mathematics

MDPI

Article

Low-Rank Methods for Solving Discrete-Time Projected Lyapunov Equations

Yiqin Lin

School of Science, Hunan University of Science and Engineering, Yongzhou 425199, China; yqlin@aliyun.com

Abstract: In this paper, we consider the numerical solution of large-scale discrete-time projected Lyapunov equations. We provide some reasonable extensions of the most frequently used low-rank iterative methods for linear matrix equations, such as the low-rank Smith method and the low-rank alternating-direction implicit (ADI) method. We also consider how to reduce complex arithmetic operations and storage when shift parameters are complex and propose a partially real version of the low-rank ADI method. Through two standard numerical examples from discrete-time descriptor systems, we will show that the proposed low-rank alternating-direction implicit method is efficient.

Keywords: discrete-time projected Lyapunov equation; Smith method; ADI method; low-rank method; matrix pencil; D-stable

MSC: 65F10; 65F30; 15A22; 15A24

1. Introduction

Solving linear matrix equations is a very important topic in control theory. Such equations include Lyapunov equations and Sylvester equations. Let $E, A \in \mathbb{R}^{n\times n}$, where E is singular. Assume that the pencil $\lambda E - A$ is d-stable. That is, the moduli of all finite eigenvalues of the pencil $\lambda E - A$ are less than 1. According to [1], there exist nonsingular $n \times n$ matrices W, T that transform E, A into a Weierstrass canonical form, i.e.,

$$E = W\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix}T, \quad A = W\begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix}T \tag{1}$$

with $J \in \mathbb{R}^{n_f\times n_f}$ and $N \in \mathbb{R}^{n_\infty\times n_\infty}$. Define the left and right spectral projection matrices P_l, P_r by

$$P_l = W\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}W^{-1}, \quad P_r = T^{-1}\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}T. \tag{2}$$

In this paper, we focus on the numerical solution of the discrete-time projected Lyapunov equation

$$EXE^T - AXA^T = P_lBB^TP_l^T, \quad X = P_rXP_r^T, \tag{3}$$

where $X \in \mathbb{R}^{n\times n}$ is the solution, $B \in \mathbb{R}^{n\times m}, m \ll n$. Here and in the following, the superscript T denotes the transpose of a vector or a matrix. Since $\lambda E - A$ is d-stable, (3) has a symmetric positive semi-definite solution; see, for example, ref. [2]. The discrete-time Lyapunov equation is also called the Stein equation in the literature. By using the Kronecker product [3], the first equation in (3) can be formulated as $(E \otimes E - A \otimes A)\mathrm{vec}(X) = \mathrm{vec}(P_lBB^TP_l^T)$, where $\mathrm{vec}(X) = [x_1^T, x_2^T, \cdots, x_n^T]^T$, and x_i is the i-th column of X.

The Stein equation with nonsingular E plays an essential role in discrete-time dynamical systems, including stability analysis and control [4–6], model reduction [7–12], solutions of discrete-time algebraic Riccati equations (by Newton's method) in optimal control [13], and the restoration of images [14]. In contrast, the projected Stein equation arises in the

check for updates

Citation: Lin, Y. Low-Rank Methods for Solving Discrete-Time Projected Lyapunov Equations. *Mathematics* **2024**, *12*, 1166. https://doi.org/10.3390/math12081166

Academic Editors: Alicia Cordero and Ioannis K. Argyros

Received: 2 January 2024
Revised: 5 April 2024
Accepted: 11 April 2024
Published: 12 April 2024

balanced truncation model reduction [2] of discrete-time descriptor systems. In the positive real and bounded real balanced truncation model reduction of discrete-time descriptor systems, we also need to solve a projected Stein equation at each iteration step of Newton's method for projected Riccati equations.

In the past few decades, many researchers have focused on constructing numerically robust algorithms for the standard Stein equation, i.e., with E being the identity matrix. For example, a standard direct method was provided in [15], which is a direct extension of the well-known Bartels–Stewart algorithm [16] for continuous-time Lyapunov equations $AX + XA^T = Q$ to the standard Stein equation. Hammarling [17] proposed a variant of the Bartels–Stewart algorithm for both the continuous-time and discrete-time cases. This variant is named the Hammarling method in the literature and aims to compute the Cholesky factor of the solution, which is desired in the balanced truncation model order reduction of discrete-time systems. The Hammarling algorithm was further improved in [18,19] by using a rank-2 updating formula. These approaches are based on the real Schur decomposition, require a computational complexity of $O(n^3)$ flops, and thus are only suitable for small to moderately sized problems.

It is known that the solution matrix of a continuous-time Lyapunov equation has a low numerical rank in cases where it has a low-rank right-hand side; see, e.g., ref. [20]. Specifically, the low numerical rank means that the singular values of the solution X decay very rapidly. Penzl [21] showed theoretically that the singular values of the solution decay exponentially for the continuous-time Lyapunov equation with a symmetric coefficient matrix and a low-rank right-hand side. Baker, Embree, and Sabino [22] considered the nonsymmetric case, and they explained that a larger departure from normality probably means a faster decay of singular values. The fact that the solution has a rapid decay of singular values and can be well approximated by its low-rank factorization now enables the use of numerous iterative methods that seek accurate low-rank approximations to the solution. These iterative methods include the low-rank Smith method [23,24], the Cholesky factor alternating-direction implicit (ADI) method [25], the (generalized) matrix sign function method [26], and the extended Krylov subspace method [27], to name a few. For the continuous-time projected Lyapunov equation, Stykel [28] extended the low-rank ADI method and the low-rank Smith method to compute low-rank approximations to the solution. In [13], the ADI method was extended to discrete-time Lyapunov equations and was further improved to compute the real factors in [29] by utilizing the technique that was proposed in [30] for continuous-time Lyapunov equations.

In recent years, numerical methods for continuous-time Lyapunov equations have been further considered. In [31], a class of low-rank iterative methods is proposed by using Runge–Kutta integration methods. It is shown that a special instance of this class of methods is equivalent to the low-rank ADI method. In [32], Benner, Palitta, and Saak further improved the low-rank ADI. They used the extended Krylov subspace method to solve the shifted linear system at each iteration. It is shown that by using only a single subspace, all the shifted linear systems can be solved to achieve a prescribed accuracy. In [33], the authors considered the inexact rational Krylov subspace method and low-rank iteration, in which a shifted linear system of equations is solved inexactly. In [34,35], the choice of shift parameters is considered, and some selection techniques are proposed to achieve a fast convergence for the low-rank ADI method.

In this paper, we first transform the projected generalized Stein Lyapunov Equation (3) to an equivalent projected standard Stein equation and then extend the low-rank Smith method to the projected standard equation. After this, we extend the low-rank ADI method to (3) and propose how to compute the real low-rank factor by following the idea in [29,30]. We also consider the choice of ADI shift parameters. Finally, through two standard numerical examples from discrete-time descriptor systems, we show the efficiency of the proposed low-rank ADI method.

The main contributions of this paper include the following:

- The low-rank ADI method is extended to solve the discrete-time projected Lyapunov equation.
- A partially real low-rank ADI algorithm is proposed.
- Two numerical examples are presented to demonstrate the good convergence of the low-rank ADI method.

The low-rank ADI method is one of the most commonly used iterative methods for solving linear matrix equations. It has a good convergence curve, although the shift parameters are not optimal. Moreover, it always produces a low-rank positive semi-definite approximate solution for Lyapunov equations, which is desired for some applications, such as the balanced truncation model order reduction. In contrast, the Krylov subspace method cannot guarantee the generation of the positive semi-definite solution. The main drawback of the low-rank ADI method is that it requires selecting shifts and solving one linear system of equations for each iteration.

The rest of the paper proceeds as follows. In Section 2, we reformulate (3) and propose the low-rank Smith method. In Section 3, we extend the real version of the low-rank ADI method for the projected Stein equation. Section 4 is devoted to two numerical examples. Finally, conclusions are given in Section 5.

2. Low-Rank Smith Method

The Smith method [36] was originally proposed for solving the continuous-time Sylvester equation $AX + X\widetilde{A} = C$. First, the continuous-time equation is equivalently transformed to a discrete-time equation via a Cayley transform. Then, the Smith iteration is derived from the series representation of the solution. For the projected Stein Equation (3), the Smith method can be applied directly without the Cayley transform.

Due to the singularity of E, its inverse does not exist. We use the $\{2\}$-inverse E^- of E, which is defined by

$$E^- = P_r(EP_r + A(I - P_r))^{-1} = (P_lE + (I - P_l)A)^{-1}P_l = T^{-1}\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}W^{-1},$$

see, e.g., ref. [37]. For the generalized inverse of a singular matrix, the interested reader is referred to [38].

Multiplying the first equation in (3) from the left and right by E^- and $(E^-)^T$ and using $E^-E = P_r$, $E^-P_l = E^-$, and $X = P_rXP_r^T$, we obtain

$$X - (E^-A)X(E^-A)^T = E^-B(E^-B)^T, \quad X = P_rXP_r^T. \tag{4}$$

The unique solution of (4) can be formulated as

$$X = \sum_{j=0}^{\infty}(E^-A)^jE^-B((E^-A)^jE^-B)^T. \tag{5}$$

Since the pencil $\lambda E - A$ is d-stable, the spectrum radius $\rho(E^-A)$ of E^-A, which is defined by $\rho(E^-A) = \max_{\lambda\in\Lambda(E^-A)}|\lambda|$, satisfies $\rho(E^-A) < 1$. So, the series converges, and the solution X is symmetric positive semi-definite, i.e., $X \geq 0$. This series representation of the solution implies that the numerical rank of X is much smaller than its dimension n if the norm of the powers of E^-A decreases rapidly. In [39], Benner, Khoury, and Sadkane considered the solution of the Stein equation with $E = I_n$ and obtained an inequality that explicitly describes the decay of the singular values of the solution. For the projected Stein Equation (4), by following [39], we can obtain

$$\frac{\sigma_{jm+1}}{\sigma_1} \leq \|(E^-A)^j\|^2,$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ denote the singular values of the solution X. This result explicitly shows that the solution has a low numerical rank if the norm of the powers of E^-A decreases rapidly.

We can now apply the Smith method [24,28] to (4). It is a fixed-point iteration and is expressed as

$$X_{j+1} = (E^-A)X_j(E^-A)^T + E^-B(E^-B)^T, \quad X_0 = 0.$$

This iteration converges since $\rho(E^-A) < 0$, and the iterations can be written as the partial sum

$$X_j = \sum_{i=0}^{j-1}(E^-A)^iE^-B((E^-A)^iE^-B)^T, \quad j = 1,2,\cdots. \tag{6}$$

We see from (6) that the iterations can be reformulated by a low-rank representation of Cholesky factors, i.e.,

$$X_j = Z_jZ_j^T$$

with $Z_j = [E^-B, (E^-A)E^-B, \cdots, (E^-A)^{j-1}E^-B]$. It follows from (5) and (6) that the error matrix $X - X_j$ can be expressed as

$$\begin{aligned} X - X_j &= \sum_{i=j}^{\infty}(E^-A)^iE^-B((E^-A)^iE^-B)^T \\ &= (E^-A)^j\left(\sum_{i=0}^{\infty}(E^-A)^iE^-B((E^-A)^iE^-B)^T\right)((E^-A)^j)^T \\ &= (E^-A)^jX((E^-A)^j)^T. \end{aligned}$$

Consequently, we can obtain the relative error bound

$$\frac{\|X - X_j\|}{\|X\|} \leq \|(E^-A)^j\|^2.$$

This shows that X_j converges linearly to the solution if the spectrum radius $\rho(E^-A) < 1$, and X can be accurately approximately by the low-rank iteration $X_j = Z_jZ_j^T$ if the norm of the powers of E^-A decreases rapidly. Note that the norm of the error matrix is not computable since the solution X is unknown. For large-scale problems, it is also difficult to accurately estimate the relative error bound $\|(E^-A)^j\|^2$.

For the residual matrix R_j defined by

$$R_j = AX_jA^T + P_lBB^TP_l^T - EX_jE^T, \tag{7}$$

we have

$$\begin{aligned} R_j &= A\left(\sum_{i=0}^{j-1}(E^-A)^iE^-B((E^-A)^iE^-B)^T\right)A^T + P_lBB^TP_l^T \\ &\quad -E\left(\sum_{i=0}^{j-1}(E^-A)^iE^-B((E^-A)^iE^-B)^T\right)E^T \\ &= E\left(\sum_{i=1}^{j}(E^-A)^iE^-B((E^-A)^iE^-B)^T\right)E^T + E(E^-P_lB(E^-P_lB)^T)E^T \\ &\quad -E\left(\sum_{i=0}^{j-1}(E^-A)^iE^-B((E^-A)^iE^-B)^T\right)E^T \\ &= E((E^-A)^jE^-B((E^-A)^jE^-B)^T)E^T. \end{aligned}$$

So, the Frobenius matrix norm of R_j can be easily computed.

The dimension of the low-rank factor Z_j will increase by m in each iteration step. Hence, if the Smith iteration converges slowly, the number of columns of Z_j will easily reach unmanageable levels of memory requirements. To reduce the dimension of Z_j, we will approximate it by using the rank-revealing QR decomposition (RRQR) [40]. Assume that Z_j has the low numerical rank r_j with a prescribed tolerance τ. Consider the RRQR decomposition of Z_j with column pivoting:

$$Z_j^T \Pi_j = Q_j \Omega_j, \quad Q_j = [Q_j^{(1)}, Q_j^{(2)}], \quad \Omega_j = \begin{bmatrix} \Omega_j^{(1)} & \Omega_j^{(2)} \\ 0 & \Omega_j^{(3)} \end{bmatrix},$$

where Ω_j is an upper triangular matrix with $\Omega_j^{(1)} \in \mathbb{R}^{r_j \times r_j}$ and $\|\Omega_j^{(3)}\| < \tau$, Q_j is orthogonal, Π_j is a permutation matrix, and $Q_j^{(1)}$ has r_j columns. Then, $Z_j Z_j^T$ can be approximated by

$$Z_j Z_j^T \approx \Pi_j \begin{bmatrix} \Omega_j^{(1)} & \Omega_j^{(2)} \end{bmatrix}^T (Q_j^{(1)})^T Q_j^{(1)} \begin{bmatrix} \Omega_j^{(1)} & \Omega_j^{(2)} \end{bmatrix} \Pi_j = \widetilde{Z}_j \widetilde{Z}_j^T,$$

where

$$\widetilde{Z}_j = \Pi_j \Omega_j^T \begin{bmatrix} I_{r_j} \\ 0 \end{bmatrix}.$$

The low-rank Smith method for solving the projected Stein Equation (3) is presented in Algorithm 1.

Algorithm 1 Low-rank Smith method

Input: $E, A, B, \varepsilon, \tau$.

Output: Z such that ZZ^T is the approximate solution of (3)

1. Set $j = 1$, $V_1 = E^- B$, $Z_1 = V_1$;
2. Compute the rank-revealing QR decomposition

$$[Q_1, \Pi_1, \Omega_1, r_1] = \mathrm{RRQR}(Z_1^T, \tau).$$

3. Update Z_1 by

$$Z_1 = \Pi_1 \Omega_1^T \begin{bmatrix} I_{r_1} \\ 0 \end{bmatrix}.$$

4. While $\|(EV_j)^T (EV_j)\|_F > \varepsilon$ do
   - $j = j + 1$.
   - $V_j = E^- A V_{j-1}$.
   - $Z_j = [Z_{j-1}, V_j]$.
   - Compute the rank-revealing QR decomposition

$$[Q_j, \Pi_j, \Omega_j, r_j] = \mathrm{RRQR}(Z_j^T, \tau).$$

   - Update Z_j by

$$Z_j = \Pi_j \Omega_j^T \begin{bmatrix} I_{r_j} \\ 0 \end{bmatrix}.$$

   End While

The main advantage of the Smith iteration (6) is that it is very simple and can be easily implemented. However, we should note that the iterations converge very slowly if the spectrum radius $\rho(E^- A) \approx 1$. This is a significant motivation for further improvement of the Smith method.

3. Low-Rank ADI Method

The ADI method was first introduced in [41] and then applied to solve continuous-time Lyapunov matrix equations in [42]. Recently, this method was extended to the Stein Equation (3) by Benner and Faßbender [13] and further improved in [29]; see also [43].

For the projected Stein Equation (3), by generalizing the ADI method, we iteratively compute approximations X_j, $j \geq 1$, of the solution X by following the iteration scheme

$$\begin{aligned} (\overline{\mu}_j A - E) X_{j-1/2} A^T &= E X_{j-1} (\overline{\mu}_j E^T - A^T) - \overline{\mu}_j P_l B B^T P_l^T, & (8) \\ E X_j (E^T - \mu_j A^T) &= (A - \mu_j E) X_{j-1/2} A^T + P_l B B^T P_l^T, & (9) \end{aligned}$$

where $0 < |\mu_j| < 1$ denotes suitable shift parameters. Note that, although the iteration can work with any initial guess X_0, we use only $X_0 = 0$ in the sequel.

Since the pencil $\lambda E - A$ is d-stable and $0 < |\mu_j| < 1$, the matrices $\overline{\mu}_j A - E$ and $E^T - \mu_j A^T$ are nonsingular. From (8), we obtain

$$X_{j-1/2} A^T = (\overline{\mu}_j A - E)^{-1} E X_{j-1} (\overline{\mu}_j E^T - A^T) - \overline{\mu}_j (\overline{\mu}_j A - E)^{-1} P_l B B^T P_l^T, \quad (10)$$

Then, these half steps in the ADI iteration are rewritten into single steps by substituting $X_{j-1/2} A^T$ into (9) by the expression (10) for $X_{j-1/2} A^T$; i.e., we arrive at the single-step iteration

$$\begin{aligned} E X_j &= (A - \mu_j E) X_{j-1/2} A^T (E - \mu_j A)^{-T} + P_l B B^T P_l^T (E - \mu_j A)^{-T} \\ &= (A - \mu_j E)(\overline{\mu}_j A - E)^{-1} E X_{j-1} (\overline{\mu}_j E^T - A^T)(E - \mu_j A)^{-T} \\ &\quad - \overline{\mu}_j (A - \mu_j E)(\overline{\mu}_j A - E)^{-1} P_l B B^T P_l^T (E - \mu_j A)^{-T} \\ &\quad + P_l B B^T P_l^T (E - \mu_j A)^{-T}. \end{aligned} \quad (11)$$

Observe that

$$\begin{aligned} & -\overline{\mu}_j (A - \mu_j E)(\overline{\mu}_j A - E)^{-1} P_l B B^T P_l^T (E - \mu_j A)^{-T} + P_l B B^T P_l^T (E - \mu_j A)^{-T} \\ = & \left(-\overline{\mu}_j (A - \mu_j E)(\overline{\mu}_j A - E)^{-1} + I \right) P_l B B^T P_l^T (E - \mu_j A)^{-T} \\ = & \left(-\overline{\mu}_j (A - \mu_j E) + (\overline{\mu}_j A - E) \right) (\overline{\mu}_j A - E)^{-1} P_l B B^T P_l^T (E - \mu_j A)^{-T} \\ = & \left(1 - |\mu_j|^2 \right) E (\overline{\mu}_j A - E)^{-1} P_l B B^T P_l^T (\mu_j A - E)^{-T}. \end{aligned} \quad (12)$$

Hence, we obtain

$$\begin{aligned} E X_j &= (A - \mu_j E)(\overline{\mu}_j A - E)^{-1} E X_{j-1} (A^T - \overline{\mu}_j E^T)(\mu_j A - E)^{-T} \\ &\quad + \left(1 - |\mu_j|^2 \right) E (\overline{\mu}_j A - E)^{-1} P_l B B^T P_l^T (\mu_j A - E)^{-T}. \end{aligned} \quad (13)$$

Multiplying (13) from the left by E^- and using $X_j = P_r X_j P_r^T$,

$$\begin{aligned} P_r (\mu A - E)^{-1} &= (\mu A - E)^{-1} P_l, \\ P_l (A - \mu E) &= (A - \mu E) P_r \end{aligned}$$

for any μ, we obtain

$$\begin{aligned} X_j &= (\overline{\mu}_j A - E)^{-1} (A - \mu_j E) X_{j-1} (A^T - \overline{\mu}_j E^T)(\mu_j A - E)^{-T} \\ &\quad + \left(1 - |\mu_j|^2 \right) (\overline{\mu}_j A - E)^{-1} P_l B B^T P_l^T (\mu_j A - E)^{-T}. \end{aligned} \quad (14)$$

One can easily verify that the solution X of the projected Stein Equation (3) is a fixed point of the single-step iteration (14). That is to say,

$$\begin{aligned} X &= (\overline{\mu}_j A - E)^{-1}(A - \mu_j E)X(A^T - \overline{\mu}_j E^T)(\mu_j A - E)^{-T} \\ &\quad + \left(1 - |\mu_j|^2\right)(\overline{\mu}_j A - E)^{-1} P_l B B^T P_l^T (\mu_j A - E)^{-T}. \end{aligned} \tag{15}$$

Consequently, from (14) and (15), we obtain the following recursive formulation for the error matrix between the solution X and the approximation X_j:

$$X - X_j = (\overline{\mu}_j A - E)^{-1}(A - \mu_j E)(X - X_{j-1})(A^T - \overline{\mu}_j E^T)(\mu_j A - E)^{-T}, \quad j \geq 1. \tag{16}$$

We see from (16) and $X_0 = 0$ that the error matrix $X - X_j$ can be written as

$$X - X_j = \left(\prod_{i=1}^{j}(\overline{\mu}_i A - E)^{-1}(A - \mu_i E)\right) X \left(\prod_{i=1}^{j}(\overline{\mu}_i A - E)^{-1}(A - \mu_i E)\right)^H. \tag{17}$$

3.1. Low-Rank Version of ADI Method

For continuous-time Lyapunov equations with a low-rank right-hand side, Li and White [25] proposed the state-of-the-art Cholesky factor ADI algorithm, which generates a low-rank approximation to the solution. This method is a significant improvement of the ADI method [42] and is very appropriate for large-scale continuous-time Lyapunov equations. The Cholesky factor ADI method is developed by exploiting the low-rank structure of the iterations and reordering the shifts; see [25] for the details. The low-rank ADI method is generalized to the Stein equation in [29].

In this section, we follow these ideas to deduce the low-rank ADI method for the projected Stein Equation (3). Suppose now that X_j and X_{j-1} are written in their factored forms: $X_j = Z_j Z_j^H$ and $X_{j-1} = Z_{j-1} Z_{j-1}^H$. Then, from (14), we obtain the following recursive relation for Z_j and Z_{j-1}:

$$Z_j = \left[\sqrt{1 - |\mu_j|^2}(\overline{\mu}_j A - E)^{-1} P_l B \quad (\overline{\mu}_j A - E)^{-1}(A - \mu_j E) Z_{j-1} \right] \tag{18}$$

with $Z_0 = 0$.

From (18), we easily see that the dimension of Z_j would increase by m at each iteration step. Therefore, the factor Z_j of X_j has mj columns. Since the number of columns increases by m at each iteration, the number of systems of linear equations with matrices $\overline{\mu}A - E$, which need to be solved at each iteration in the low-rank ADI method (18), increases by m. So, this iteration (18) for the factors is not suitable for practical implementation. By making use of the trick in [25], Z_j can be reformulated as

$$\begin{aligned} Z_j &= \Big[\sqrt{1 - |\mu_1|^2} T_1 P_l B, \sqrt{1 - |\mu_2|^2} T_2 S_1 T_1 P_l B, \\ &\quad \cdots, \sqrt{1 - |\mu_j|^2} T_j S_{j-1} T_{j-1} \cdots S_2 T_2 S_1 T_1 P_l B \Big], \end{aligned}$$

where

$$T_j = (\overline{\mu}_j A - E)^{-1}, \quad S_j = (A - \mu_j E).$$

This directly leads to an efficient low-rank ADI iteration scheme: Let $V_1 = (\overline{\mu}_1 A - E)^{-1} P_l B$, and $Z_1 = \sqrt{1 - |\mu_1|^2} V_1$. Then, for $j \geq 1$,

$$\widetilde{V}_j = (A - \mu_j E) V_j, \tag{19}$$

$$V_{j+1} = (\overline{\mu}_{j+1} A - E)^{-1} \widetilde{V}_j, \tag{20}$$

$$Z_{j+1} = [Z_j, \sqrt{1 - |\mu_{j+1}|^2} V_{j+1}]. \tag{21}$$

We now investigate the residual matrix R_j corresponding to the j-th approximate solution X_j. In the following theorem, we show that R_j has a low-rank factorization of rank at most m.

Theorem 1. *Let $X_j = Z_j Z_j^H$, where Z_j is the j-th iteration generated by the low-rank ADI for the projected Stein Equation (3), and let the $n \times m$ matrix $\widetilde{V}_j$ be defined by (19). Then, the residual matrix R_j, defined by (7), can be formulated as*

$$R_j = \widetilde{V}_j \widetilde{V}_j^H.$$

Proof. From (3), $P_l B B^T P_l^T = EXE^T - AXA^T$. Thus,

$$R_j = E(X - X_j)E^T - A(X - X_j)A^T. \tag{22}$$

By inserting (17) into (22), we obtain

$$\begin{aligned} R_j &= E\left(\prod_{i=1}^{j}(\overline{\mu}_i A - E)^{-1}(A - \mu_i E)\right) X \left(\prod_{i=1}^{j}(\overline{\mu}_i A - E)^{-1}(A - \mu_i E)\right)^H E^T \\ &\quad - A\left(\prod_{i=1}^{j}(\overline{\mu}_i A - E)^{-1}(A - \mu_i E)\right) X \left(\prod_{i=1}^{j}(\overline{\mu}_i A - E)^{-1}(A - \mu_i E)\right)^H A^T \\ &= \left(\prod_{i=1}^{j}(\overline{\mu}_i A - E)^{-1}(A - \mu_i E)\right) P_l B B^T P_l^T \left(\prod_{i=1}^{j}(\overline{\mu}_i A - E)^{-1}(A - \mu_i E)\right)^H. \end{aligned}$$

From (19) and (20), and $V_1 = (\overline{\mu}_1 A - E)^{-1} P_l B$, it follows that

$$\widetilde{V}_j = \left(\prod_{i=1}^{j}(\overline{\mu}_i A - E)^{-1}(A - \mu_i E)\right) P_l B.$$

Thus, $R_j = \widetilde{V}_j \widetilde{V}_j^H$. □

The following theorem states that V_{j+1} and $\widetilde{V}_{j+1}$ can be obtained without solving systems of linear equations once V_j has been computed.

Theorem 2. *Assume that a proper set of shift parameters is used in the low-rank ADI iteration. For the two subsequent blocks V_{j+1} and $\widetilde{V}_{j+1}$ related to the pair of complex shifts $\{\mu_j, \mu_{j+1}\}$ with $\mu_{j+1} = \overline{\mu}_j$, it holds that*

$$V_{j+1} = \mu_j \mathrm{Re}(V_j) + \frac{1}{\overline{\mu}_j}\left((1 - |\mu_j|^2)\frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)} - \jmath\right)\mathrm{Im}(V_j), \tag{23}$$

$$\widetilde{V}_{j+1} = \frac{1}{|\mu_j|^2}\left(\widetilde{V}_{j-1} + (1 - |\mu_j|^4)E\mathrm{Re}(V_j) + (1 - |\mu_j|^2)^2 \frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)} E\mathrm{Im}(V_j)\right). \tag{24}$$

Moreover, after applying a pair of complex shifts $\{\mu_j, \mu_{j+1}\}$, the generated $\widetilde{V}_{j+1}$ is real.

Proof. Although the proof is similar to that of Theorem 1 in [30], we include the proof for completeness. In [30], the proof is split into three cases concerning different possible (sub)sequences of shift parameters.

Here, we only consider the first case in [30] for illustration. Assume that $\mu_1, \cdots, \mu_{j-1}$ are real, μ_j has a nonzero imaginary part, and $\mu_{j+1} = \overline{\mu}_j$. In this case, obviously, from (19)

and (20), $V_1, \cdots, V_{j-1}$, $\widetilde{V}_1, \cdots, \widetilde{V}_{j-1}$ are real, and V_j is the first complex iteration. From (20), it follows that

$$(\overline{\mu}_j A - E)V_j = \widetilde{V}_{j-1}.$$

Now, splitting μ_j and V_j into their real and imaginary parts reveals

$$\begin{aligned}\widetilde{V}_{j-1} &= (\mathrm{Re}(\mu_j)A - E)\mathrm{Re}(V_j) + \mathrm{Im}(\mu_j)A\mathrm{Im}(V_j)\\ &\quad +\jmath\,[(\mathrm{Re}(\mu_j)A - E)\mathrm{Im}(V_j) - \mathrm{Im}(\mu_j)A\mathrm{Re}(V_j)].\end{aligned}$$

Since $\widetilde{V}_{j-1}$ is real, then

$$(\mathrm{Re}(\mu_j)A - E)\mathrm{Im}(V_j) - \mathrm{Im}(\mu_j)A\mathrm{Re}(V_j) = 0,$$

which leads to

$$A\mathrm{Re}(V_j) = \frac{1}{\mathrm{Im}(\mu_j)}(\mathrm{Re}(\mu_j)A - E)\mathrm{Im}(V_j), \tag{25}$$

$$\begin{aligned}EV_j &= E\mathrm{Re}(V_j) + \jmath E\mathrm{Im}(V_j)\\ &= (E - \jmath\mathrm{Im}(\mu_j)A)\mathrm{Re}(V_j) + \jmath\mathrm{Re}(\mu_j)A\mathrm{Im}(V_j).\end{aligned} \tag{26}$$

From (19) and (20), it follows that

$$\begin{aligned}\widetilde{V}_j &= (A - \mu_j E)V_j = (A - \mu_j E)(\overline{\mu}_j A - E)^{-1}\widetilde{V}_{j-1}\\ &= \frac{1}{\overline{\mu}_j}(\overline{\mu}_j A - E + E - |\mu_j|^2 E)(\overline{\mu}_j A - E)^{-1}\widetilde{V}_{j-1}\\ &= \frac{1}{\overline{\mu}_j}\widetilde{V}_{j-1} + \frac{1 - |\mu_j|^2}{\overline{\mu}_j}E(\overline{\mu}_j A - E)^{-1}\widetilde{V}_{j-1}\\ &= \frac{1}{\overline{\mu}_j}\widetilde{V}_{j-1} + \frac{1 - |\mu_j|^2}{\overline{\mu}_j}EV_j.\end{aligned} \tag{27}$$

From (20), we obtain

$$\begin{aligned}\overline{\mu}_j V_{j+1} &= \overline{\mu}_j(\overline{\mu}_{j+1}A - E)^{-1}\widetilde{V}_j\\ &= (\mu_j A - E)^{-1}\left(\widetilde{V}_{j-1} + (1 - |\mu_j|^2)EV_j\right)\\ &= \overline{V}_j + (1 - |\mu_j|^2)(\mu_j A - E)^{-1}EV_j\\ &= |\mu_j|^2\mathrm{Re}(V_j) - \jmath\mathrm{Im}(V_j)\\ &\quad +(1 - |\mu_j|^2)(\mu_j A - E)^{-1}((\mu_j A - E)\mathrm{Re}(V_j) + EV_j).\end{aligned} \tag{28}$$

From (25) and (26), we obtain

$$\begin{aligned}&(\mu_j A - E)\mathrm{Re}(V_j) + EV_j\\ =\ &(\mu_j A - E)\mathrm{Re}(V_j) + (E - \jmath\mathrm{Im}(\mu_j)A)\mathrm{Re}(V_j) + \jmath\mathrm{Re}(\mu_j)A\mathrm{Im}(V_j)\\ =\ &\mathrm{Re}(\mu_j)A\mathrm{Re}(V_j) + \jmath\mathrm{Re}(\mu_j)A\mathrm{Im}(V_j)\\ =\ &\frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)}(\mathrm{Re}(\mu_j)A - E)\mathrm{Im}(V_j) + \jmath\mathrm{Re}(\mu_j)A\mathrm{Im}(V_j)\\ =\ &\frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)}(\mu_j A - E)\mathrm{Im}(V_j).\end{aligned}$$

By inserting (29) into (28), we obtain

$$\overline{\mu}_j V_{j+1} = |\mu_j|^2 \mathrm{Re}(V_j) + \left((1-|\mu_j|^2)\frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)} - \jmath\right)\mathrm{Im}(V_j),$$

which leads to (23).

From (27) and $\mu_{j+1} = \overline{\mu}_j$, it follows that

$$\begin{aligned}
\widetilde{V}_{j+1} &= \frac{1}{\mu_j}\widetilde{V}_j + \frac{1-|\mu_j|^2}{\mu_j} E V_{j+1} \\
&= \frac{1}{\mu_j}\left(\frac{1}{\overline{\mu}_j}\widetilde{V}_{j-1} + \frac{1-|\mu_j|^2}{\overline{\mu}_j} E V_j\right) \\
&\quad + \frac{1-|\mu_j|^2}{\mu_j} E\left(\mu_j \mathrm{Re}(V_j) + \frac{1}{\overline{\mu}_j}\left((1-|\mu_j|^2)\frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)} - \jmath\right)\mathrm{Im}(V_j)\right). \\
&= \frac{1}{|\mu_j|^2}\widetilde{V}_{j-1} + \frac{1-|\mu_j|^2}{|\mu_j|^2} E V_j + (1-|\mu_j|^2) E \mathrm{Re}(V_j) \\
&\quad - \jmath\frac{1-|\mu_j|^2}{|\mu_j|^2} E \mathrm{Im}(V_j) + \frac{(1-|\mu_j|^2)^2}{|\mu_j|^2}\frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)} E \mathrm{Im}(V_j) \\
&= \frac{1}{|\mu_j|^2}\left(\widetilde{V}_{j-1} + (1-|\mu_j|^4) E \mathrm{Re}(V_j) + (1-|\mu_j|^2)^2 \frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)} E \mathrm{Im}(V_j)\right).
\end{aligned}$$

Obviously, $\widetilde{V}_{j+1}$ is real. □

3.2. Dealing with Complex Shifts

From $V_j = (\overline{\mu}_j A - E)^{-1}\widetilde{V}_{j-1}$, it follows that if μ_j is a complex shift, then the complex V_j will be added to the low-rank factor Z_j. In this case, complex arithmetic operations and storage are introduced into the process such that a complex low-rank factor Z_j is generated in the end. From the numerical point of view, it is undesirable to use complex arithmetic operations in the iteration since operations and storage will increase.

Recently, some research has focused on how to deal with complex shift parameters in the Cholesky factor ADI method for continuous-time Lyapunov equations. In [24,25,44], a completely real formulation of the Cholesky factor ADI method is presented by concatenating steps associated with a pair of complex conjugate shift parameters into one step. Although this reformulation has the advantage that complex arithmetic operations and storage are avoided, systems of linear equations with matrices of the form $A^2 + 2\mathrm{Re}(\mu_j)A + |\mu_j|^2 I_n$ need to be solved in every two steps of the ADI method. This is a major drawback for the completely real formulation. Firstly, for large-scale problems, $A^2 + 2\mathrm{Re}(\mu_j)A + |\mu_j|^2 I_n$ may not preserve the original sparsity of A, and thus, sparse direct solvers cannot be applied to linear systems with such coefficient matrices. Secondly, from the perspective of numerical stability, it is undesirable to solve such linear equations since the condition number can be increased due to squaring. Iterative solvers such as Krylov subspace methods [45,46] can still be applied to linear systems with such coefficient matrices since they work with matrix–vector products only. However, it is known that the large condition number will deteriorate the efficiency of iterative solvers. In order to overcome these disadvantages in the completely real formulation and to avoid complex arithmetic and the storage of complex matrices as much as possible, Benner, Kürschner, and Saak [30] introduced a partially real reformulation of the Cholesky factor ADI method for continuous-time Lyapunov equations. They exploit the fact that the ADI shifts need to occur as a real number or as a pair of complex conjugate numbers. As a result, the resulting low-rank ADI method works with real low-rank factors Z_j; see also [47]. This idea is extended to the Stein equation in [29].

We now consider the generalization to obtain a partially real version of the low-rank ADI method for the projected Stein Equation (3) by investigating the blocks V_j, V_{j+1}, which are generated in the low-rank ADI with a pair of complex conjugate shifts $\mu_j, \mu_{j+1} = \overline{\mu}_j$.

Define

$$\begin{aligned} \widehat{Z} &= \left[\begin{array}{cc} \sqrt{1-|\mu_j|^2}V_j & \sqrt{1-|\mu_{j+1}|^2}V_{j+1} \end{array} \right], \\ \widetilde{Z} &= \left[\begin{array}{cc} \mathrm{Re}(V_j) & \mathrm{Im}(V_j) \end{array} \right]. \end{aligned}$$

From (19) and (23), with the help of the Kronecker product and $\mu_{j+1} = \overline{\mu}_j$, we obtain

$$\widehat{Z} = \widetilde{Z}(\Xi \otimes I_m), \tag{29}$$

where

$$\Xi = \sqrt{1-|\mu_j|^2}\left[\begin{array}{cc} 1 & \mu_j \\ \jmath & \frac{1}{\overline{\mu}_j}\left((1-|\mu_j|^2)\frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)} - \jmath\right) \end{array} \right].$$

Direct calculation reveals

$$\Xi\Xi^H = (1-|\mu_j|^2)\left[\begin{array}{cc} 1+|\mu_j|^2 & (1-|\mu_j|^2)\frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)} \\ (1-|\mu_j|^2)\frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)} & 1+\frac{1}{|\mu_j|^2}\left((1-|\mu_j|^2)^2\frac{(\mathrm{Re}(\mu_j))^2}{(\mathrm{Im}(\mu_j))^2}+1\right) \end{array} \right].$$

The real symmetric positive definite matrix $\Xi\Xi^H$ has a unique Cholesky factorization given by

$$\Xi\Xi^H = LL^T, \quad L = \left[\begin{array}{cc} l_1 & 0 \\ l_2 & l_3 \end{array} \right], \tag{30}$$

where

$$\begin{aligned} l_1 &= \sqrt{1-|\mu_j|^4}, \\ l_2 &= l_1^{-1}(1-|\mu_j|^2)^2\frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)}, \\ l_3 &= \sqrt{(1-|\mu_j|^2)\left(1+\frac{1}{|\mu_j|^2}\left((1-|\mu_j|^2)^2\frac{(\mathrm{Re}(\mu_j))^2}{(\mathrm{Im}(\mu_j))^2}+1\right)\right) - l_2^2}. \end{aligned} \tag{31}$$

From (29) and the Cholesky factorization (30) of $\Pi\Pi^H$, we obtain a real low-rank expression for $\widehat{Z}\widehat{Z}^H$:

$$\widehat{Z}\widehat{Z}^H = \check{Z}\check{Z}^T$$

with

$$\check{Z} = \left[\begin{array}{cc} l_1\mathrm{Re}(V_j) + l_2\mathrm{Im}(V_j) & l_3\mathrm{Im}(V_j) \end{array} \right].$$

In summary, we can propose a partially real low-rank ADI method for solving the projected Stein Equation (3), which is presented in Algorithm 2.

Algorithm 2 Real low-rank ADI method

Input: $E, A, B, \varepsilon, \mu_j$ with $0 < |\mu_j| < 1$.

Output: Z such that ZZ^T is the approximate solution of (3)

1. Set $j = 1$, $\widetilde{V}_0 = P_l B$, $Z = []$;
2. While $\|\widetilde{V}_{j-1}^T \widetilde{V}_{j-1}\|_F > \varepsilon$ do
   - Solve $V_j = (\overline{\mu}_j A - E)^{-1} \widetilde{V}_{j-1}$ for V_j.
   - If $\mathrm{Im}(\mu_j) = 0$ then
     - $Z = [Z, \sqrt{1 - |\mu_j|^2} V_j]$.
     - $\widetilde{V}_j = \frac{1}{\mu_j}\left(\widetilde{V}_{j-1} + (1 - |\mu_j|^2) E V_j\right)$.
   - else
     - Compute l_1, l_2, l_3 according to (31).
     - Set $Z = [Z, l_1 \mathrm{Re}(V_j) + l_2 \mathrm{Im}(V_j), l_3 \mathrm{Im}(V_j)]$.
     - $\widetilde{V}_{j+1} = \frac{1}{|\mu_j|^2}\left(\widetilde{V}_{j-1} + (1 - |\mu_j|^4) E \mathrm{Re}(V_j) + (1 - |\mu_j|^2)^2 \frac{\mathrm{Re}(\mu_j)}{\mathrm{Im}(\mu_j)} E \mathrm{Im}(V_j)\right)$.
     - $j = j + 1$.
   - End If
   - $j = j + 1$.

   End While

3.3. Choosing the ADI Shift Parameters

We now consider how to compute appropriate shift parameters. These shifts are vitally important to the convergence rate of the ADI iteration.

For the ADI method for the projected Stein equation, the parameters $\{\mu_j\}_{j=1}^k$ should be chosen according to the minimax problem

$$\min_{\substack{0 < |\mu_j| < 1, \\ j = 1, 2, \ldots, k}} \max_{t \in \Lambda_f} \prod_{j=1}^{k} \left| \frac{t - \mu_j}{\overline{\mu}_j t - 1} \right|, \tag{32}$$

where Λ_f denotes the set of finite eigenvalues of the pencil $\lambda E - A$. In practice, since the eigenvalues of the pencil $\lambda E - A$ are unknown and computationally expensive, Λ_f is usually replaced by a domain containing a finite set of eigenvalues of $\lambda E - A$.

A heuristic algorithm [21,48] can calculate the suboptimal ADI shift parameters for standard Lyapunov or Sylvester equations. It selects suboptimal ADI parameters from a set Ω, which is taken as the union of Ritz values of A and the reciprocals of the Ritz values of A^{-1}, obtained by two Arnoldi processes, with A and A^{-1}.

The heuristic algorithm can also be naturally extended to the minimax problem (32). Since E is assumed to be singular, the inverse of E does not exist. However, it is clear that $E^- A$ has the same nonzero finite eigenvalues as the pencil $\lambda E - A$. Moreover, the reciprocals of the largest nonzero eigenvalues of $A^{-1}E$ are the smallest eigenvalues of $E^- A$. Thus, we can run one Arnoldi process with the matrix $A^- E$ to compute the smallest nonzero eigenvalues of $E^- A$.

The algorithm for computing $\{\mu_j\}_{j=1}^k$ is summarized in Algorithm 3. For more details about the implementation of this algorithm, the interested reader is referred to [24,28,48].

Algorithm 3 Choose ADI parameters

Input: $E, A \in \mathbb{R}^{n\times n}$ with $\lambda E - A$ being d-stable, $b \in \mathbb{R}^n$, k_+, k_-.

Output: ADI parameters $\mathbb{P}$.

1. Run k_+ steps of the Arnoldi process with respect to E^-A on b to obtain the set Ω_+ of Ritz values.
2. Run k_- steps of the Arnoldi process with respect to $A^{-1}E$ on b to obtain the set Ω_- of Ritz values.
3. Set $\Omega = \Omega_+ \cup (1/\Omega_-)$.
4. Set
$$\mu_1 = \arg \min_{\mu \in \Omega} \max_{t \in \Omega} \left| \frac{t-\mu}{\overline{\mu}t - 1} \right|.$$
5. If $Im(\mu_1) = 0$, $\mathbb{P} = \mathbb{P} \cup \{\mu_1\}$, $j = 1$; else $\mathbb{P} = \mathbb{P} \cup \{\mu_1, \mu_2 = \overline{\mu}_1\}$, $j = 2$.
6. While $j < k$ do
   - Set
   $$\mu_{j+1} = \arg \min_{\mu \in \Omega'} \max_{t \in \Omega} \left| \frac{t-\mu}{\overline{\mu}t - 1} \right| \prod_{i=1}^{j} \left| \frac{t-\mu_i}{\overline{\mu}_i t - 1} \right|,$$
   where Ω' is Ω with $\mathbb{P}$ deleted.
   - If $Im(\mu_{j+1}) = 0$, $\mathbb{P} = \mathbb{P} \cup \{\mu_{j+1}\}$, $j = j+1$; else $\mathbb{P} = \mathbb{P} \cup \{\mu_{j+1}, \mu_{j+2} = \overline{\mu}_{j+1}\}$, $j = j+2$.

   End While

4. Numerical Examples

We provide two numerical examples to demonstrate the convergence performance of the LR-ADI method and the LR-Smith method for (3) in this section. Define the relative residual (RRes) as

$$\text{RRes} \equiv \frac{\|AX_jA^T + P_lBB^TP_l^T - EX_jE^T\|_F}{\|P_lBB^TP_l^T\|_F},$$

where X_j is generated by LR-Smith or LR-ADI.

For LR-ADI, we first compute $k = 20$ shift parameters by making use of Algorithm 3 and then reuse these parameters in a circular manner if the number of shift parameters is less than the number of iterations required to achieve the specified tolerance. In LR-ADI, we solve the shift linear systems by the *LU* factorization of the corresponding coefficient matrices.

All the numerical results are obtained by performing calculations on an Intel Core i7-8650U with CPU 1.90 GHz and RAM 16 GB.

4.1. Example 1

In this example, the differential algebraic equation (DAE) is

$$\begin{cases} \widehat{E}_{11}\dot{x}(t) = \widehat{A}_{11}x(t) + \widehat{A}_{12}p(t) + \widehat{B}_1u(t), \\ 0 = \widehat{A}_{21}x(t) + \widehat{B}_2u(t) \end{cases} \tag{33}$$

with $\widehat{E}_{11} = I$, $\widehat{A}_{11} = \widehat{A}_{11}^T$, and $\widehat{A}_{21} = \widehat{A}_{12}^T$. It comes from the spatial discretization of the 2D instationary Stokes equation [37].

In order to obtain discrete-time equations, we first use a semi-explicit Euler and a semi-implicit Euler method [49] with a timestep size Δt to discretize the differential equation of (33). This leads to two difference equations:

$$\widehat{E}_{11}x_{k+1} = (\widehat{E}_{11} + \Delta t\widehat{A}_{11})x_k + \Delta t\widehat{A}_{12}p_k + \Delta t\widehat{B}_1u_k, \tag{34}$$

$$(\widehat{E}_{11} - \Delta t\widehat{A}_{11})x_{k+1} = \widehat{E}_{11}x_k + \Delta t\widehat{A}_{12}p_k + \Delta t\widehat{B}_1u_k. \tag{35}$$

Then, by averaging (34) and (35) and also discretizing the algebraic equation of (33), we obtain the final difference-algebraic equations

$$\underbrace{\begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix}}_{E}\begin{bmatrix} x_{k+1} \\ p_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & 0 \end{bmatrix}}_{A}\begin{bmatrix} x_k \\ p_k \end{bmatrix} + Bu_k, \tag{36}$$

where $E_{11} = \widehat{E}_{11} - \frac{\Delta t}{2}\widehat{A}_{11}$, $A_{11} = \widehat{E}_{11} + \frac{\Delta t}{2}\widehat{A}_{11}$, $A_{12} = A_{21}^T = \Delta t\widehat{A}_{12}$, and $B = \Delta t[\widehat{B}_1^T, \widehat{B}_2^T]^T$.

Note that E, A in (36) are sparse and have special block structures. Using these structures, the projectors P_l and P_r can be formulated as

$$P_l = \begin{bmatrix} \Pi_l & -\Pi_l A_{11}\Psi \\ 0 & 0 \end{bmatrix}, \tag{37}$$

$$P_r = \begin{bmatrix} \Pi_r & 0 \\ -\Phi A_{11}\Pi_r & 0 \end{bmatrix}, \tag{38}$$

where

$$\begin{aligned} \Pi_l &= I - A_{12}(A_{21}E_{11}^{-1}A_{12})^{-1}A_{21}E_{11}^{-1} \\ \Pi_r &= I - E_{11}^{-1}A_{12}(A_{21}E_{11}^{-1}A_{12})^{-1}A_{21} = E_{11}^{-1}\Pi_l E_{11}, \\ \Phi &= (A_{21}E_{11}^{-1}A_{12})^{-1}A_{21}E_{11}^{-1}, \\ \Psi &= E_{11}^{-1}A_{12}(A_{21}E_{11}^{-1}A_{12})^{-1}. \end{aligned}$$

Moreover,

$$\begin{aligned} E^- &= (P_lE + (I - P_l)A)^{-1}P_l = \begin{bmatrix} \Pi_r E_{11}^{-1} & -\Pi_r E_{11}^{-1}A_{11}\Psi \\ -\Phi A_{11}\Pi_r E_{11}^{-1} & \Phi A_{11}\Pi_r E_{11}^{-1}A_{11}\Psi \end{bmatrix}, \\ E^-A &= \begin{bmatrix} \Pi_r E_{11}^{-1}A_{11}\Pi_r & 0 \\ -\Phi A_{11}\Pi_r E_{11}^{-1}A_{11}\Pi_r & 0 \end{bmatrix}, \end{aligned}$$

see, for example, refs. [28,37] for details.

In this example, the timestep size Δt is taken to be 0.05, and $[\widehat{B}_1^T, \widehat{B}_2^T]^T \in \mathbb{R}^{n\times 2}$ is a matrix with each element being 1, except the $(1,2)$-th element, which is 0.

We first test a medium-size problem of order $n = 1280$. Figure 1 illustrates the sparsity structure of the matrix $A_{21}E_{11}^{-1}A_{12}$. This may show that for larger problems, it is expensive to compute the LU factorization of $A_{21}E_{11}^{-1}A_{12}$. For this experiment, as well as for the larger problems later, the final relative residual accuracy for both the LR-ADI method and the LR-Smith method was set to 10^{-8}. The convergence curves of LR-ADI and LR-Smith are depicted in Figure 2. The ADI method reaches a relative residual of 6.2472×10^{-9} after 13 steps of iteration, while the Smith method has a relative residual of 9.1324×10^{-9} after 75 steps. From Figure 2, it is clear that LR-Smith is much slower than LR-ADI with respect to the number of iterations. From Section 2, we know that the convergence factor of LR-Smith is the spectrum radius $\rho(E^-A)$. If the spectrum radius $\rho(E^-A) \approx 1$, LR-Smith will converge very slowly. Note that the spectrum radius $\rho(E^-A)$ is the absolute value of the largest finite eigenvalues (in modulus) of the pencil $\lambda E - A$. For this medium-size problem, the largest finite eigenvalue is 0.9554, which verifies the slow convergence of LR-Smith. However, from Table 1, we can see that LR-Smith is only

slightly more expensive, with respect to execution time, than LR-ADI. The reason is that linear systems with coefficient matrices A and $\overline{\mu}A - E$ must be solved in the latter method.

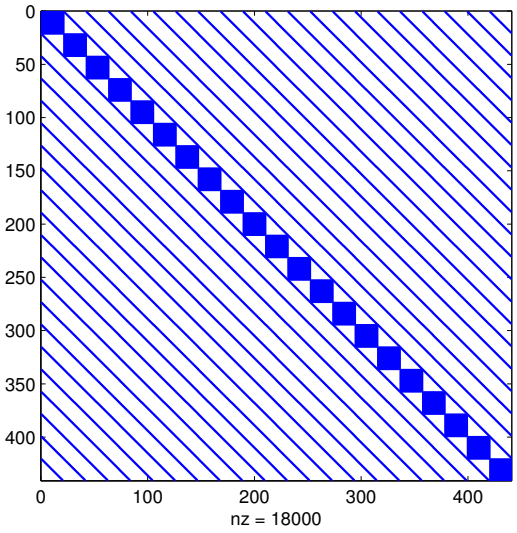

Figure 1. Example 1. Sparsity structure of the matrix $A_{21}E_{11}^{-1}A_{12}$.

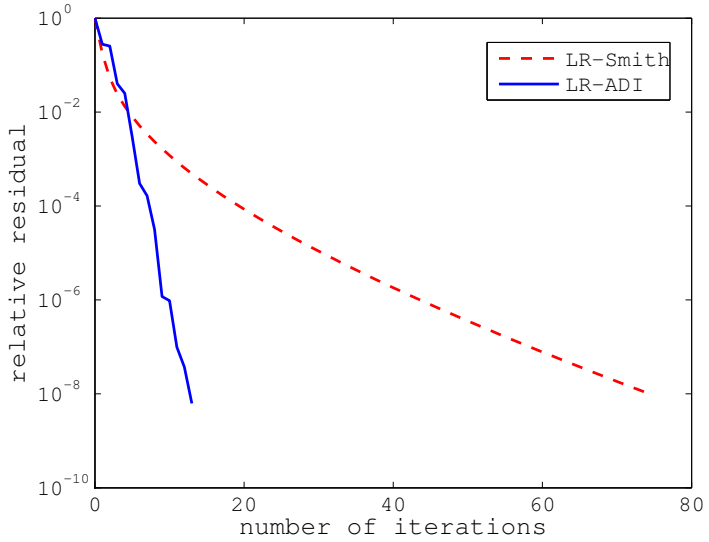

Figure 2. Example 1 with $n = 1280$.

We also tested problems with larger dimensions, namely, $n = 3604$, $n = 7700$, and $n = 14{,}559$. All the numerical results are reported in Table 1. As expected, the execution time becomes increasingly larger for both LR-Smith and LR-ADI as the problem dimension expands. Moreover, it is clear that both the number of iterations and the cpu time of LR-ADI are better than those of LR-Smith. However, it is interesting that the number of iterations for LR-Smith increases much more dramatically than for LR-ADI. This is the reason that LR-ADI is more favorable with respect to the cpu time for problems with larger dimensions. For illustration, we also present the convergence curves of the two methods for dimensions $n = 7700$ and $n = 14{,}559$, respectively, in Figures 3 and 4.

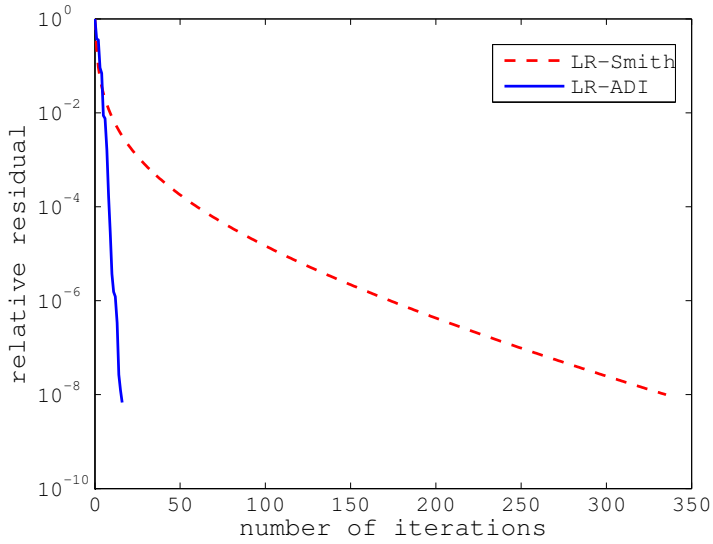

Figure 3. Example 1 with $n = 7700$.

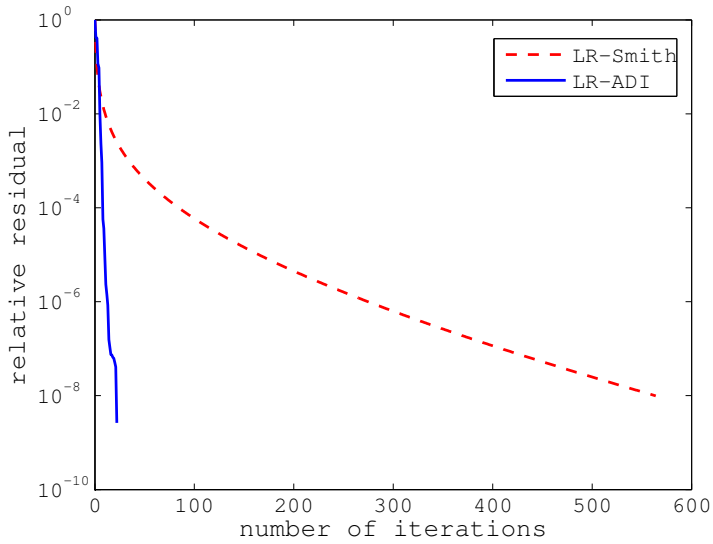

Figure 4. Example 1 with n = 14,559.

Table 1. Numerical results for Example 1.

n	Method	Iter	Cpu Time	RRes
1280	LR-ADI	13	0.27	6.2472×10^{-9}
	LR-Smith	75	0.34	9.1324×10^{-9}
3604	LR-ADI	13	2.1	5.8859×10^{-9}
	LR-Smith	179	5.0	9.6261×10^{-9}
7700	LR-ADI	16	14.8	6.8133×10^{-9}
	LR-Smith	335	40.8	9.9569×10^{-9}
14,559	LR-ADI	22	86.8	2.6304×10^{-9}
	LR-Smith	564	227.9	9.9038×10^{-9}

4.2. Example 2

In the second example, the DAE (33) is used to describe a holonomically constrained damped mass–spring system with g masses [28,37]. The system matrices have the following structures:

$$\widehat{E}_{11} = \begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix}, \qquad \widehat{A}_{11} = \begin{bmatrix} 0 & I \\ K & D \end{bmatrix},$$
$$\widehat{A}_{12} = \begin{bmatrix} 0 \\ -N^T \end{bmatrix}, \qquad \widehat{A}_{21} = \begin{bmatrix} N & 0 \end{bmatrix}.$$

Similarly, we discretize the DAE by the same method as in the first example to obtain difference-algebraic equations, in which the matrices E, A have the same block structures as in the first example.

By setting $g = 2000, 6000, 10{,}000$, we obtain three DAEs of order $n = 2g + 1 = 4001$, 12,001, 200,001. The timestep size Δt is taken to be 0.1. All the elements of $[\widehat{B}_1^T, \widehat{B}_2^T]^T \in \mathbb{R}^{n\times 1}$ are set to 0, except the $(g+1)$-th element, which is 1.

In Figure 5, we first illustrate the convergence curves of LR-Smith and LR-ADI for the problem with the dimension $n = 4001$. Oddly enough, we can obviously see that the relative residual of LR-Smith does not monotonously decrease. This is due to the rounding error, which can make the relative residual strangely increase if the convergence factor is almost equal to 1. LR-Smith did not converge even if it had run 1000 steps, which cost 45.6 s. However, LR-ADI, with 21 steps of iteration, had a relative residual of 1.2886×10^{-9} and a cpu time of 1.8 s.

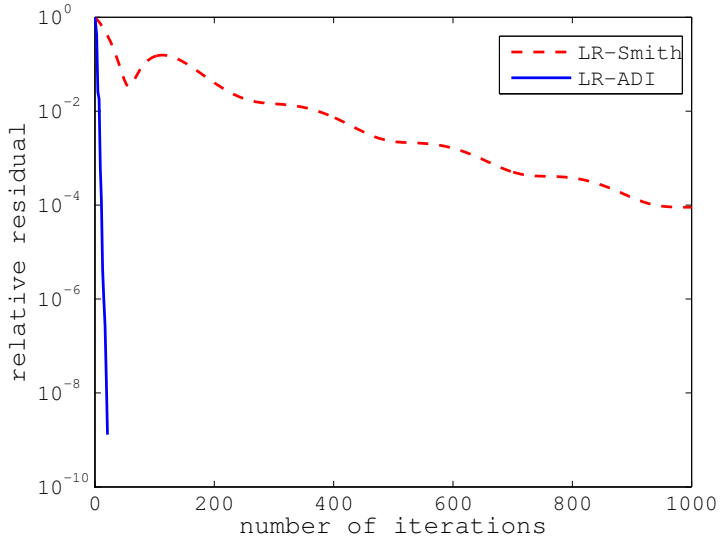

Figure 5. Example 2 with $n = 4001$.

Since LR-Smith does not converge for this example, we only report our numerical results of LR-ADI for all problems of different dimensions ($n = 4001$, 12,001, 20,001) in Table 2. We observe that LR-ADI is very fast for this example and, in particular, that the number of iterations does not increase considerably with the expansion of the problem dimension. Moreover, we also notice that the execution time for this example is much less than that for the first example. The reason is that, in the second example, $A_{21}E_{11}^{-1}A_{12}$ is a number, and E, A are sparser than those in the first one, which makes it cheaper to compute the LU factorization of $A_{21}E_{11}^{-1}A_{12}$, A, and $\overline{\mu}A - E$ in this example. Finally, we show the convergence curve of LR-ADI for the problem with the dimension n = 20,001 in Figure 6.

Table 2. Numerical results for Example 2.

n	Method	Iter	Cpu Time	RRes
4001	LR-ADI	21	1.8	1.2886×10^{-9}
12,001	LR-ADI	22	5.3	3.0284×10^{-9}
20,001	LR-ADI	26	9.1	2.3404×10^{-9}

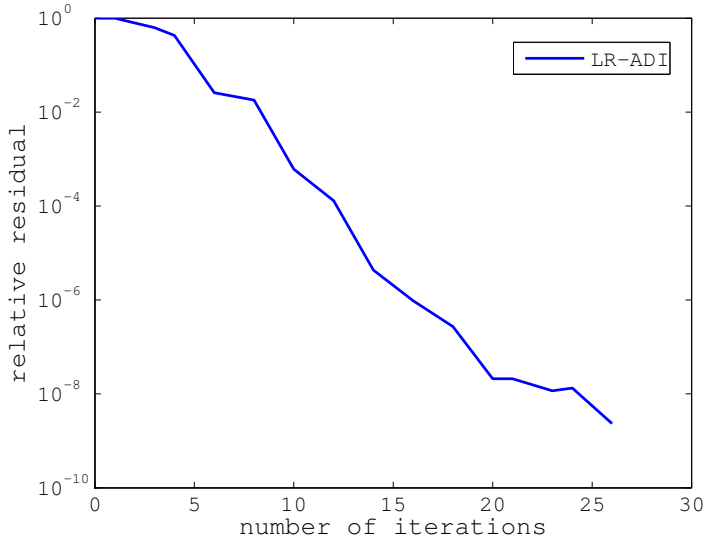

Figure 6. Example 2 with n = 20,001.

5. Conclusions

We have proposed a low-rank Smith method and a low-rank ADI method for the solutions of large-scale projected Stein equations. Although the Smith iteration and ADI iteration are very common in the field of efficient numerical solutions of linear matrix equations such as Lyapunov equations and Sylvester equations, this is the first time that they are adapted to numerically solve large-scale projected Stein equations, which arise in the balanced truncation model reduction of discrete-time descriptor systems. We also present a partially real version of the low-rank ADI method, as some of the shift parameters are imaginary. Our numerical experiments seem to show that the low-rank ADI method is more competitive than the low-rank Smith method with respect to the execution time and the number of iterations for large-scale problems if the convergence factor of the low-rank Smith method is large.

Funding: This research was funded by the Natural Science Foundation of Hunan Province under grant 2017JJ2102, the Academic Leader Training Plan of Hunan Province, and the Applied Characteristic Discipline at Hunan University of Science and Engineering.

Data Availability Statement: The author confirms that the data supporting the findings of this study are available within the article.

Acknowledgments: The author thanks the editors and anonymous referees for their helpful suggestions, which greatly improve the paper.

Conflicts of Interest: The author declares that there are no conflicts of interest.

References

1. Gantmacher, F. *Theory of Matrices*; Chelsea: New York, NY, USA, 1959.
2. Stykel, T. Analysis and Numerical Solution of Generalized Lyapunov Equations. Ph.D. Thesis, Technische Universität Berlin, Berlin, Germany, 2002.
3. Demmel, J.W. *Applied Numerical Linear Algebra*; SIAM: Philadelphia, PA, USA, 1997.

4. Gajič, Z.; Qureshi, M.T.J. *Lyapunov Matrix Equation in System Stability and Control*; Dover Civil and Mechanical Engineering: Dover, Mineola, NY, USA, 2008.
5. Ionescu, V.; Oara, C.; Weiss, M. *Generalized Riccati Theroy and Robust Control: A Popov Function Approach*; John Wiley & Sons: Chichester, UK, 1999.
6. Petkov, P.; Christov, N.; Konstantinov, M. *Computational Methods for Linear Control Systems*; Prentice-Hall: Hertfordshire, UK, 1991.
7. Antoulas, A.C. *Approximation of Large-Scale Dynamical Systems*; SIAM: Philadelphia, PA, USA, 2005.
8. Alfke, D.; Feng, L.; Lombardi, L.; Antonini, G.; Benner, P. Model order reduction for delay systems by iterative interpolation. *Int. J. Numer. Methods Eng.* **2020**, *122*, 670–684. [CrossRef]
9. Benner, P.; Gugercin, S.; Willcox, K. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Rev.* **2015**, *57*, 483–531. [CrossRef]
10. Benner, P.; Mehrmann, V.; Sorensen, D.C. (Eds.) *Dimension Reduction of Large-Scale Systems*; Lecture Notes in Computational Science and Engineering; Springer: Berlin/Heidelberg, Germany, 2005; Volume 45.
11. Sorensen, D.C.; Antoulas, A.C. The sylvester equation and approximate balanced reduction. *Linear Algebra Appl.* **2002**, *352*, 671–700. [CrossRef]
12. Lin, Y. Cross-Gramian-based model reduction for descriptor systems. *Symmetry* **2022**, *14*, 2400. [CrossRef]
13. Benner, P.; Faßbender, H. On the numerical solution of large-scale sparse discrete-time Riccati equations. *Adv. Comput. Math.* **2011**, *35*, 119–147. [CrossRef]
14. Calvetti, D.; Reichel, L. Application of ADI iterative methods to the restoration of noisy images. *SIAM J. Matrix Anal. Appl.* **1996**, *17*, 165–186. [CrossRef]
15. Barraud, A.Y. A numerical algorithm to solve $A^TXA - X = Q$. *IEEE Trans. Autom. Control* **1977**, *22*, 883–885. [CrossRef]
16. Bartels, R.; Stewart, G. Solution of the equation $AX + XB = C$. *Commun. ACM* **1972**, *15*, 820–826. [CrossRef]
17. Hammarling, S. Numerical solution of the stable non-negative definite Lyapunov equation. *IMA J. Numer. Anal.* **1982**, *2*, 303–323. [CrossRef]
18. Hammarling, S. Numerical solution of the discrete-time, convergent, non-negative definite Lyapunov equation. *Syst. Control. Lett.* **1991**, *17*, 137–139. [CrossRef]
19. Varga, A. A note on Hammarling's algorithm for the discrete Lyapunov equation. *Syst. Control. Lett.* **1990**, *15*, 273–275. [CrossRef]
20. Antoulas, A.C.; Sorensen, D.C.; Zhou, Y. On the decay rate of Hankel singular values and related issues. *Syst. Control Lett.* **2002**, *46*, 323–342. [CrossRef]
21. Penzl, T. Eigenvalue decay bounds for solutions of Lyapunov equations: The symmetric case. *Syst. Control Lett.* **2000**, *40*, 139–144. [CrossRef]
22. Baker, J.; Embree, M.; Sabino, J. Fast singular value decay for Lyapunov solutions with nonnormal coefficients. *SIAM J. Matrix Anal. Appl.* **2015**, *36*, 656–668. [CrossRef]
23. Gugercin, S.; Sorensen, D.C.; Antoulas, A.C. A modified low-rank Smith method for large-scale Lyapunov equations. *Numer. Algorithms* **2003**, *32*, 27–55. [CrossRef]
24. Penzl, T. A cyclic low-rank smith method for large sparse lyapunov equations. *SIAM J. Sci. Comput.* **2000**, *21*, 1401–1418. [CrossRef]
25. Li, J.; White, J. Low rank solution of lyapunov equations. *SIAM J. Matrix Anal. Appl.* **2002**, *24*, 260–280. [CrossRef]
26. Benner, P.; Quintana-Ortí, E.S. Solving stable generalized Lyapunov equations with the matrix sign function. *Numer. Algorithms* **1999**, *20*, 75–100. [CrossRef]
27. Simoncini, V. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.* **2007**, *29*, 1268–1288. [CrossRef]
28. Stykel, T. Low-rank iterative methods for projected generalized Lyapunov equations. *Electron. Trans. Numer. Anal.* **2008**, *30*, 187–202.
29. Benner, P.; Kürschner, P. Computing real low-rank solutions of Sylvester equations by the factored ADI method. *Comput. Math. Appl.* **2014**, *67*, 1656–1672. [CrossRef]
30. Benner, P.; Kürschner, P.; Saak, J. Efficient handling of complex shift parameters in the low-rank Cholesky factor ADI method. *Numer. Algorithms* **2013**, *62*, 225–251. [CrossRef]
31. Bertram, C.; Faßbender, H. A quadrature framework for solving Lyapunov and Sylvester equations. *Linear Algebra Its Appl.* **2021**, *622*, 66–103. [CrossRef]
32. Benner, P.; Palitta, D.; Saak, J. On an integrated Krylov-ADI solver for large-scale Lyapunov equations. *Numer. Algorithms* **2023**, *92*, 1–29. [CrossRef]
33. Kürschner, P.; Freitag, M. Inexact methods for the low rank solution to large scale Lyapunov equations. *BIT Numer. Math.* **2020**, *92*, 1221–1259. [CrossRef]
34. Benner, P.; Kürschner, P.; Saak, J. Self-generating and efficient shift parameters in ADI methods for large lyapunov and sylvester equations. *Electron. Trans. Numer. Anal.* **2014**, *43*, 142–162.
35. Kürschner, P. Approximate residual-minimizing shift parameters for the low-rank ADI iteration. *Electron. Trans. Numer. Anal.* **2019**, *51*, 240–261. [CrossRef]
36. Smith, R. Matrix equation $XA + BX = C$. *SIAM J. Appl. Math.* **1968**, *16*, 198–201. [CrossRef]

37. Sokolov, V.I. Contributions to the Minimal Realization Problem for Descriptor Systems. Ph.D. Thesis, Fakultät für Mathematik, Technische Universität Chemnitz, Chemnitz, Germany, 2006.
38. Wang, G.; Wei, Y.; Qiao, S. *Generalized Inverses: Theory and Computations*; Science Press: Beijing, China; New York, NY, USA, 2004.
39. Benner, P.; Khoury, G.E.; Sadkane, M. On the squared Smith method for large-scale Stein equations. *Numer. Linear Algebra Appl.* **2014**, *21*, 645–665. [CrossRef]
40. Chan, T. Rank revealing QR factorizations. *Linear Algebra Its Appl.* **1987**, *88/89*, 67–82. [CrossRef]
41. Peaceman, D.; Rachford, H. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Ind. Appl. Math.* **1955**, *3*, 28–41. [CrossRef]
42. Wachspress, E. Iterative solution of the Lyapunov matrix equation. *Appl. Math. Lett.* **1988**, *1*, 87–90. [CrossRef]
43. Sadkane, M. A low-rank Krylov squared Smith method for large-scale discrete-time Lyapunov equations. *Linear Algebra Its Appl.* **2012**, *436*, 2807–2827. [CrossRef]
44. Benner, P.; Li, J.R.; Penzl, T. Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems. *Numer. Linear Algebra Appl.* **2008**, *15*, 755–777. [CrossRef]
45. Saad, Y. *Iterative Methods for Sparse Linear Systems*; SIAM: Philadelphia, PA, USA, 2003.
46. Golub, G.H.; Loan, C.F.V. *Matrix Computations*, 3rd ed.; Johns Hoplins University Press: Baltimore, MD, USA, 1996.
47. Benner, P.; Kürschner, P.; Saak, J. An improved numerical method for balanced truncation for symmetric second-order systems. *Math. Comput. Model. Dyn. Syst.* **2013**, *19*, 593–615. [CrossRef]
48. Benner, P.; Li, R.; Truhar, N. On the ADI method for Sylvester equations. *J. Comput. Appl. Math.* **2009**, *233*, 1035–1045. [CrossRef]
49. Ascher, U.; Greif, C. *A First Course in Numerical Methods*; SIAM: Philadelphia, PA, USA, 2011.