Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

Matrix iteration algorithms for solving the generalized Lyapunov matrix equation

Juan Zhang[1*], Huihui Kang[1] and Shifeng Li[1]

*Correspondence: zhangjuan@xtu.edu.cn
[1]Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Department of Mathematics and Computational Science, Xiangtan University, Xiangtan, Hunan, 411105, P.R. China

Abstract

In this paper, we first recall some well-known results on the solvability of the generalized Lyapunov equation and rewrite this equation into the generalized Stein equation by using Cayley transformation. Then we introduce the matrix versions of biconjugate residual (BICR), biconjugate gradients stabilized (Bi-CGSTAB), and conjugate residual squared (CRS) algorithms. This study's primary motivation is to avoid the increase of computational complexity by using the Kronecker product and vectorization operation. Finally, we offer several numerical examples to show the effectiveness of the derived algorithms.

Keywords: Generalized Lyapunov equation; Cayley transformation; BICR algorithm; Bi-CGSTAB algorithm; CRS algorithm

1 Introduction

In this paper, we consider the generalized Lyapunov equation as follows:

$$AX + XA^T + \sum_{j=1}^{m} N_j X N_j^T + C = 0, \tag{1}$$

where $A, N_j \in \mathbb{R}^{n\times n}$ $(j = 1, 2, \ldots, m)$, $m \ll n$ and $C \in \mathbb{R}^{n\times n}$ is symmetric, $X \in \mathbb{R}^{n\times n}$ is the symmetric solution of (1).

The generalized Lyapunov equation (1) is related to several linear matrix equations displayed in Table 1. A large and growing amount of literature has considered the solution for these equations; see [1, 2] and the references therein for an overview of developments and methods.

The generalized Lyapunov equation (1) often appears in the context of bilinear systems [3, 4], stability analysis of linear stochastic systems [5, 6], special linear stochastic differen-

Table 1 Several linear matrix equations

$AX + XA^T + C = 0$	continuous-time Lyapunov matrix equation
$AXD^T + DXA^T + C = 0$	generalized continuous-time Lyapunov matrix equation
$A^TXA - D^TXD + C = 0$	generalized discrete-time Lyapunov matrix equation
$AX - XD + C = 0$	continuous-time Sylvester matrix equation
$AXD^T - X + C = 0$	discrete-time Sylvester matrix equation

tial equations [7] and other areas. For example, we discuss the origin of Eq. (1) in bilinear systems. The bilinear system is an interesting subclass of nonlinear control systems that naturally occurs in some boundary control dynamics [6]. The bilinear control system has been studied by scholars for many years and has the following state space representation:

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + \sum_{j=1}^{m} N_j x(t) u_j(t) + Bu(t), \\ y(t) = Cx(t), \qquad x(0) = x_0, \end{cases} \tag{2}$$

where t is the time variable, $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^n$ are the stable, input and output vectors, respectively, $u_j(t)$ is the jth component of $u(t)$. $B \in \mathbb{R}^{n\times m}$, and A, N_j, C are defined in (1).

For the bilinear control system (2), define

$$\begin{aligned} &P_1 = e^{At_1}B, \\ &P_i(t_1, \ldots, t_i) = e^{At_i}[N_1 P_{i-1}, \ldots, N_m P_{i-1}], \quad i = 2, 3, \ldots. \end{aligned}$$

Using the concept of reachability in [3, 8], the reachability corresponding to (2) is

$$P = \sum_{i=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} P_i P_i^T \, dt_1 \cdots dt_i,$$

where P is the solution of (1).

Moreover, the generalized Lyapunov equation (1) has wide applications in PDEs. Consider the heat equation subjected to mixed boundary conditions [9]

$$\begin{aligned} &x_t = \Delta x \quad \text{in } \Omega, \\ &n \cdot \nabla x = u(x-1) \quad \text{on } \Gamma_1, \\ &x = 0 \quad \text{on } \Gamma_2, \Gamma_3, \Gamma_4, \end{aligned} \tag{3}$$

where Γ_1, Γ_2, Γ_3 and Γ_4 are the boundaries of Ω. For example, for a simple 2×2 mesh, the state vector $x = [x_{11}, x_{21}, x_{12}, x_{22}]^T$ contains the temperatures at the inner points and the Laplacian is approximated via

$$\Delta x_{ij} \approx -\frac{1}{h^2}(4x_{ij} - x_{i+1,j} - x_{i,j+1} - x_{i-1,j} - x_{i,j-1}),$$

with meshsize $h = 1/3$. If the Robin condition is imposed on the whole boundary, then we have

$$\begin{aligned} &x_{10} \approx x_{11} - hu(x_{11} - 1), \qquad x_{20} \approx x_{21} - hu(x_{21} - 1), \\ &x_{01} \approx x_{11} - hu(x_{11} - 1), \qquad \ldots \end{aligned}$$

Altogether this leads to the bilinear system

$$\begin{aligned}\dot{x} &= \frac{1}{h^2}\begin{pmatrix}-2 & 1 & 1 & 0\\ 1 & -2 & 0 & 1\\ 1 & 0 & -2 & 1\\ 0 & 1 & 1 & -2\end{pmatrix}x\\ &\quad + \frac{1}{h}\left(\begin{pmatrix}x_{11}-1\\ x_{21}-1\\ 0\\ 0\end{pmatrix}u_1 + \begin{pmatrix}x_{11}-1\\ 0\\ x_{12}-1\\ 0\end{pmatrix}u_2 + \begin{pmatrix}0\\ 0\\ x_{21}-1\\ x_{22}-1\end{pmatrix}u_3 + \begin{pmatrix}0\\ x_{21}-1\\ 0\\ x_{22}-1\end{pmatrix}u_4\right)\\ &= \frac{1}{h^2}Ax + \frac{1}{h}\big((A_1x+b_1)u_1 + (A_2x+b_2)u_2 + (A_3x+b_3)u_3 + (A_4x+b_4)u_4\big),\end{aligned} \tag{4}$$

where $E_j = e_je_j^T$ with canonical unit vector $e_j \in \mathbb{R}^2$, and

$$A = \begin{pmatrix}-2 & 1 & 1 & 0\\ 1 & -2 & 0 & 1\\ 1 & 0 & -2 & 1\\ 0 & 1 & 1 & -2\end{pmatrix}, \qquad A_1 = (E_1 \otimes I),$$

$$A_2 = (I \otimes E_1), \qquad A_3 = (E_2 \otimes I), \qquad A_4 = (I \otimes E_2),$$

$$b_1 = E_1 \otimes e, \qquad b_1 = E_1 \otimes e, \qquad b_1 = E_1 \otimes e, \qquad b_1 = E_1 \otimes e, \quad e = [1,1].$$

Thus, the optimal control problem of (4) reduces to the bilinear control system (2) and we ultimately need solve the generalized Lyapunov equation:

$$AX + XA + \sum_{j=1}^{4} A_jXA_j = -BB^T.$$

Therefore, considering the important applications of the generalized Lyapunov equation (1), many researchers pay much attention to study the solution for this equation in recent years. Damm showed the direct method to solve the generalized Lyapunov equation [9]. Fan et al. transformed this equation into the generalized Stein equation by generalized Cayley transformation and solved it using GSM [10]. Dai et al. proposed the HSS algorithm to solve the generalized Lyapunov equation. Li et al. proposed the PHSS iterative method for solving this equation when A is asymmetric positive definite [11]. Based on the recent results, we mainly discuss the matrix iteration algorithms for the generalized Lyapunov equation (1).

The rest of the paper is organized as follows. In Sect. 2, we recall some known results on the generalized Lyapunov equation's solvability and rewrite this equation into the generalized Stein equation by using Cayley transformation. In Sect. 3, we present the matrix versions and variant forms of the BICR, Bi-CGSTAB, and CRS algorithms. In Sect. 4, we offer several numerical examples to test the effectiveness of the derived algorithms. In Sect. 5, we draw some concluding remarks.

Throughout this paper, we shall adopt the following notations. $\mathbb{R}^{m\times n}$ and $\mathbb{Z}^+$ stand for the set of all $m \times n$ real matrices and positive integers. For $A = (a_{ij}) = (a_1, a_2, \dots, a_n) \in$

$\mathbb{R}^{m\times n}$, the symbol $\operatorname{vec}(A)$ is a vector defined by $\operatorname{vec}(A) = (a_1^T, a_2^T, \ldots, a_n^T)^T$. A^T and $\|A\|$ represent the transpose and 2-norm of matrix A, respectively. The symbol $A \geq 0$ means that A is symmetric positive semi-definite. For $B \in \mathbb{R}^{m\times n}$, the Kronecker product and inner product of A and B are defined by $A \otimes B = (a_{ij}B)$ and $\langle A, B\rangle = \operatorname{tr}(B^T A)$. The open right-half and left-half planes are denoted by $\mathbb{C}_+$ and $\mathbb{C}_-$, respectively.

2 Solvability and Cayley transformation

2.1 Solvability of the generalized Lyapunov equation

This section introduces the solvability for the generalized Lyapunov equation (1).

Denote $\sigma(T) \in \mathbb{C}$ by the spectrum of a linear operator T and $\rho(T) = \max\{|\lambda||\lambda \in \sigma(T)\}$ by the spectral radius. Define the linear matrix operators $\mathcal{L}_A$ and $\Pi : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ by

$$\mathcal{L}_A = A^T X + XA, \qquad \Pi(X) \mapsto \sum_{j=1}^{m} N_j X N_j^T. \tag{5}$$

Obviously, $\Pi(X) \geq 0$ when $X \geq 0$.

Therefore, using Theorem 3.9 in [6], we immediately get the generalized Lyapunov equation's stability result.

Theorem 2.1 *Let $A \in \mathbb{R}^{n\times n}$ and Π be positive. The following conclusions are equivalent*:

(a) *For all $Y > 0$, $\exists X > 0$ such that $\mathcal{L}_A(X) + \Pi(X) = -Y$;*

(b) *$\exists Y, X > 0$ such that $\mathcal{L}_A(X) + \Pi(X) = -Y$;*

(c) *$\exists Y \geq 0$ with (A, Y) controllable, $\exists X > 0$ such that $\mathcal{L}(X) + \Pi(X) = -Y$;*

(d) *$\sigma(\mathcal{L}_A(X) + \Pi(X)) \subset \mathbb{C}_-$;*

(e) *$\sigma(\mathcal{L}_A(X)) \subset \mathbb{C}_-$ and $\rho(\mathcal{L}_A^{-1}(X)\Pi(X)) < 1$,*

where the linear matrix operators $\mathcal{L}_A$ and Π are defined by (5).

Remark 2.1 For the generalized Lyapunov equation (1), we often choose $C = BB^T$, i.e., C is symmetric positive semi-definite. Using Theorem 2.1, Eq. (1) has a positive definite solution X if A is stable, (A, B) is controllable, and the norm of the N_j is sufficiently small.

2.2 Cayley transformation for (1)

In this section, we introduce Cayley transformation for the generalized Lyapunov equation.

It is well known that Cayley transformation is a link between the classical Lyapunov and Stein equations. Fan et al. have shown that the stability of the Lyapunov and Stein equations is different. Naturally, we wonder if the stability is different and the counterparty method has other effects. It is verified in Sect. 4 that our iteration methods are more efficient after applying Cayley transformation to the generalized Lyapunov equation. We first recall the definition of Cayley transformation.

Definition 2.1 (Cayley transformation) Let $M \in \mathbb{R}^{n\times n}$ be a skew-symmetric matrix. Then $\mathcal{N} = (I + M)^{-1}(I - M)$ is called Cayley transformation of M.

Next, we show that the generalized Lyapunov equation can be changed to the generalized Setin equation after Cayley transformation.

Theorem 2.2 *For the generalized Lyapunov equation* (1), *take the positive parameter* γ *such that the matrices* $(\gamma I + A)$ *and* $(\gamma I + A^T)$ *are both nonsingular. Then* (1) *is equivalent to the generalized Stein equation*

$$X - \hat{A}X\hat{A}^T + 2\gamma \sum_{j=1}^{m} \hat{N}_j X \hat{N}_j^T + 2\gamma \hat{C} = 0, \tag{6}$$

where

$$\hat{A} = (\gamma I + A)^{-1}(\gamma I - A),$$
$$\hat{N}_j = (\gamma I + A)^{-1} N_j,$$
$$\hat{C} = (\gamma I + A)^{-1} C (\gamma I + A)^{-T}.$$

Proof Introducing the positive parameter γ to (1), we get

$$(\gamma I + A)X(\gamma I + A^T) - (\gamma I - A)X(\gamma I - A^T) + 2\gamma\left(\sum_{j=1}^{m} N_j X N_j^T\right) + 2\gamma C = 0. \tag{7}$$

Since $(\gamma I + A)$ and $(\gamma I + A^T)$ are both nonsingular, premultiplying $(\gamma I + A)^{-1}$ and postmultiplying $(\gamma I + A^T)^{-1}$ on both sides to (7) yield (6). Thus we complete the proof of Theorem 2.2. □

Remark 2.2 Viewing Theorem 2.2, it involves a positive parameter γ. We offer a practical way to choose γ. Set

$$\gamma = \max_{1\le i\le n} a_{ii},$$

then $(\gamma I + A)$ and $(\gamma I + A^T)$ are both nonsingular. Thus the condition of Theorem 2.2 is satisfied. Appropriate adjustments can be made according to different situations.

Remark 2.3 Next, we show the relationship between the generalized Lyapunov equation (1) and the generalized Stein equation (6) by using the preconditioner method of linear systems.

By utilizing the operator vec, the generalized Lyapunov equation can be rewritten as

$$\mathcal{A}_1 \mathcal{X} = \left(I \otimes A + A \otimes I + \sum_{j=1}^{m} N_j \otimes N_j\right) \operatorname{vec}(X) = -\operatorname{vec}(C).$$

The generalized Stein equation can be rewritten as

$$\mathcal{A}_2 \mathcal{X} = \left(I \otimes I + \hat{A} \otimes \hat{A} + \sum_{j=1}^{m} \hat{N}_j \otimes \hat{N}_j\right) \operatorname{vec}(X) = -2\gamma \operatorname{vec}(\hat{C}).$$

Hence, it is not difficult to derive the following relation between $\mathcal{A}_1$ and $\mathcal{A}_2$:

$$\mathcal{A}_2 = 2\gamma\left[(\gamma I + A) \otimes (\gamma I + A)\right]^{-1} \mathcal{A}_1,$$

where

$$P_{pre} = \frac{1}{2\gamma}\big[(\gamma I + A) \otimes (\gamma I + A)\big]$$

is the preconditioning matrix and the corresponding generalized Stein equation is the preconditioning system.

By Remark 2.2, the operator $\mathcal{P}$ can be defined as

$$\mathcal{P}(X) = X - \hat{A}X\hat{A}^T + 2\gamma \sum_{j=1}^{m} \hat{N}_j X \hat{N}_j^T.$$

In Sect. 3, we apply this operator to derive the variant forms of the BICR, Bi-CGSTAB, and CRS algorithms, respectively. The iteration methods are efficient. Numerical examples address this point in Sect. 4.

3 Iteration algorithms

This section presents the matrix versions and variant forms of the BICR, Bi-CGSTAB, and CRS algorithms in three subsections, respectively.

3.1 BICR algorithm

The BiCR method [12] has been proposed as a generalization of the conjugate residual (CR) [13] method for nonsymmetric matrices. Recently, Abe et al. designed BiCR for symmetric complex matrices (SCBiCR) and analyzed the factor in the loss of convergence speed [14]. It is easy to see that the BICR algorithm cannot be directly used to solve the generalized Lyapunov equation. Naturally, one can convert this matrix equation into the linear system through Kronecker product and vectorization operators. However, this makes the computational cost especially expensive. When the matrix order becomes larger, as the computer memory is limited, it is hard to implement in practice.

Therefore, we need to modify the BICR algorithm and ensure that the calculation cost is relatively cheap. In this subsection, we propose the matrix version of the BICR algorithm (Algorithm 1). Then we show the variant version of the BICR algorithm (Algorithm 2).

Using the iteration schemes of Algorithm 1 and Algorithm 2, we can directly solve the generalized Lyapunov equation. Further, we show the bi-orthogonal properties and convergent analysis of Algorithm 1 by Theorem 3.1 and Theorem 3.2.

Theorem 3.1 *For Algorithm 1, we assume that there exists a positive integer number such that $W(k) \neq 0$ and $R(k) \neq 0$ for all $k = 1, 2, \dots, r$. Then we get*

$$\operatorname{tr}\big(R(v)^T W(u)\big) = 0, \quad \textit{for } u, v = 1, 2, \dots, r, u < v, \tag{8}$$

$$\operatorname{tr}\big(S(v)^T Z(u)\big) = 0, \quad \textit{for } u, v = 1, 2, \dots, r, u < v, \tag{9}$$

$$\operatorname{tr}\big(Z(v)^T Z(u)\big) = 0, \quad \textit{for } u, v = 1, 2, \dots, r, u \neq v, \tag{10}$$

$$\operatorname{tr}\big(W(v)^T W(u)\big) = 0, \quad \textit{for } u, v = 1, 2, \dots, ru \neq v. \tag{11}$$

For the proof of Theorem 3.1, refer to the Appendix.

Algorithm 1: Matrix form of the BICR algorithm

```
step 1: Choose initial matrices $X(1) \in \mathbb{R}^{n\times n}$, $S(1) \in \mathbb{R}^{n\times n}$;
step 2: $X$ with $\|R(k)\| \le tol$;
step 3: Compute $R(1) = AX(1) + X(1)A^T + \sum_{j=1}^m N_jX(1)N_j^T + C$, $U(1) = S(1)$,
$V(1) = R(1)$, $W(1) = AU(1) + U(1)A^T + \sum_{j=1}^m N_jU(1)N_j^T$,
$Z(1) = A^TV(1) + V(1)A + \sum_{j=1}^m N_j^TV(1)N_j$, $k = 1$;
step 4: While $\|R(k)\| >$ tol do:
    $\alpha(k) = \frac{\mathrm{tr}(W(k)^TR(k))}{\mathrm{tr}(W(k)^TW(k))}$,   $X(k+1) = X(k) - \alpha(k)U(k)$,
    $R(k+1) = R(k) - \alpha(k)W(k)$,
    $\beta(k) = \frac{\mathrm{tr}(Z(k)^TS(k))}{\mathrm{tr}(Z(k)^TZ(k))}$,   $S(k+1) = S(k) - \beta(k)Z(k)$,
    $\gamma(k) = \frac{\mathrm{tr}(W(k)^T(AS(k+1) + S(k+1)A^T + \sum_{j=1}^m N_jS(k+1)N_j^T))}{\mathrm{tr}(W(k)^TW(k))}$,
    $\eta(k) = \frac{\mathrm{tr}(Z(k)^T(A^TR(k+1) + R(k+1)A + \sum_{j=1}^m N_j^TR(k+1)N_j))}{\mathrm{tr}(Z(k)^TZ(k))}$,
    $U(k+1) = S(k+1) - \gamma(k)U(k)$,   $V(k+1) = R(k+1) - \eta(k)V(k)$,
    $W(k+1) = AS(k+1) + S(k+1)A^T + \sum_{j=1}^m N_jS(k+1)N_j^T - \gamma(k)W(k)$,
    $Z(k+1) = A^TR(k+1) + R(k+1)A + \sum_{j=1}^m N_j^TR(k+1)N_j - \eta(k)Z(k)$.
```

Theorem 3.2 *For Algorithm 1, the relative residual error has the following property*:

$$\|R(k+1)\|^2 \le \|R(k)\|^2.$$

Proof Using Theorem 3.1, we have

$$\begin{aligned}\|R(k+1)\|^2 &= \mathrm{tr}\big(R(k+1)^TR(k+1)\big)\\ &= \mathrm{tr}\big(\big(R(k) - \alpha(k)W(k)\big)^T\big(R(k) - \alpha(k)W(k)\big)\big)\\ &= \|R(k)\|^2 + \alpha(k)^2\|W(k)\|^2 - 2\alpha(k)\,\mathrm{tr}\big(W(k)^TR(k)\big)\\ &= \|R(k)\|^2 - \alpha(k)\,\mathrm{tr}\big(W(k)^TR(k)\big)\\ &= \|R(k)\|^2 - \frac{\mathrm{tr}(W(k)^TR(k))^2}{\mathrm{tr}(W(k)^TW(k))}\\ &\le \|R(k)\|^2.\end{aligned}$$

Hence, the proof of Theorem 3.2 is completed. □

```
Algorithm 2: The variant form of the BICR algorithm
  Input: Choose initial matrices $X(1) \in \mathbb{R}^{n\times n}$, $S(1) \in \mathbb{R}^{n\times n}$;
  Output: $X$ with $\|R(k)\| \leq tol$;
  step 1: Compute $R(1) = \mathcal{P}(X(1)) - 2\gamma\hat{C}$, $U(1) = S(1)$, $V(1) = R(1)$,
  $W(1) = AU(1) + U(1)A^T + \sum_{j=1}^{m} N_j U(1) N_j^T$, $Z(1) = \mathcal{P}^T(V(1))$, $k = 1$;
  step 2: While $\|R(k)\| > \mathrm{tol}$ do:
      $\alpha(k) = \frac{\mathrm{tr}(W(k)^T R(k))}{\mathrm{tr}(W(k)^T W(k))}$,    $X(k+1) = X(k) - \alpha(k)U(k)$,
      $R(k+1) = R(k) - \alpha(k)W(k)$,
      $\beta(k) = \frac{\mathrm{tr}(Z(k)^T S(k))}{\mathrm{tr}(Z(k)^T Z(k))}$,    $S(k+1) = S(k) - \beta(k)Z(k)$,
      $\gamma(k) = \frac{\mathrm{tr}(W(k)^T(\mathcal{P}(S(k+1))))}{\mathrm{tr}(W(k)^T W(k))}$,    $U(k+1) = S(k+1) - \gamma(k)U(k)$,
      $\eta(k) = \frac{\mathrm{tr}(Z(k)^T(\mathcal{P}^\top(R(k+1))))}{\mathrm{tr}(Z(k)^T Z(k))}$,    $V(k+1) = R(k+1) - \eta(k)V(k)$,
      $W(k+1) = \mathcal{P}(S(k+1)) - \gamma(k)W(k)$,    $Z(k+1) = \mathcal{P}^\top(R(k+1)) - \eta(k)Z(k)$.
```

Remark 3.1 In terms of Theorem 3.2, the property $\|R(k+1)\| \leq \|R(k)\|$ ensures that Algorithm 1 possesses fast and smooth convergence.

3.2 Bi-CGSTAB algorithm

Sonneveld [15] has shown a variant of BiCG, referred to the conjugate gradient squared (CGS). Van der Vorst [16] has derived one of the most successful variants of BiCG, known as the Bi-CGSTAB method. The Bi-CGSTAB algorithm is an effective algorithm for solving large sparse linear systems [16, 17]. Chen et al. [18] proposed a flexible version of the BiCGStab algorithm for solving the linear system. It is easy to see that the Bi-CGSTAB algorithm cannot be directly used to solve the generalized Lyapunov equation. Similarly, we need to modify the Bi-CGSTAB algorithm to the matrix version. The matrix version of the Bi-CGSTAB algorithm is summarized in Algorithm 3. The variant form of the BICR algorithm is shown in Algorithm 4.

Viewing the iteration schemes, we can be seen that Algorithm 3 is a simple matrix form of the Bi-CGSTAB algorithm. Hence, Algorithm 3 has the same properties as the Bi-CGSTAB algorithm. Algorithm 4 is an improved version of the Bi-CGSTAB algorithm, which has high computing efficiency. This point has been addressed by numerical examples in Sect. 4.

3.3 CRS algorithm

Zhang et al. proposed the conjugate residual squared (CRS) method in [19, 20] to solve the linear system. The CRS algorithm is mainly aimed to avoid using the transpose of A in the BiCR algorithm and get faster convergence for the same computational cost [19]. Recently, Ma et al. [21] used the matrix CRS iteration method to solve a class of coupled Sylvester-transpose matrix equations. Later, they extended two mathematical equivalent

```
Algorithm 3: Matrix form of the Bi-CGSTAB algorithm
  step 1: Choose initial matrix $X(1)$;
  step 2: Compute $R(1) = -C - AX(1) - X(1)A^T - \sum_{j=1}^m N_jX(1)N_j^T$, pick arbitrary matrix $\tilde{R}(1)$ (e.g. $\tilde{R}(1) = R(1)$);
  step 3: Compute $V(1) = P(1) = 0$, $\rho(1) = \alpha(1) = \omega(1) = 1$;
  step 4: While $\|R(k)\| > \mathrm{tol}$ do:
      $\rho(k+1) = \langle R(k), \tilde{R}(1)\rangle, \quad \beta(k+1) = \left(\frac{\rho(k+1)}{\rho(k)}\right)\left(\frac{\alpha(k)}{\omega(k)}\right),$
      $P(k+1) = R(k) + \beta(k+1)\bigl(P(k) - \omega(k)V(k)\bigr),$
      $V(k+1) = AP(k+1) + P(k+1)A^T + \sum_{j=1}^m N_jP(k+1)N_j^T,$
      $\sigma(k+1) = \langle V(k+1), \tilde{R}(1)\rangle,$
      $\alpha(k+1) = \frac{\rho(k+1)}{\sigma(k+1)}, \quad S(k+1) = R(k) - \alpha(k+1)V(k+1),$
      $T(k+1) = AS(k+1) + S(k+1)A^T + \sum_{j=1}^m N_jS(k+1)N_j^T,$
      $\omega(k+1) = \frac{\langle S(k+1), T(k+1)\rangle}{\langle T(k+1), T(k+1)\rangle}, \quad R(k+1) = S(k+1) - \omega(k+1)T(k+1),$
      $X(k+1) = X(k) + \alpha(k+1)P(k+1) + \omega(k+1)S(k+1).$
```

```
Algorithm 4: The variant form of the Bi-CGSTAB algorithm
  step 1: Choose initial matrix $X(1)$;
  step 2: Compute $R(1) = \mathcal{P}(X(1)) - 2\gamma\hat{C}$, pick arbitrary matrix $\tilde{R}(1)$ (for example $\tilde{R}(1) = R(1)$);
  step 3: Set $V(1) = P(1) = 0$, $\rho(1) = \alpha(1) = \omega(1) = 1$;
  step 4: While $\|R(k)\| > \mathrm{tol}$ do:
      $\rho(k+1) = \langle R(k), \tilde{R}(1)\rangle, \quad \beta(k+1) = \left(\frac{\rho(k+1)}{\rho(k)}\right)\left(\frac{\alpha(k)}{\omega(k)}\right),$
      $P(k+1) = R(k) + \beta(k+1)\bigl(P(k) - \omega(k)V(k)\bigr), \quad V(k+1) = \mathcal{P}\bigl(P(k+1)\bigr),$
      $\sigma(k+1) = \langle V(k+1), \tilde{R}(1)\rangle, \quad \alpha(k+1) = \frac{\rho(k+1)}{\sigma(k+1)},$
      $S(k+1) = R(k) - \alpha(k+1)V(k+1), \quad T(k+1) = \mathcal{P}\bigl(S(k+1)\bigr),$
      $\omega(k+1) = \frac{\langle S(k+1), T(k+1)\rangle}{\langle T(k+1), T(k+1)\rangle}, \quad R(k+1) = S(k+1) - \omega(k+1)T(k+1),$
      $X(k+1) = X(k) + \alpha(k+1)P(k+1) + \omega(k+1)S(k+1).$
```

forms of the CRS algorithm to solve the periodic Sylvester matrix equation by applying Kronecker product and vectorization operator [21]. In fact, in many cases, the CRS algorithm converges twice as fast as the BiCR algorithm [22, 23]. The BiCR method can be derived from the preconditioned conjugate residual (CR) algorithm [24]. In exact arithmetic, they terminate after a limited number of iterations. In short, we can expect that the CRS algorithm will work well in many cases. The numerical examples in Sect. 4 are shown to address this point.

It is easy to see that the CRS algorithm cannot be directly used to solve the generalized Lyapunov equation. Similarly, we need to modify the CRS algorithm to the matrix version. The matrix version of the CRS algorithm is summarized in Algorithm 5. The variant version of the CRS algorithm is shown in Algorithm 6.

Viewing the iteration schemes, it can be seen that Algorithm 5 is a simple matrix form of the CRS algorithm. Hence, Algorithm 5 has the same properties as the CRS algorithm. Algorithm 6 is the variant version of the CRS algorithm, which has high computing efficiency. The numerical examples have verified the validity of the iteration algorithms in Sect. 4.

Remark 3.2 The BICGSTAB and CRS algorithms have an orthogonality property similar to that of BICR and thus are omitted.

The convergence result of Algorithms 2 to 6 has been summarized in Theorem 3.3.

Theorem 3.3 *For the generalized Lyapunov equation* (1), *if Algorithms* 2 *to* 6 *do not break down by zero division, for any initial matrix* $X(1) \in \mathbb{R}^{n\times n}$, *Algorithms* 2 *to* 6 *can compute the solution of* (1) *within a finite number of iterations in the absence of the roundoff error.*

Algorithm 5: Matrix form of the CRS algorithm

```
step 1: Choose $X(1)$, compute $R(1) = -C - AX(1) - X(1)A^T - \sum_{j=1}^m N_jX(1)N_j^T$;
step 2: Pick a matrix $R(1)^*$, $\langle AR(1) + R(1)A^T + \sum_{j=1}^m N_jR(1)N_j^T, R_1^*\rangle \neq 0$,
$R(1)^* = P(1) = U(1) = R(1)$, $S = A^TR(1)^* + R(1)^*A + \sum_{j=1}^m N_j^TR(1)^*N_j$, $k = 1$;
step 3: While $\|R(k)\| > tol$;
    $$V(k) = AP(k) + P(k)A^T + \sum_{j=1}^m N_jP(k)N_j^T, \qquad \alpha(k) = \frac{\langle R(k), S\rangle}{\langle V(k), S\rangle},$$
    $$Q(k) = U(k) - \alpha(k)V(k), \qquad X(k+1) = X(k) + \alpha(k)\bigl(U(k) + Q(k)\bigr),$$
    $$W(k) = A\bigl(U(k) + Q(k)\bigr) + \bigl(U(k) + Q(k)\bigr)A^T + \sum_{j=1}^m N_j\bigl(U(k) + Q(k)\bigr)N_j^T,$$
    $$R(k+1) = R(k) - \alpha(k)W(k), \qquad \beta(k) = \frac{\langle R(k+1), S\rangle}{\langle R(k), S\rangle},$$
    $$U(k+1) = R(k+1) + \beta(k)Q(k), \qquad P(k+1) = U(k+1) + \beta(k)\bigl(Q(k) + \beta(k)P(k)\bigr).$$
```

Algorithm 6: The variant form of the CRS algorithm

```
step 1: Choose $X(1)$, compute $R(1) = \mathcal{P}(X(1)) - 2\gamma\hat{C}$;
step 2: Pick a matrix $R(1)^*$, $\langle AR(1) + R(1)A^T + \sum_{j=1}^m N_jR(1)N_j^T, R_1^*\rangle \neq 0$,
$R(1)^* = P(1) = U(1) = R(1)$, $S = \mathcal{P}^\top(R(1)^*)$, $k = 1$;
step 3: While $\|R(k)\| > tol$:
    $V(k) = \mathcal{P}(Pk)$,    $\alpha(k) = \frac{\langle R(k), S\rangle}{\langle V(k), S\rangle}$,
    $Q(k) = U(k) - \alpha(k)V(k)$,    $X(k+1) = X(k) + \alpha(k)(U(k) + Q(k))$,
    $W(k) = \mathcal{P}^{-1}(U(k) + Q(k))$,    $R(k+1) = R(k) - \alpha(k)W(k)$,
    $\beta(k) = \frac{\langle R(k+1), S\rangle}{\langle R(k), S\rangle}$,    $U(k+1) = R(k+1) + \beta(k)Q(k)$,
    $P(k+1) = U(k+1) + \beta(k)(Q(k) + \beta(k)P(k))$.
```

4 Numerical experiments

In this section, we give several examples to show the numerical feasibility and effectiveness of Algorithm 1 (BICR), Algorithm 3 (Bi-CGSTAB algorithm), Algorithm 5 (CRS algorithm) and their improved algorithms, including Algorithm 2 (Var-BICR algorithm), Algorithm 4 (Var-Bi-CGSTAB algorithm), Algorithm 6 (Var-CRS algorithm). Set $tol = 1.0e-8$. The numerical behavior of iteration methods will be listed with respect to the number of iteration steps (ITs), the computing time (CPU)(s) and relative residual error (Error). All experiments are performed in Matlab (version R2017a) with double precision on a personal computer with 3.20 GHz central processing unit (Inter(R) Core(TM) i5-6500 CPU), 6.00G memory and Windows 7 operating system.

Example 4.1 Consider the generalized Lyapunov equation (1) with

$$A = (a_{ij})_{n\times n} = \begin{cases} 1.6 & i = j, \\ 0.3 & |i-j| = 1, \\ 0 & \text{else}, \end{cases} \qquad N = (n_{ij})_{n\times n} = \begin{cases} 0.05 & i = j, \\ -0.01 & |i-j| = 1, \\ 0 & \text{else}, \end{cases}$$

$$B = -A^{-1}\begin{pmatrix} 0_{n_1\times n_1} & 0_{n_1\times n_2} \\ 0_{n_2\times n_1} & I_{n_2} \end{pmatrix} A^{-1}, \qquad C = BB^T, \qquad N_j = 0.1\times j \times N \quad (j = 1,\dots,5).$$

Set the initial value

$$X(1) = 0, \qquad S(1) = I.$$

We use Table 2 to show the error analysis for this example.

Moreover, when $n = 600$, we use Fig. 1 to show the error analysis of Algorithms 1 to 6.

By comparing with these algorithms, it is clear that the algorithms' efficiency will greatly be improved after using a Cayley transformation. The variant versions of the Bi-CGSTAB and CRS algorithms have the best efficiency.

Table 2 Numerical results for Example 4.1

Algorithm	*n*	ITs	Error	CPU	*n*	ITs	Error	CPU
BICR	400	112	7.8882e-7	142.7510	600	112	8.0506e-7	667.4057
Var-BICR	400	14	4.4326e-8	19.8881	600	14	4.4408e-8	83.3214
CRS	400	15	3.2668e-9	13.4433	600	15	3.2507e-9	52.5653
Var-CRS	400	5	2.6410e-9	3.2538	600	5	2.6351e-11	16.3354
Bi-CGSTAB	400	16	9.5717e-8	10.1564	600	16	9.5980e-8	55.0818
Var-Bi-CGSTAB	400	4	6.8168e-9	2.4104	600	4	6.8092e-9	11.6166

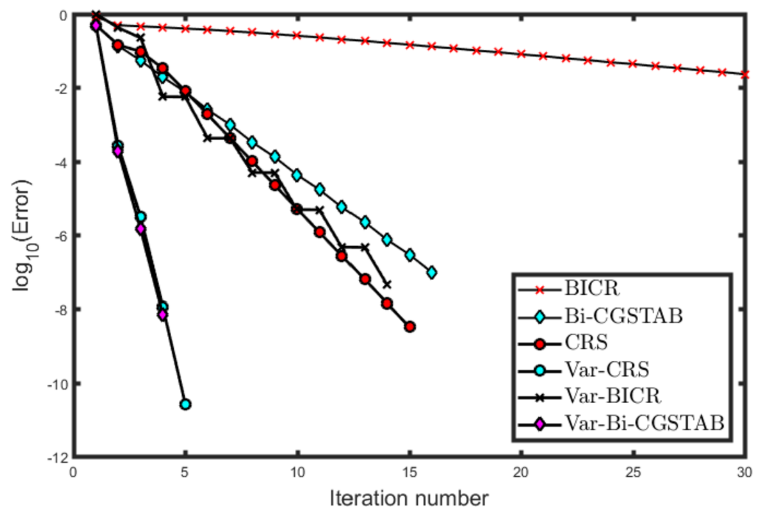

Figure 1 Comparison between the residual error of Algorithms 1 to 6 for Example 4.1

Example 4.2 Let P be the block tridiagonal sparse $m^2 \times m^2$ matrix, given by a finite difference disretization of the heat equation (3) on an $m \times m$-mesh, i.e.,

$$P = I \otimes T_m + T_m \otimes I \in \mathbb{R}^{n\times n}, \qquad T_k = \begin{bmatrix} -2 & 1 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}.$$

If the Robin condition is imposed on the whole boundary, then we have

$$A = P + E_1 \otimes I + I \otimes E_1 + E_m \otimes I + I \otimes E_m,$$

where $E_j = e_j e_j^T$ with canonical unit vector e_j. The coefficient matrices N_j and the columns b_j of B corresponding to the left, upper, lower, and right boundaries are given by

$$N_1 = E_1 \otimes I, \qquad N_2 = I \otimes E_1, \qquad N_3 = E_m \otimes I, \qquad N_4 = I \otimes E_m,$$
$$b_1 = E_1 \otimes e, \qquad b_2 = e \otimes E_1, \qquad b_3 = E_m \otimes e, \qquad b_4 = e \otimes E_m.$$

Then the above heat equation's optimal control problem reduces to solving the generalized Lyapunov equation (1).

Table 3 Numerical results for Example 4.2

Algorithm	n	ITs	Error	CPU	n	ITs	Error	CPU
BICR	400	112	3.7523e-6	48.5777	900	110	7.3974e-6	325.2917
Var-BICR	400	57	1.1463e-7	30.7176	900	56	7.3974e-6	167.4174
CRS	400	25	8.9720e-8	6.8191	900	23	8.8132e-9	31.8204
Var-CRS	400	15	7.0257e-9	2.3428	900	13	5.4997e-9	16.4174
Bi-CGSTAB	400	26	9.8350e-8	5.1010	900	24	9.6342e-8	45.3461
Var-Bi-CGSTAB	400	8	7.1782e-8	1.6054	900	7	9.0689e-8	10.1155

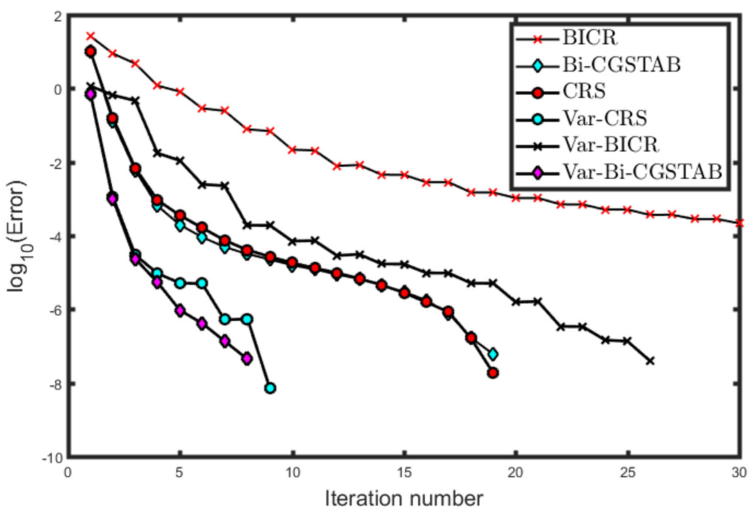

Figure 2 Comparison between the residual error of Algorithms 1 to 6 for Example 4.2

We use Table 3 to show the residual error analysis. It is obvious that the effect of the Var-Bi-CGSTAB algorithm is optimal compared with other algorithms.

Further, we use Fig. 2 to show the error analysis when $n = 64$. It can be seen that the variant versions of the algorithms perform better.

Example 4.3 Consider the RC trapezoidal circuit with m resistors with g extensions

$$\begin{cases} \dot{x}(t) = Ax(t) + Nx(t)u(t) + bu(t), \\ y(t) = c^T x(t). \end{cases}$$

Since the original system is nonlinear, it is linearized by the second-order Carleman bilinear method to obtain a system of order $n = m + m^2$.

The matrices A, N and b can be referred to [25]. The corresponding generalized Lyapunov equation is

$$AX + XA^T + NXN^T + C = 0.$$

We use Table 4 to show the residual error analysis. Further, we use Fig. 3 to show the error analysis when $n = 8$. It can be seen that the Var-Bi-CGSTAB algorithm performs best.

Table 4 Numerical results for Example 4.3

Algorithm	*n*	ITs	Error	CPU	*n*	ITs	Error	CPU
CRS	110	68	2.7672e-9	0.4970	420	81	4.2339e-7	11.9000
var-CRS	110	27	9.8696e-6	0.1882	420	41	6.3712e-9	7.4257
Bi-CGSTAB	110	52	4.7429e-8	0.3558	420	75	6.1473e-8	12.7971
Var-Bi-CGSTAB	110	32	8.1503e-8	0.2264	420	33	4.9993e-8	5.6614

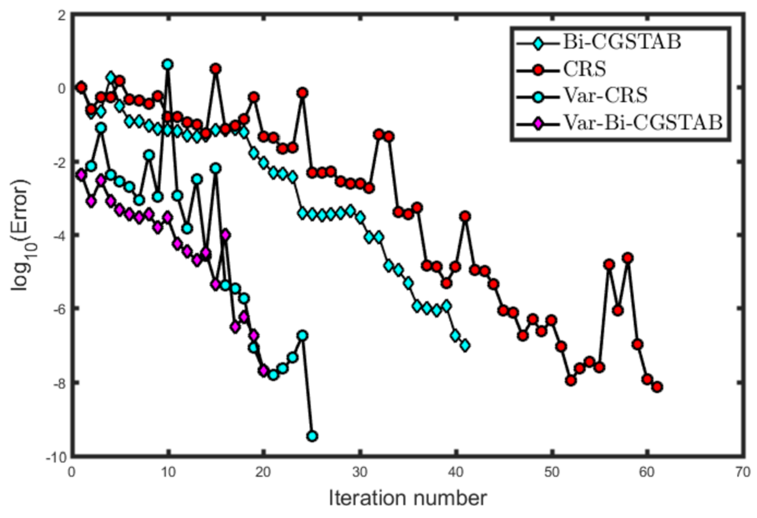

Figure 3 Comparison between the residual error of Algorithms 3 to 6 for Example 4.3

Remark 4.1 From the three numerical examples above, it can be seen that the variant algorithms proposed in this paper will greatly improve the operating efficiency. In other words, the conjugate gradient-like methods are more efficient than the generalized Setin equation.

5 Conclusions

This paper has proposed the matrix versions of the BICR algorithm, Bi-CGSTAB algorithm, and CRS algorithm to solve the generalized Lyapunov equation (1). Then we have introduced the variant versions of these three algorithms. Finally, we have provided numerical examples to illustrate the feasibility and effectiveness of the derived algorithms.

Appendix: Proof of Theorem 3.1

We prove Theorem 3.1 by mathematical induction to v and u. It is enough to prove (8)–(11) for $1 \leq u < v \leq r$.

(i) If $v = 2$, $u = 1$, then we have

$$\begin{aligned} \operatorname{tr}\bigl(R(2)^T W(1)\bigr) &= \operatorname{tr}\bigl(\bigl(R(1) - \alpha(1) - W(1)\bigr)^T W(1)\bigr) \\ &= \operatorname{tr}\bigl(R(1)^T W(1)\bigr) - \operatorname{tr}\bigl(W(1)^T R(1)\bigr) \\ &= 0, \end{aligned}$$

$$\operatorname{tr}\bigl(S(2)^T Z(1)\bigr) = \operatorname{tr}\bigl(\bigl(S(1) - \beta(1) Z(1)\bigr)^T Z(1)\bigr)$$

$$= \operatorname{tr}\big(S(1)^T Z(1)\big) - \operatorname{tr}\big(Z(1)^T S(1)\big)$$
$$= 0,$$

$$\begin{aligned}
&\operatorname{tr}\big(Z(2)^T Z(1)\big) \\
&\quad = \operatorname{tr}\left(\left(A^T R(2) + R(2)A + \sum_{j=1}^{m} N_j^T R(2) N_j - \eta(1) Z(1)\right)^T Z(1)\right) \\
&\quad = \operatorname{tr}\big(\big(A^T R(2)\big)^T Z(1)\big) + \operatorname{tr}\big(\big(R(2)A\big)^T Z(1)\big) + \operatorname{tr}\left(\left(\sum_{j=1}^{m} N_j^T R(2) N_j\right)^T Z(1)\right) \\
&\qquad - \operatorname{tr}\big(Z(1)^T \big(A^T R(2)\big)\big) - \operatorname{tr}\big(Z(1)^T \big(R(2)A\big)\big) - \operatorname{tr}\left(Z(1)^T \left(\sum_{j=1}^{m} N_j^T R(2) N_j\right)\right) \\
&\quad = 0,
\end{aligned}$$

and

$$\begin{aligned}
\operatorname{tr}\big(W(2)^T W(1)\big) &= \operatorname{tr}\left(\left(AS(2) + S(2)A^T + \sum_{j=1}^{m} N_j S(2) N_j^T - \gamma(1) W(1)\right)^T W(1)\right) \\
&= \left(\left(AS(2) + S(2)A^T + \sum_{j=1}^{m} N_j S(2) N_j^T\right)^T W(1)\right) \\
&\quad - \operatorname{tr}\left(W(1)^T \left(AS(2) + S(2)A^T + \sum_{j=1}^{m} N_j S(2) N_j^T\right)\right) \\
&= 0.
\end{aligned}$$

Thus when $u = 1$, $v = 2$, (8)–(11) is true.

(ii) Now for $u < w < r$, we assume that

$$\begin{aligned}
&\operatorname{tr}\big(R(w)^T W(u)\big) = 0, \\
&\operatorname{tr}\big(S(w)^T Z(u)\big) = 0, \\
&\operatorname{tr}\big(Z(w)^T Z(u)\big) = 0, \\
&\operatorname{tr}\big(W(w)^T W(u)\big) = 0.
\end{aligned}$$

(iii) Next, we will prove (8)–(11) for $w + 1$. Using the induction hypothesis, we get

$$\begin{aligned}
&\operatorname{tr}\big(R(w+1)^T W(u)\big) = \operatorname{tr}\big(\big(R(w) - \alpha(w) W(w)\big)^T W(u)\big) = 0 \\
&\operatorname{tr}\big(S(w+1)^T Z(u)\big) = \operatorname{tr}\big(\big(S(w) - \beta(w) Z(w)\big)^T Z(u)\big) = 0,
\end{aligned}$$

and

$$\begin{aligned}
&\operatorname{tr}\big(Z(w+1)^T Z(u)\big) \\
&\quad = \operatorname{tr}\left(\left(A^T R(w+1) + R(w+1)A + \sum_{j=1}^{m} N_j^T R(w+1) N_j - \eta(w) Z(w)\right)^T Z(u)\right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\beta(u)}\Bigg[\operatorname{tr}\big(R(w+1)^T\big(A\big(S(u)-S(u+1)\big)\big)\big)\\
&\quad + \operatorname{tr}\big(R(w+1)^T\big(S(u)-S(u+1)A^T\big)\big)\\
&\quad + \operatorname{tr}\Bigg(R(w+1)^T\Bigg(\sum_{j=1}^{m} N_j^T\big(S(u)-S(u+1)\big)N_j\Bigg)\Bigg)\Bigg]\\
&= \frac{1}{\beta(u)}\big[\operatorname{tr}\big(R(w+1)^T\big(W(u)+\gamma(u-1)W(u-1)\big)\big)\\
&\quad - \operatorname{tr}(R(w+1)^T\big(W(u+1)+\gamma(u)W(u)\big)\big]\\
&= -\frac{1}{\beta(u)}\big[\operatorname{tr}\big(R(w+1)^T W(u+1)\big)\big], \qquad (12)\\
&\operatorname{tr}\big(W(w+1)^T W(u)\big)\\
&= \operatorname{tr}\Bigg(\Bigg(AS(w+1)+S(w+1)A^T+\sum_{j=1}^{m} N_j S(w+1)N_j^T\\
&\quad - \gamma(w)W(w)\Bigg)^T w(u)\Bigg)\\
&= \operatorname{tr}\big(S(w+1)^T\big(AS(w+1)\big)^T\big) + \operatorname{tr}\big(S(w+1)^T\big(S(w+1)A^T\big)^T\big)\\
&\quad + \operatorname{tr}\Bigg(S(w+1)^T\Bigg(\sum_{j=1}^{m} N_j S(w+1)N_j^T\Bigg)^T\Bigg)\\
&= \frac{1}{\alpha(u)}\Bigg[\operatorname{tr}\big(S(w+1)^T\big(A\big(R(u)-R(u+1)\big)\big)\big)\\
&\quad + \operatorname{tr}\big(S(w+1)^T\big(R(u)-R(u+1)A^T\big)\big)\\
&\quad + \operatorname{tr}\Bigg(S(w+1)^T\Bigg(\sum_{j=1}^{m} N_j^T\big(R(u)-R(u+1)\big)N_j\Bigg)\Bigg)\Bigg]\\
&= \frac{1}{\alpha(u)}[\operatorname{tr}\big(S(w+1)^T\big(Z(u)+\eta(u-1)Z(u-1)\big) - Z(u+1) - \eta(u)Z(u)\big)\\
&= -\frac{1}{\alpha(u)}\big[\operatorname{tr}\big(S(w+1)^T Z(u+1)\big)\big]. \qquad (13)
\end{aligned}$$

For $u = w$, again from the induction hypothesis we can obtain

$$\begin{aligned}
&\operatorname{tr}\big(R(w+1)^T W(w)\big) = \operatorname{tr}\big(\big(R(w)-\alpha(w)W(w)^T\big)W(w)\big),\\
&\operatorname{tr}\big(R(w)^T W(w)\big) - \operatorname{tr}\big(W(w)^T R(w)\big) = 0,\\
&\operatorname{tr}\big(S(w+1)^T Z(w)\big) = \operatorname{tr}\big(\big(S(w)-\beta(w)Z(w)\big)^T Z(w)\big)\\
&\qquad = \operatorname{tr}\big(S(w)^T Z(w)\big) - \operatorname{tr}\big(Z(w)^T S(w)\big)\\
&\qquad = 0,
\end{aligned}$$

and

$$\begin{aligned}
\operatorname{tr}\bigl(Z(w+1)^T Z(w)\bigr) &= \operatorname{tr}\Biggl(\Biggl(A^T R(w+1) + R(w+1)A \\
&\quad + \sum_{j=1}^{m} N_j^T R(w+1) N_j - \eta(w) Z(w)\Biggr)^T Z(w)\Biggr) \\
&= \operatorname{tr}\bigl(\bigl(A^T R(w+1)\bigr)^T Z(w)\bigr) + \operatorname{tr}\bigl(\bigl(R(w+1)A\bigr)^T Z(w)\bigr) \\
&\quad + \operatorname{tr}\Biggl(\Biggl(\sum_{j=1}^{m} N_j^T R(w+1) N_j\Biggr)^T Z(w)\Biggr) \\
&\quad - \operatorname{tr}\bigl(Z(w)^T \bigl(A^T R(w+1)\bigr)\bigr) - \operatorname{tr}\bigl(Z(w)^T \bigl(R(w+1)A\bigr)\bigr) \\
&\quad - \operatorname{tr}\Biggl(Z(w)^T \Biggl(\sum_{j=1}^{m} N_j^T R(w+1) N_j\Biggr)\Biggr) \\
&= 0,
\end{aligned}$$

$$\begin{aligned}
&\operatorname{tr}\bigl(W(w+1)^T W(w)\bigr) \\
&\quad = \operatorname{tr}\Biggl(\Biggl(AS(w+1) + S(w+1)A^T \\
&\quad\quad + \sum_{j=1}^{m} N_j S(w+1) N_j^T - \gamma(w) W(w)\Biggr)^T W(w)\Biggr) \\
&\quad = \Biggl(\Biggl(AS(w+1) + S(w+1)A^T + \sum_{j=1}^{m} N_j S(w+1) N_j^T\Biggr)^T W(w)\Biggr) \\
&\quad\quad - \operatorname{tr}\Biggl(W(w)^T \Biggl(AS(w+1) + S(w+1)A^T + \sum_{j=1}^{m} N_j S(w+1) N_j^T\Biggr)\Biggr) \\
&\quad = 0.
\end{aligned}$$

Noting that $\operatorname{tr}(Z(w)^T Z(u)) = 0$, $\operatorname{tr}(R(w+1)^T W(w)) = 0$ with (12) we deduce that

$$\operatorname{tr}\bigl(Z(w+1)^T Z(u)\bigr) = 0.$$

Similarly from $\operatorname{tr}(W(w)^T W(u)) = 0$, $\operatorname{tr}(S(w)^T Z(w)) = 0$ and (13), it can be seen that

$$\operatorname{tr}\bigl(W(w+1)^T W(u)\bigr) = 0.$$

Hence, (8)–(11) hold true for $w+1$. Using mathematical induction, we complete the proof.

Acknowledgements
The research is supported by Hunan Key Laboratory for Computation and Simulation in Science and Engineering, School of Mathematics and Computational Science, Xiangtan University.

Funding
The work was supported in part by National Natural Science Foundation of China (11771368, 11771370) and the Project of Education Department of Hunan Province (19A500).

Availability of data and materials
Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
All authors read and approved the final manuscript.

Publisher's Note

Received: 23 September 2020 Accepted: 15 April 2021 Published online: 28 April 2021

References

1. Kurschner, P.: Efficient Low-Rank Solution of Large-Scale Matrix Equations. Dissertation, Otto von Guericke Universitat, Magdeburg (2016)
2. Simoncini, V.: Analysis of the rational Krylov subspace projection method for large-scale algebraic Riccati equations. SIAM J. Matrix Anal. Appl. **37**, 1655–1674 (2016)
3. Paolo, D.A., Alberto, I., Antonio, R.: Realization and structure theory of bilinear dynamical systems. SIAM J. Control Optim. **12**, 517–535 (1974)
4. Samir, A., Baiyat, A.L., Bettayeb, M.A., Saggaf, M.A.L.: New model reduction scheme for bilinear systems. Int. J. Syst. Sci. **25**, 1631–1642 (1994)
5. Kleinman, D.L.: On the stability of linear stochastic systems. IEEE Trans. Autom. Control **14**, 429–430 (1969)
6. Benner, P., Damm, T.: Lyapunov equations, energy functionals, and model order reduction of bilinear and stochastic systems. SIAM J. Control Optim. **49**, 686–711 (2011)
7. Gray, W.S., Mesko, J.: Energy functions and algebraic Gramians for bilinear systems. IFAC Proc. Vol. **31**, 101–106 (1998)
8. Dorissen, H.: Canonical forms for bilinear systems. Syst. Control Lett. **13**, 153–160 (1989)
9. Damm, T.: Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equation. Numer. Linear Algebra Appl. **15**, 853–871 (2008)
10. Fan, H.Y., Weng, P., Chu, E.: Numerical solution to generalized Lyapunov, Stein and rational Riccati equations in stochastic control. Numer. Algorithms **71**, 245–272 (2016)
11. Li, S.Y., Shen, H.L., Shao, X.H.: PHSS iterative method for solving generalized Lyapunov equations. Mathematics **7**, 1–13 (2019)
12. Sogabe, T., Sugihara, M., Zhang, S.L.: An extension of the conjugate residual method to nonsymmetric linear systems. J. Comput. Appl. Math. **226**, 103–113 (2009)
13. Stiefel, E.L.: Relaxationsmethoden bester strategie zur losung linearer gleichungssysteme. Comment. Math. Helv. **29**, 157–179 (1955)
14. Abea, K., Fujino, S.: Converting BiCR method for linear equations with complex symmetric matrices. Appl. Math. Comput. **321**, 564–576 (2018)
15. Sonneveld, P.: CGS, a fast Lanczos-type solver for nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **10**, 36–52 (1989)
16. Vander, H.A.: Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **13**, 631–644 (1992)
17. Hajarian, M.: Developing Bi-CG and Bi-CR methods to solve generalized Sylvester-transpose matrix equations. Int. J. Autom. Comput. **11**, 25–29 (2014)
18. Chen, J., McInnes, L.C., Zhang, H.: Analysis and practical use of flexible BiCGStab. J. Sci. Comput. **68**, 803–825 (2016)
19. Zhang, L.T., Zuo, X.Y., Gu, T.X., Huang, T.Z.: Conjugate residual squared method and its improvement for non-symmetric linear systems. Int. J. Comput. Math. **87**, 1578–1590 (2010)
20. Zhang, L.T., Huang, T.Z., Gu, T.X., Zuo, X.Y.: An improved conjugate residual squared algorithm suitable for distributed parallel computing. Microelectron. Comput. **25**, 12–14 (2008) (in Chinese)
21. Chen, L.J., Ma, C.F.: Developing CRS iterative methods for periodic Sylvester matrix equation. Adv. Differ. Equ. **1**, 1–11 (2019)
22. Zhao, J., Zhang, J.H.: A smoothed conjugate residual squared algorithm for solving nonsymmetric linear systems. In: 2009 Second Int. Confe. Infor. Comput. Sci., vol. 4, pp. 364–367 (2009)
23. Sogabe, T., Zhang, S.L.: Extended conjugate residual methods for solving nonsymmetric linear systems. In: International Conference on Numerical Optimization and Numerical Linear Algebra, pp. 88–99 (2003)
24. Sogabe, T., Sugihara, M., Zhang, S.L.: An extension of the conjugate residual method to nonsymmetric linear systems. J. Comput. Appl. Math. **226**, 103–113 (2009)
25. Bai, Z.J., Skoogh, D.: A projection method for model reduction of bilinear dynamical systems. Linear Algebra Appl. **415**, 406–425 (2006)