	IT	167	184	201	218	235
$\mu = 1.37e-02$	CPU	3.6882	3.9536	4.3340	4.6930	5.1178
	RES	9.6503e-12	9.5642e-13	9.3266e-14	9.0767e-15	9.0251e-16
AJGI	IT	94	103	112	122	131
$\mu = 5.4e-02$	CPU	2.0608	2.2297	2.4185	2.6734	2.8348
$\omega = \frac{1}{4}$	RES	8.9108e-12	9.1837e-13	9.5810e-14	7.9475e-15	8.4493e-16
EJGI	IT	84	91	99	106	114
$\mu = 9.1e-02$	CPU	1.7813	1.9564	2.1095	2.3178	2.4503
$\omega = \frac{1}{6}$	RES	9.3393e-12	8.8473e-13	7.7208e-14	9.6760e-15	9.2025e-16

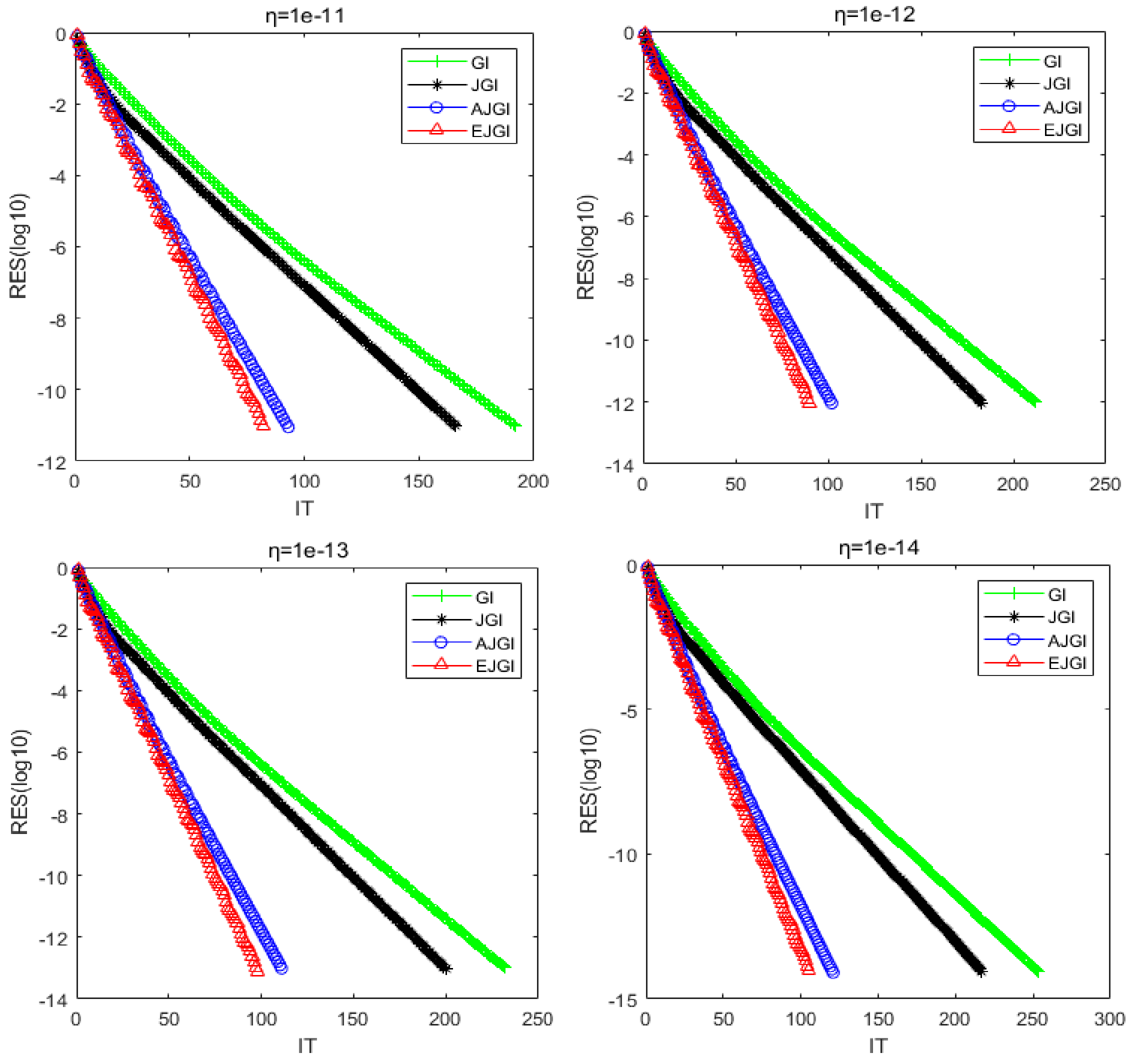

Figure 1: Comparisons for the convergence curves of four GI-like algorithms for Example 6.1.

Example 6.2. *Consider the discrete-time periodic Sylvester (DTPS) matrix equations*

$$A_jY_j + Y_{j+1}B_j = C_j,\ \ j = 1, 2, 3,$$

where the coefficient matrices are as follows:

$$\begin{aligned}
A_1 &= I_2 \otimes G_1 + 0.022E_2 \otimes triu(ones(5,5)) + 0.015E_1 \otimes tril(ones(5,5)),\\
A_2 &= I_2 \otimes G_2 + 0.013E_2 \otimes triu(ones(5,5)) + 0.020E_1 \otimes tril(ones(5,5)),\\
A_3 &= I_2 \otimes G_3 + 0.016E_2 \otimes tril(ones(5,5)) + 0.013E_1 \otimes triu(ones(5,5)),\\
B_1 &= I_2 \otimes H_1 + 0.012E_2 \otimes tril(ones(5,5)) + 0.016E_1 \otimes triu(ones(5,5)),\\
B_2 &= I_2 \otimes H_2 + 0.011E_2 \otimes triu(ones(5,5)) + 0.021E_1 \otimes tril(ones(5,5)),\\
B_3 &= I_2 \otimes H_3 + 0.014E_2 \otimes tril(ones(5,5)) + 0.015E_1 \otimes triu(ones(5,5)),\\
C_1 &= I_2 \otimes T_1 + 0.52E_2 \otimes triu(ones(5,5)) + 0.31E_1 \otimes tril(ones(5,5)),\\
C_2 &= I_2 \otimes T_2 + 0.29E_2 \otimes triu(ones(5,5)) + 0.34E_1 \otimes tril(ones(5,5)),\\
C_3 &= I_2 \otimes T_3 + 0.54E_2 \otimes tril(ones(5,5)) + 0.41E_1 \otimes tril(ones(5,5)),
\end{aligned}$$

with

$$G_1 = \begin{bmatrix} 3.0725 & 0.0975 & 0.1576 & 0.1419 & 0.6557 \\ 0.9058 & 1.8103 & 0.9706 & 0.4218 & 0.0357 \\ 0.1270 & 0.5469 & 2.7743 & 0.9157 & 0.8491 \\ 0.9134 & 0.9575 & 0.4854 & 4.0874 & 0.9340 \\ 0.6324 & 0.9649 & 0.8003 & 0.9595 & 2.9334 \end{bmatrix}, G_2 = \begin{bmatrix} 2.1122 & 0.3517 & 0.2858 & 0.0759 & 0.1299 \\ 0.1966 & 2.6938 & 0.7572 & 0.0540 & 0.5688 \\ 0.2511 & 0.5853 & 2.5827 & 0.5308 & 0.4694 \\ 0.6160 & 0.5497 & 0.3804 & 2.4573 & 0.0119 \\ 0.4733 & 0.9172 & 0.5678 & 0.9340 & 2.7544 \end{bmatrix},$$

$$G_3 = \begin{bmatrix} -7.4617 & 0.9200 & 0.1939 & 0.5488 & 0.6273 \\ 0.0199 & -1.7766 & 0.9048 & 0.9316 & 0.6991 \\ 0.4199 & 0.3678 & -7.2374 & 0.3352 & 0.3972 \\ 0.7597 & 0.6208 & 0.6318 & -6.4845 & 0.4136 \\ 0.7939 & 0.7313 & -0.2344 & 0.3919 & -2.4036 \end{bmatrix}, H_1 = \begin{bmatrix} 0.1529 & 0.7621 & 0.6154 & 0.4057 & 0.0579 \\ -0.2311 & 0.1033 & 0.7919 & 0.9005 & -0.3529 \\ 0.0068 & 0.0185 & 0.0470 & -0.0169 & 0.5132 \\ 0.2860 & 0.0214 & 0.2382 & -1.0898 & -0.0099 \\ 0.8913 & 0.4447 & 0.1763 & 0.8936 & 1.0085 \end{bmatrix},$$

$$H_2 = \begin{bmatrix} 0.0962 & 0.6979 & 0 & 0 & 0.0010 \\ 0.6822 & 0.3353 & 0.3998 & 0 & 0 \\ 0.1028 & 0.8600 & 0.0740 & 0.2897 & 0 \\ 0.5417 & 0.8537 & 0.6449 & -0.5403 & 0.5681 \\ 0.1509 & 0.4936 & -0.8180 & 0.5341 & -0.3587 \end{bmatrix}, H_3 = \begin{bmatrix} 0.2536 & 0.1259 & 0 & 0 & 0 \\ 0.2235 & 0.1233 & 0.1798 & 0 & 0 \\ 0.5155 & 0.6604 & 0.0513 & -0.0592 & 0 \\ 0.3340 & 0.5298 & 0.6808 & 0.1317 & 0.0150 \\ -0.4329 & 0.5405 & 0.4611 & 0.0503 & 0.0431 \end{bmatrix},$$

$$T_1 = \begin{bmatrix} -5.7240 & 0.4984 & 0.7513 & 0.9593 & 0.8407 \\ 0.6797 & -5.0403 & 0.2551 & 0.5472 & 0.2543 \\ 0.6551 & 0.3404 & -5.4940 & 0.1386 & 0.8143 \\ 0.1626 & 0.5853 & 0.6991 & -5.8507 & 0.2435 \\ 0.1190 & 0.2238 & 0.8909 & 0.2575 & -5.0707 \end{bmatrix}, T_2 = \begin{bmatrix} -5.7240 & 0.4984 & 0.7513 & 0.9593 & 0.8407 \\ 0.6797 & -5.0403 & 0.2551 & 0.5472 & 0.2543 \\ 0.6551 & 0.3404 & -5.4940 & 0.1386 & 0.8143 \\ 0.1626 & 0.5853 & 0.6991 & -5.8507 & 0.2435 \\ 0.1190 & 0.2238 & 0.8909 & 0.2575 & -5.0707 \end{bmatrix},$$

$$T_3 = \begin{bmatrix} -5.7240 & 0.4984 & 0.7513 & 0.9593 & 0.8407 \\ 0.6797 & -5.0403 & 0.2551 & 0.5472 & 0.2543 \\ 0.6551 & 0.3404 & -5.4940 & 0.1386 & 0.8143 \\ 0.1626 & 0.5853 & 0.6991 & -5.8507 & 0.2435 \\ 0.1190 & 0.2238 & 0.8909 & 0.2575 & -5.0707 \end{bmatrix}, E_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, E_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

In this example, we choose the initial matrices to be

$$Y_j(0) = 10^{-6} \times I_{10}, j = 1, 2, 3$$

and adopt the termination criterion as in Example 6.1, i.e.,

$$RES = \sqrt{\frac{\|C_1 - A_1Y_1(l) - Y_2(l)B_1\|^2 + \|C_2 - A_2Y_2(l) - Y_3(l)B_2\|^2 + \|C_3 - A_3Y_3(l) - Y_1(l)B_3\|^2}{\|C_1\|^2 + \|C_2\|^2 + \|C_3\|^2}} \leq \eta,$$

with $\eta > 0$ or l exceeds the prescribed maximal number of iteration steps 10000.

As in Example 6.1, the parameters adopted in the GI, JGI, AJGI and EJGI algorithms for Example 6.2 are the experimentally found optimal ones, which are obtained experimentally by minimizing the corresponding iteration steps. In Table 2, we list the parameters, IT, CPU time and RES of the tested algorithms for Example 6.2 with five different values of η. According to the numerical results in Table 2, we can conclude some observations: Firstly, all tested algorithms are valid for all cases. Secondly, the IT of all tested algorithms are increasing with the decreasing of η. Thirdly, the numerical performances of the JGI and the AJGI algorithms are comparable, and they outperform the GI one with respect to computational efficiency. Fourthly, among the tested algorithms, the proposed EJGI algorithm performs the best in view of IT and CPU time, and the advantage becomes more pronounced as η decreases. Besides, the IT and CPU time of the EJGI algorithm are almost one in ten of those for the GI one. Finally, the EJGI algorithm is the most stable among the tested algorithms, because the variational range of IT of the EJGI algorithm is the smallest compared with other tested ones. In summary, the EJGI algorithm has higher computational efficiency than the GI, JGI and AJGI ones, and applying the new updated technique to the JGI one can ameliorate the convergence speeds and efficiencies of the GI, JGI and AJGI ones.

In Figure 2, we compare the RES(log10) curves of the GI, JGI, AJGI and EJGI algorithms in terms of IT with $\eta = 10^{-13}$ and $\eta = 10^{-14}$. It can be seen from Figure 2 that the IT of the JGI, AJGI and EJGI algorithms are far less than that of the GI one. This indicates that the JGI, AJGI and EJGI algorithms have faster convergence rates than the GI one, which coincides with the results in Table 2. To further confirm the effectiveness of the proposed EJGI algorithm compared with the JGI and AJGI ones, the graphs of RES(log10) against number of iterations for four different values of η are displayed in Figure 3. By observation, we find that among these tested algorithms, the EJGI one is the most effective algorithm as its residual reduces the fastest, and the advantage of the EJGI algorithm becomes more pronounced as the value of η decreases. This is consistent with the results in Table 2.

Table 2: IT, CPU and RES of four GI-like algorithms for Example 6.2 with five values of η

Algorithm		η				
		10^{-10}	10^{-11}	10^{-12}	10^{-13}	10^{-14}
GI	IT	6587	7339	8090	8841	9595
$\mu = 5.988e-02$	CPU	0.2202	0.2751	0.2821	0.3060	0.3232
	RES	9.9984e-11	9.9697e-12	9.9718e-13	9.9795e-14	9.9910e-15
JGI	IT	731	813	896	978	1061
$\mu = 6.82e-02$	CPU	0.0222	0.0333	0.0399	0.0438	0.0344
	RES	9.8601e-11	9.9699e-12	9.8119e-13	9.9302e-14	9.7611e-15
AJGI	IT	724	806	887	969	1050
$\mu = 7.565e-01$	CPU	0.0222	0.0320	0.0357	0.0345	0.0342
$\omega = \frac{1}{22}$	RES	9.9059e-11	9.7627e-12	9.9075e-13	9.7742e-14	9.8805e-15
EJGI	IT	682	759	835	912	988
$\mu = 2.175e-1$	CPU	0.0207	0.0284	0.0279	0.0254	0.0271
$\omega = \frac{1}{1.1}$	RES	9.8787e-11	9.7136e-12	9.8593e-13	9.7236e-14	9.8030e-15

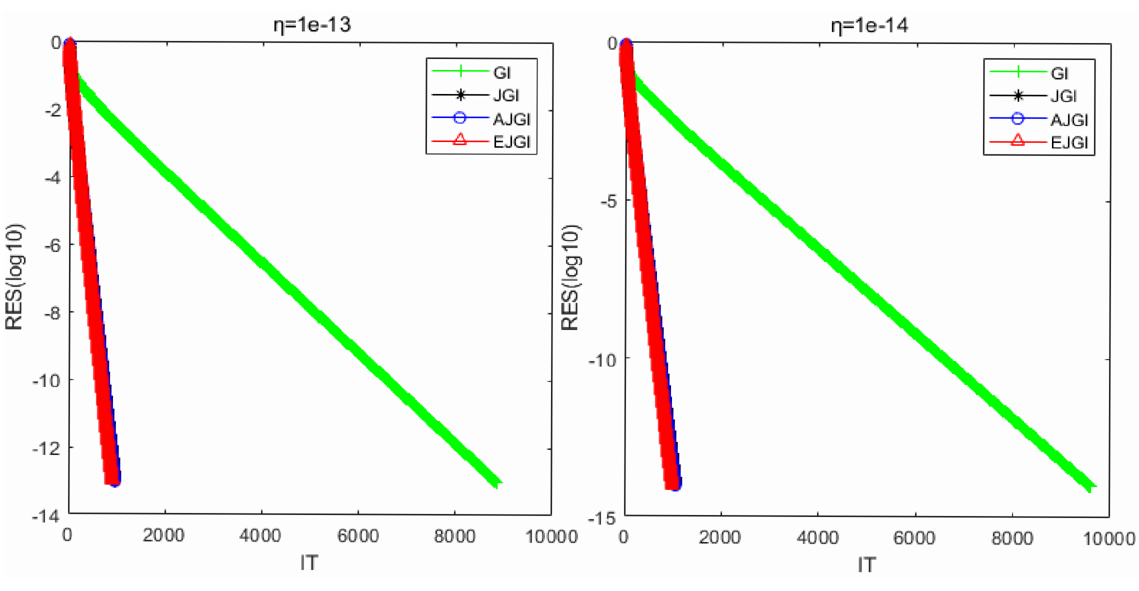

Figure 2: Comparisons for the convergence curves of four GI-like algorithms for Example 6.1 with $\eta = 10^{-13}$ and $\eta = 10^{-14}$.

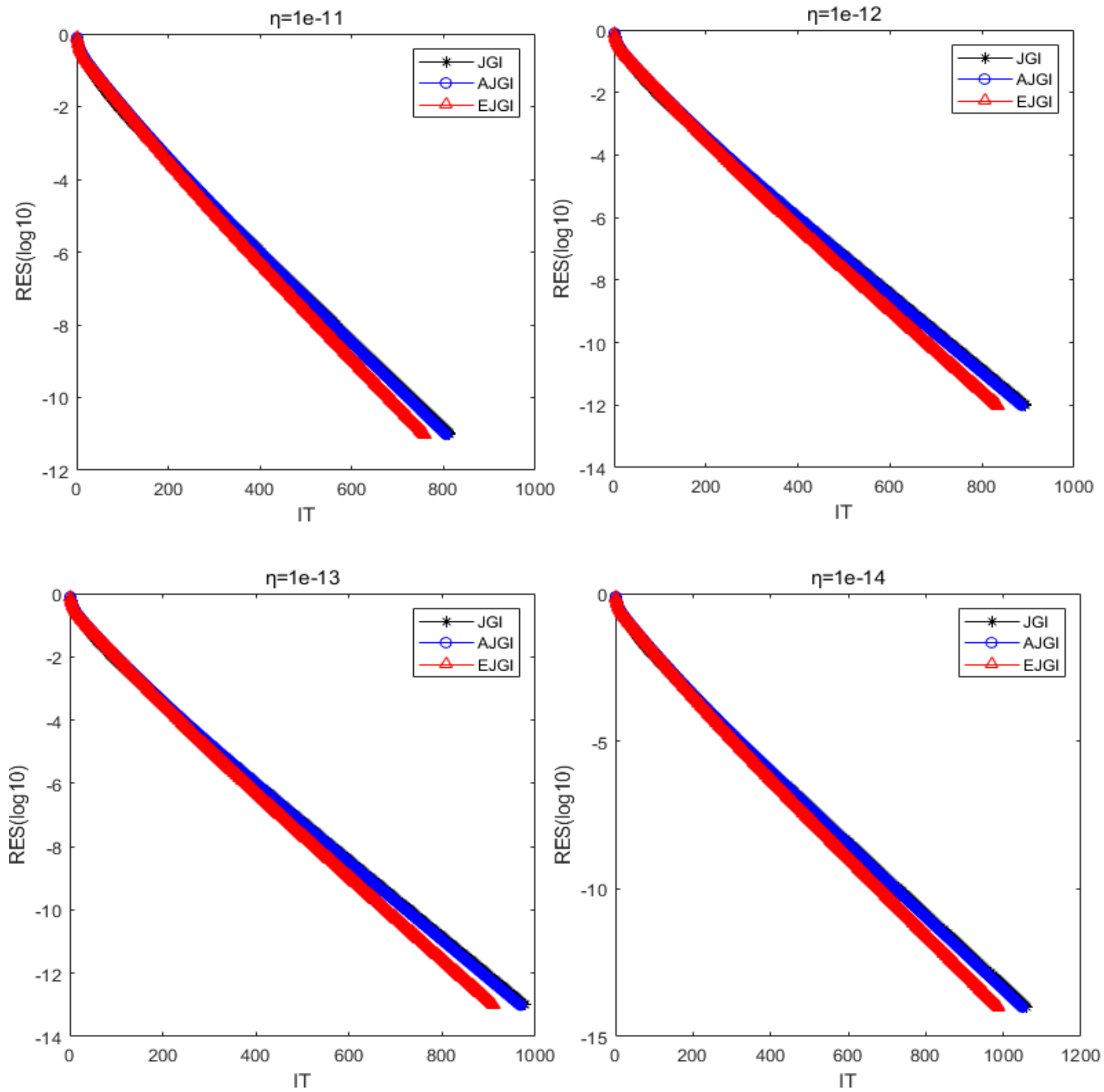

Figure 3: Comparisons for the convergence curves of the JGI, AJGI and EJGI algorithms for Example 6.2.

Example 6.3. *Consider the generalized discrete-time periodic Sylvester (GDTPS) matrix equations*

$$\sum_{s=1}^{2} A_{j,s}Y_jB_{j,s} + \sum_{v=1}^{2} E_{j,v}Y_{j+1}F_{j,v} = C_j, j = 1, 2, 3,$$

where the coefficient matrices are as follows

$$\begin{aligned}
A_{11} &= I_{20} \otimes G_{11} + 0.022E_2 \otimes triu(ones(30, 30)) + 0.015E_3 \otimes tril(ones(30, 30)),\\
A_{12} &= I_{20} \otimes G_{12} + 0.012E_1 \otimes triu(ones(30, 30)) + 0.017E_4 \otimes tril(ones(30, 30)),\\
A_{21} &= I_{20} \otimes G_{21} + 0.011E_2 \otimes triu(ones(30, 30)) + 0.021E_4 \otimes tril(ones(30, 30)),\\
A_{22} &= I_{20} \otimes G_{22} + 0.012E_1 \otimes triu(ones(30, 30)) + 0.025E_2 \otimes tril(ones(30, 30)),\\
A_{31} &= I_{20} \otimes G_{31} + 0.031E_3 \otimes triu(ones(30, 30)) + 0.008E_4 \otimes tril(ones(30, 30)),\\
A_{32} &= I_{20} \otimes G_{32} + 0.024E_2 \otimes triu(ones(30, 30)) + 0.013E_1 \otimes tril(ones(30, 30)),\\
B_{11} &= I_{20} \otimes H_{11} + 0.020E_4 \otimes triu(ones(30, 30)) + 0.018E_3 \otimes tril(ones(30, 30)),\\
B_{12} &= I_{20} \otimes H_{12} + 0.032E_4 \otimes triu(ones(30, 30)) + 0.011E_1 \otimes tril(ones(30, 30)),\\
B_{21} &= I_{20} \otimes H_{21} + 0.017E_3 \otimes triu(ones(30, 30)) + 0.025E_2 \otimes tril(ones(30, 30)),\\
B_{22} &= I_{20} \otimes H_{22} + 0.026E_3 \otimes triu(ones(30, 30)) + 0.035E_1 \otimes tril(ones(30, 30)),\\
B_{31} &= I_{20} \otimes H_{31} + 0.032E_2 \otimes triu(ones(30, 30)) + 0.025E_4 \otimes tril(ones(30, 30)),
\end{aligned}$$

$$B_{32}=I_{20}\otimes H_{32}+0.012E_3\otimes triu(ones(30,30))+0.033E_1\otimes tril(ones(30,30)),$$
$$E_{11}=T_{11}\otimes I_{20}+0.012E_1\otimes triu(ones(30,30))+0.025E_2\otimes tril(ones(30,30)),$$
$$E_{12}=T_{12}\otimes I_{20}+0.032E_3\otimes triu(ones(30,30))+0.012E_4\otimes tril(ones(30,30)),$$
$$E_{21}=T_{21}\otimes I_{20}+0.022E_3\otimes triu(ones(30,30))+0.045E_1\otimes tril(ones(30,30)),$$
$$E_{22}=T_{22}\otimes I_{20}+0.023E_4\otimes triu(ones(30,30))+0.015E_2\otimes tril(ones(30,30)),$$
$$E_{31}=T_{31}\otimes I_{20}+0.019E_2\otimes triu(ones(30,30))+0.021E_4\otimes tril(ones(30,30)),$$
$$E_{32}=T_{32}\otimes I_{20}+0.021E_1\otimes triu(ones(30,30))+0.018E_3\otimes tril(ones(30,30)),$$
$$F_{11}=W_{11}\otimes I_{20}+0.023E_2\otimes triu(ones(30,30))+0.015E_1\otimes tril(ones(30,30)),$$
$$F_{12}=W_{12}\otimes I_{20}+0.012E_4\otimes triu(ones(30,30))+0.012E_3\otimes tril(ones(30,30)),$$
$$F_{21}=W_{21}\otimes I_{20}+0.017E_1\otimes triu(ones(30,30))+0.015E_3\otimes tril(ones(30,30)),$$
$$F_{22}=W_{22}\otimes I_{20}+0.013E_2\otimes triu(ones(30,30))+0.015E_4\otimes tril(ones(30,30)),$$
$$F_{31}=W_{31}\otimes I_{20}+0.029E_4\otimes triu(ones(30,30))+0.011E_2\otimes tril(ones(30,30)),$$
$$F_{32}=W_{32}\otimes I_{20}+0.011E_3\otimes triu(ones(30,30))+0.028E_1\otimes tril(ones(30,30)),$$
$$C_1=I_{20}\otimes V_1+V_1\otimes I_{20},\ C_2=I_{20}\otimes V_2+V_2\otimes I_{20},\ C_3=I_{20}\otimes V_3+V_3\otimes I_{20},$$

with

$$G_{11}=\begin{bmatrix}3.2796&0&0\\0.9058&0&0.5469\\0.1270&0.0975&3.3732\end{bmatrix},\ G_{12}=\begin{bmatrix}2.6844&0.0357&0.6787\\0.9595&2.3810&0.7577\\0.9595&0&0\end{bmatrix},\ G_{21}=\begin{bmatrix}0&0.0206&0.1140\\0.0478&0&0.3962\\0.5940&0.8986&-1.0405\end{bmatrix},$$

$$G_{22}=\begin{bmatrix}0.1339&-0.5163&-0.1176\\-0.1176&1.0520&-0.1478\\-0.6505&-0.6618&0.2441\end{bmatrix},G_{31}=\begin{bmatrix}-8&0.0838&0.3524\\0.7482&-8.4872&0.8258\\0.4&0.9133&-7.9728\end{bmatrix},G_{32}=\begin{bmatrix}0&0.8001&0\\0&3.5764&0\\0.2599&0.8&3.6588\end{bmatrix},$$

$$H_{11}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},H_{12}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},H_{21}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},H_{22}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},H_{31}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},$$

$$H_{32}=\begin{bmatrix}1&0&0\\0&0.9&0\\0&0&1\end{bmatrix},T_{11}=\begin{bmatrix}5&0&0\\0&5&0\\0&0&5\end{bmatrix},T_{12}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},T_{21}=\begin{bmatrix}15&0&0\\0&15&0\\0&0&15\end{bmatrix},T_{22}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},$$

$$T_{31}=\begin{bmatrix}8&0&0\\0&8&0\\0&4&8\end{bmatrix},T_{32}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},W_{11}=\begin{bmatrix}0&0.03&0\\0&4.6934&0\\0.9502&0&1.8495\end{bmatrix},W_{12}=\begin{bmatrix}-4.7&0&0.7513\\0.4984&-2.0892&0.2551\\0.9597&0.2238&-2.6312\end{bmatrix},$$

$$W_{21}=\begin{bmatrix}-1.5165&0.3500&0.6160\\0.2435&-1.5892&0\\0.9293&0.2511&-1.7162\end{bmatrix},W_{22}=\begin{bmatrix}0&0.7792&0.5688\\0.0540&-1.2131&0.4694\\0.53&0.1299&0\end{bmatrix},W_{31}=\begin{bmatrix}-3.1652&-0.4357&-0.4302\\-0.1707&-1.5373&-0.1848\\-0.2277&-0.9234&-3.0385\end{bmatrix},$$

$$W_{32}=\begin{bmatrix}0.28&0.31&0.0855\\0.11&0&0.2625\\0.2967&0&1.8946\end{bmatrix},V_1=\begin{bmatrix}10.6009&0.9114&2.9006\\50.9111&19.5182&16.3448\\48.8167&6.5514&7.1615\end{bmatrix},V_2=\begin{bmatrix}3.0733&-8.6223&0.4267\\-0.7762&-3.2363&-13.3910\\-19.4331&4.4601&10.7858\end{bmatrix},$$

$$V_3=\begin{bmatrix}-0.7251&-7.3488&3.0989\\-1.7189&-19.4865&-31.2162\\-46.5468&5.0599&5.4487\end{bmatrix},E_1=\begin{bmatrix}0&1\\0&0\end{bmatrix},E_2=\begin{bmatrix}0&0\\1&0\end{bmatrix},E_3=\begin{bmatrix}0&0\\1&0\end{bmatrix},E_4=\begin{bmatrix}0&0\\0&1\end{bmatrix}.$$

For this example, we adopt

$$Y_j(0)=10^{-6}\times I_{60}, j=1,2,3,$$

as the initial matrices for all tested algorithms, and all iterations are terminated once

$$RES=\sqrt{\frac{r(l+1)}{r(0)}}<\delta,$$

with δ being a positive constant and

$$r(l)=\sum_{j=1}^{3}\left\|C_j-\sum_{s=1}^{2}A_{j,s}Y_j(l)B_{j,s}-\sum_{v=1}^{2}E_{j,v}Y_{j+1}(l)F_{j,v}\right\|^2,$$

or the number of iteration steps l reaches the prescribed maximal number of iteration steps $l_{\max}=10000$. And the latter case is marked by “Fail” and “–” in tables.

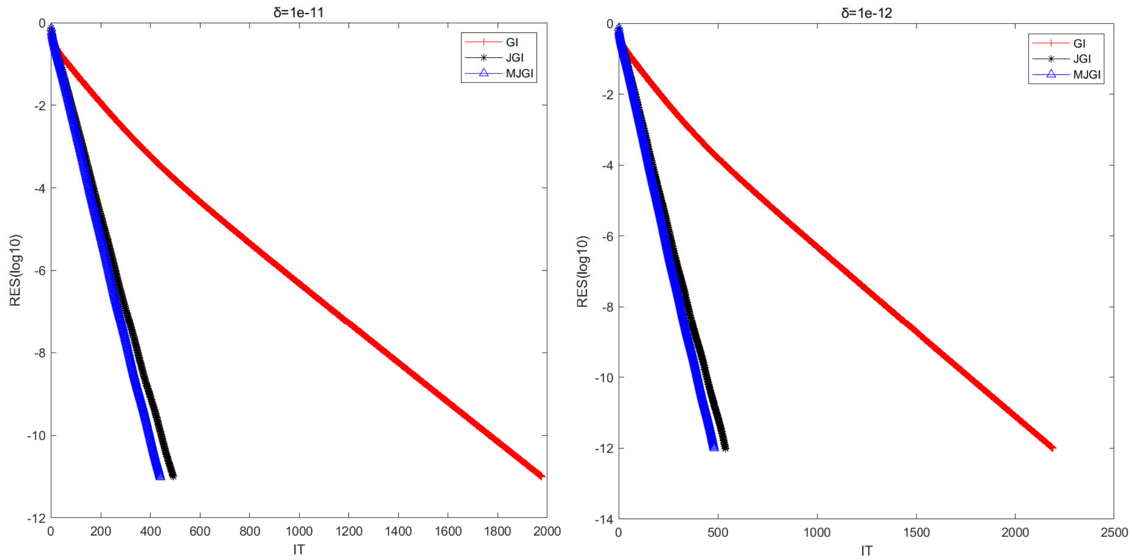

Figure 4: Comparisons for the convergence curves of three GI-like algorithms for Example 6.3 with $\delta = 10^{-11}$ and $\delta = 10^{-12}$.

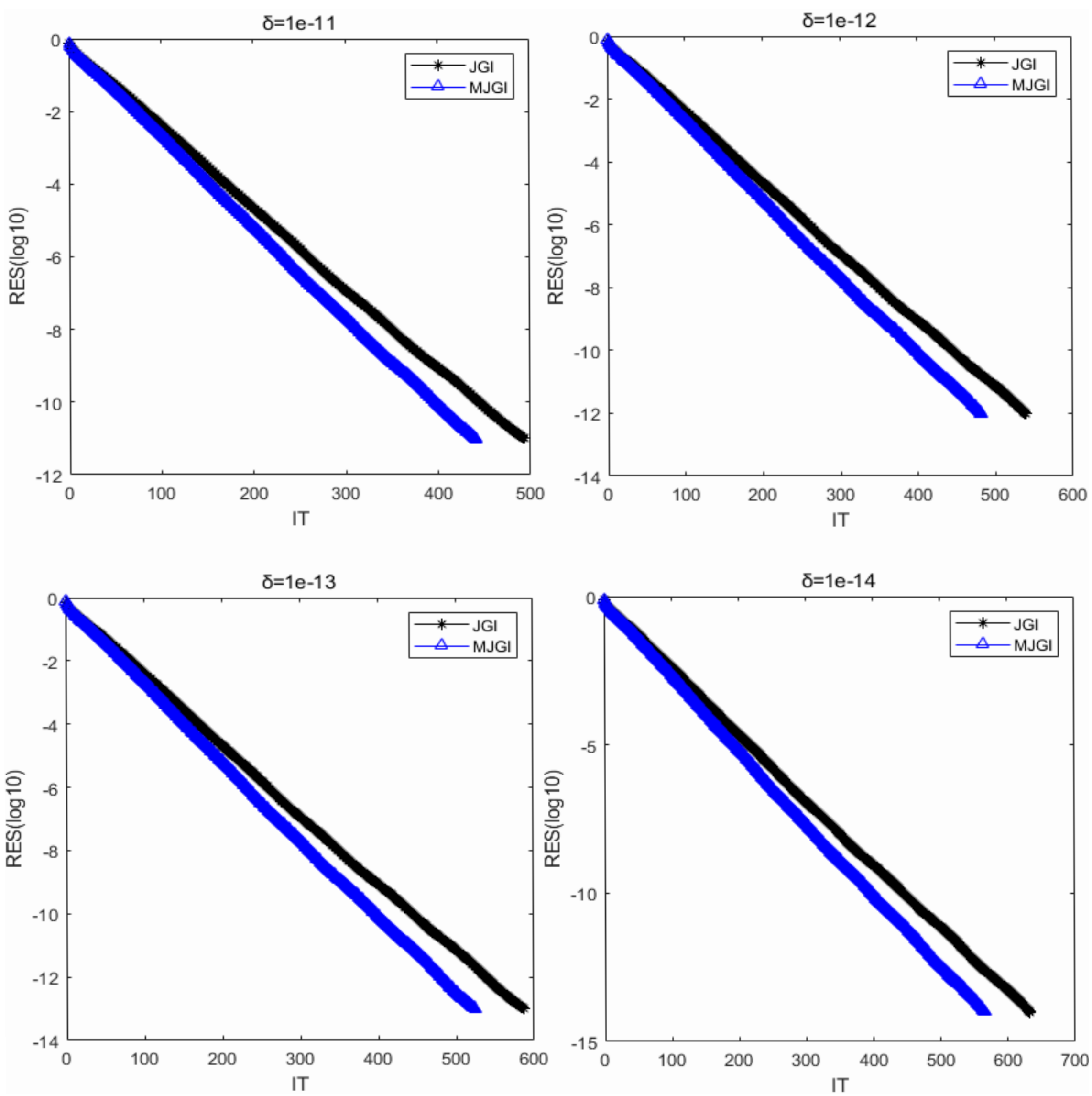

Figure 5: Comparisons for the convergence curves of the JGI and MJGI algorithms for Example 6.3.

Table 3: IT, CPU and RES of three tested algorithms for Example 6.3 with five values of δ

Algorithm		δ				
		10^{-11}	10^{-12}	10^{-13}	10^{-14}	10^{-15}
GI	IT	1980	2191	2402	2615	Fail
$\mu = 2.43e-3$	CPU	3.9936	4.7360	4.9651	5.1907	–
	RES	9.9423e-12	9.8927e-13	9.8948e-14	9.9586e-15	–
JGI	IT	494	540	588	633	701
$\mu = 3.81e-3$	CPU	1.1895	1.2923	1.3002	1.4048	1.5182
	RES	9.9490e-12	9.5835e-13	9.6901e-14	9.9507e-15	9.9620e-16
MJGI	IT	442	482	525	566	620
$\mu = 4.22e-3$	CPU	1.0044	1.1603	1.1428	1.2306	1.3378
	RES	9.5747e-12	9.9003e-13	9.7730e-14	9.7858e-15	9.9989e-16

As for Examples 6.1–6.2, the parameters of the GI, JGI and MJGI algorithms are adopted to be the experimentally found optimal ones which minimize their IT. In Table 3, we compare the numerical results of the three tested algorithms for Example 6.3 with five different values of δ. From the results in Table 3, it is observed that all tested algorithms are convergent for all cases except that the GI algorithm fails to converge for $\delta = 10^{-15}$. And when the calculation error decreases, the IT and CPU time of the tested algorithms increase. In addition, the proposed MJGI algorithm has better numerical performance than the other ones due to the fact that the IT and CPU time of the former one are always less than those of the GI and JGI ones. And the advantage of the MJGI algorithm becomes more pronounced as the value of δ becomes smaller, because the numerical performance gap between the MJGI algorithm and GI, JGI algorithms is increasingly larger with the decreases of δ. Also, the IT and CPU time of the MJGI algorithm are nearly one fourth of those of the GI one. Last but not least, the changing scope of the IT for the proposed MJGI algorithm is bigger than those of the GI and JGI ones, which indicates that the stability of the MJGI algorithm is the highest among all tested algorithms. All in all, the technique utilized in the MJGI algorithm can improve the convergence speeds of the GI and JGI ones, and the MJGI algorithm outperforms the other ones from the point of view of computing efficiency.

To better validate the advantage of the MJGI algorithm, we present the graphs of RES(log10) against IT of the three tested algorithms in Figure 4 for $\delta = 10^{-11}$ and $\delta = 10^{-12}$. As shown in Figure 4, all algorithms are convergent while the MJGI algorithm has faster convergence rate than the GI and JGI ones as the IT of the MJGI algorithm is the least among the tested algorithms. This is consistent with the results in Table 3. To further verify the superiority of the proposed MJGI algorithm to the JGI one, we plot the IT curves of the MJGI and JGI algorithms with respect to four different values of δ in Figure 5. From Figure 5, we observe that the MJGI algorithm performs better than the JGI one in view of IT, and the advantage of the MJGI algorithm is more obvious when the value of δ becomes smaller.

7. Conclusions

In this work, we first correct some errors in the convergence proofs of the JGI and the AJGI algorithms in [39], and establish new and correct convergence conditions of these two algorithms. Then by applying a new update technique to the JGI algorithm, we develop a new algorithm called the EJGI algorithm for the DTPS matrix equations, which is different from the AJGI one and has advantage over the AJGI one. In addition, we combine the idea of the Jacobi method with the update strategy, and construct the MJGI algorithm for the GDTPS matrix equations, which requires less computations than the GI one. Besides, compared with the JGI algorithm, the MJGI algorithm can use the latest results to compute the next results, which leads to a faster convergence rate. In addition, by making use of the properties of the vector stretching operator, matrix norm and Kronecker product of two matrices, we establish the convergence theorems of the the EJGI and the MJGI algorithms. Finally, numerical experiments are performed to show the effectiveness and the superiorities of the new algorithms.

However, the convergent intervals of the parameters μ, ω in the EJGI and the MJGI algorithms and their optimal values have not been derived at present. We will investigate these problems in our future work, which are meaningful to implement the EJGI and the MJGI algorithms effectively in practical applications.

Competing interests

The authors declare that they have no competing interests.

Acknowledgments

We would like to express our sincere thanks to editor and anonymous reviewer for their valuable suggestions and constructive comments which greatly improved the presentation of this paper.

References

[1] S. Bittanti, P. Bolzern. Stabilizability and detectability of linear periodic systems. *Systems and control letters*, 1985, 6(2): 141-145.

[2] S. Bittanti, P. Bolzern, P. Colaneri. Stability analysis of linear periodic systems via the Lyapunov equation. *IFAC Proceedings Volumes*, 1984, 17(2): 213-216.

[3] M. Hajarian. Extending LSQR methods to solve the generalized Sylvester - transpose and periodic Sylvester matrix equations. *Mathematical Methods in the Applied Sciences*, 2014, 37(13): 2017-2028.

[4] Z.-H. He, Q.-W. Wang. A system of periodic discrete-time coupled Sylvester quaternion matrix equations. *World Scientific Publishing Company*, 2017, 24(01): 169-180.

[5] V. Hernanadez, A. M. Urbano. Pole-placement problem for discrete-time linear periodic systems. *International Journal of Control*, 1989, 50(1): 361-371.

[6] M. Kono. Eigenvalue assignment in linear periodic discrete-time systems. *International Journal of Control*, 1980, 32(1): 149-158.

[7] S. Longhi, R. Zulli. A note on robust pole assignment for periodic systems *IEEE Transactions on Automatic Control*, 1996, 41(10): 1493-1497.

[8] L.-L. Lv, J.-B. Chen, L. Zhang, F.-R. Zhang. Gradient-based neural networks for solving periodic Sylvester matrix equations. *Journal of the Franklin Institute*, 2022, 359(18): 10849-10866.

[9] A. Varga. Periodic Lyapunov equations: some applications and new algorithms. *International Journal of Control*, 1997, 67(1): 69-88.

[10] M. Hajarian. A finite iterative method for solving the general coupled discrete-time periodic matrix equations. *Circuits, Systems, and Signal Processing*, 2015, 34: 105-125.

[11] M. Hajarian. Matrix form of biconjugate residual algorithm to solve the discrete-time periodic Sylvester matrix equations. *Asian Journal of Control*, 2018, 20(1): 49-56.

[12] L.-L. Lv, Z. Zhang, L. Zhang, X.-X. Liu. Gradient based approach for generalized discrete-time periodic coupled Sylvester matrix equations. *Journal of the Franklin Institute*, 2018, 355(15): 7691-7705.

[13] M. Hajarian. Three types of biconjugate residual method for general periodic matrix equations over generalized bisymmetric periodic matrices. *Transactions of the Institute of Measurement and Control*, 2019, 41(10): 2708-2725.

[14] M. Hajarian. Reflexive periodic solutions of general periodic matrix equations. *Mathematical Methods in the Applied Sciences*, 2019, 42(10): 3527-3548.

[15] M. K. Zak, F. Toutounian. Nested splitting conjugate gradient method for matrix equation $AXB = C$ and preconditioning. *Computers & Mathematics with Applications*, 2013, 66(3): 269-278.

[16] H. M. Kim, Y. Kim, J. Meng. NEWTON SCHULZ METHOD FOR SOLVING NONLINEAR MATRIX EQUATION $X^p + A^*XA = Q$. *Journal of the Korean Mathematical Society*, 2018, 55(6): 1529-1540.

[17] L.-L. Lv, S.-Y. Tang, L. Zhang. Parametric solutions to generalized periodic Sylvester bimatrix equations. *Journal of the Franklin Institute*, 2020, 357: 3601-3621.

[18] L. L. Lv, W. Q. Lai, X. Y. Liu. A parametric poles assignment algorithm for high-order linear discrete periodic systems. *Journal of the Franklin Institute*, Filomat, 2023, 360: 11855-11870.

[19] A.-G. Wu, X. L. Zeng, G.-R. Duan, W.-J. Wu. Iterative solutions to the extended Sylvester-conjugate matrix equations. *Applied Mathematics and Computation*, 2010, 217(1): 130-142.

[20] H.-M. Zhang, H.-C. Yin. New proof of the gradient-based iterative algorithm for a complex conjugate and transpose matrix equation. *Journal of the Franklin Institute*, 2017, 354(16): 7585-7603.

[21] C.-X. Li, S.-L. Wu. The SHSS preconditioner for saddle point problems. *Journal of Applied Analysis and Computation*, 2023, 13(6): 3221-3230.

[22] T.-X. Yan, C.-F. Ma. An iterative algorithm for solving a class of generalized coupled Sylvester-transpose matrix equations over bisymmetric or skew-anti-symmetric matrices. *Journal of Applied Analysis and Computation*, 2020, 10(4): 1282-1310.

[23] Y.-N. Wu, M.-L. Zeng. On ADMM-based methods for solving the nearness symmetric solution of the system of matrix equations. *Journal of Applied Analysis and Computation*, 2021, 11(1): 227-241.

[24] W.-L. Wang, C.-Q. Song, W.-L. Wang. A new BCR method for coupled operator equations with submatrix constraint. *Journal of Applied Analysis and Computation*, 2024, 14(4): 2002-2036.

[25] A. Shirilord, M. Dehghan. Iterative algorithm for a generalized matrix equation with momentum acceleration approach and its convergence analysis. *Journal of the Franklin Institute*, 2024, 361: 107021.

[26] A. Shirilord, M. Dehghan. Solving a system of complex matrix equations using a gradient-based method and its application in image restoration. *Numerical Algorithms*, 2024, DOI: 10.1007/s11075-024-01856-2.

[27] A. Shirilord, M. Dehghan. Iterative method for constrained systems of conjugate transpose matrix equations. *Applied Numerical Mathematics*, 2024, 198: 474-507.

[28] A. Shirilord, M. Dehghan. Stationary Landweber method with momentum acceleration for solving least squares problems. *Applied Mathematics Letters*, 2024, 157: 109174.

[29] Z.-G. Huang, J.-J. Cui. Modified and accelerated relaxed gradient-based iterative algorithms for the complex conjugate and transpose matrix equations. *Numerical Algorithms*, 2024, 97: 1955-2009.

[30] M. Hajarian. Convergence properties of BCR method for generalized Sylvester matrix equation over generalized reflexive and anti-reflexive matrices. *Linear and Multilinear Algebra*, 2018, 66(10): 1975-1990.

[31] M. Hajarian. Computing symmetric solutions of general Sylvester matrix equations via Lanczos version of biconjugate residual algorithm. *Computers & Mathematics with Applications*, 2018, 76(4): 686-700.

[32] H.-M. Zhang. Gradient-based iterative algorithm for the extended coupled Sylvester matrix equations. *2017 29th Chinese Control And Decision Conference (CCDC). IEEE*, 2017: 1562-1567.

[33] Y.-J. Xie, C.-F. Ma. The accelerated gradient based iterative algorithm for solving a class of generalized Sylvester-transpose matrix equation. *Applied Mathematics and Computation*, 2016, 273: 1257-1269.

[34] J.-J. Zhang. A note on the iterative solutions of general coupled matrix equation. *Applied Mathematics and Computation*, 2011, 217(22): 9380-9386.

[35] H.-M. Zhang. Reduced-rank gradient-based algorithms for generalized coupled Sylvester matrix equations and its applications. *Computers & Mathematics with Applications*, 2015, 70(8): 2049-2062.

[36] B.-H. Huang, C.-F. Ma. The relaxed gradient-based iterative algorithms for a class of generalized coupled Sylvester-conjugate matrix equations. *Journal of the Franklin Institute*, 2018, 355(6): 3168-3195.

[37] W.-L. Wang, C.-Q. Song. A novel iterative method for solving the coupled Sylvester-conjugate matrix equations and its application in antilinear system. *Journal of Applied Analysis and Computation*, 2023, 13(1): 1-26.

[38] W.-L. Wang, G.-R. Qu, C.-Q. Song. Cyclic gradient based iterative algorithm for a class of generalized coupled Sylvester-conjugate matrix equations. *Journal of the Franklin Institute*, 2023, 360: 7206-7229.

[39] W.-L. Wang, C.-Q. Song. Iterative algorithms for discrete-time periodic Sylvester matrix equations and its application in antilinear periodic system. *Applied Numerical Mathematics*, 2021, 168: 251-273.

[40] M. Hajarian. Solving the general Sylvester discrete-time periodic matrix equations via the gradient based iterative method. *Applied Mathematics Letters*, 2016, 52: 87-95.

[41] S.-H. Li, C.-F. Ma. Factor gradient iterative algorithm for solving a class of discrete periodic Sylvester matrix equations. *Journal of the Franklin Institute*, 2022, 359(17): 9952-9970.

[42] A.-G. Wu, Y. Zhang, Y.-Y. Qian. Complex Conjugate Matrix Equations. *Beijing: Science Press*, 2017.

[43] W.-L. Wang, C.-Q. Song, S.-P. Ji. Iterative solution to a class of complex matrix equations and its application in time-varying linear system. *Journal of Applied Mathematics and Computing*, 2021, 67: 317-341.

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