NUMERICAL SOLUTION OF PROJECTED ALGEBRAIC RICCATI EQUATIONS

PETER BENNER† AND TATJANA STYKEL‡

Abstract. We consider the numerical solution of projected algebraic Riccati equations using Newton’s method. Such equations arise, for instance, in model reduction of descriptor systems based on positive real and bounded real balanced truncation. We also discuss the computation of low-rank Cholesky factors of the solutions of projected Riccati equations. Numerical examples are given that demonstrate the properties of the proposed algorithms.

Key words. projected Riccati equation, projected Lyapunov equation, Newton’s method, alternating direction implicit (ADI) method

AMS subject classifications. 15A22, 15A24, 65F99, 65H10, 93B40

DOI. 10.1137/130923993

1. Introduction. Consider generalized algebraic Riccati equations of the form

\[ H + FXE^T + EXF^T + EXGXE^T = 0, \]

where \( E, F, G, H \in \mathbb{R}^{n \times n} \) are given matrices and \( X \in \mathbb{R}^{n \times n} \) is unknown. If \( E \) is nonsingular, then (1.1) can be transformed into a standard Riccati equation with \( E = I_n \). Such equations arise in many control problems for dynamical systems, including the linear-quadratic optimal regulator problem, \( H_2/H_\infty \) controller design, spectral factorization, and balancing-related model reduction, e.g., [19, 21, 28, 31]. The generalized Riccati equation (1.1) with singular \( E \) occurs in control problems for differential-algebraic equations or descriptor systems [3, 31, 45]. Unfortunately, the analysis of such an equation is more complicated compared to the standard case. In the literature, different types of generalized Riccati equations have been introduced for descriptor systems, e.g., [25, 26, 36, 50]. However, most of them are restricted to index one problems.

In this paper, we study the projected algebraic Riccati equation (PARE) of the form

\[ P_l H P_l^T + FXE^T + EXF^T + EXGXE^T = 0, \quad X = P_r X P_r^T, \]

where the pencil \( \lambda E - F \) is assumed to be regular, i.e., \( \det(\lambda E - F) \neq 0 \) for some \( \lambda \in \mathbb{C} \), and \( P_r \) and \( P_l \) are the spectral projectors onto the right and left deflating subspaces of \( \lambda E - F \) corresponding to the finite eigenvalues along the right and left deflating
considering the Riccati equation (1.1) with these methods rely on a Hamiltonian eigenvalue problem. Another approach is based on the Schur vector method [29], the sign function method [14], the structured doubling algorithm [15], Krylov subspace methods [22, 23], and symplectic methods [13]. All these methods rely on a Hamiltonian eigenvalue problem. Another approach is based on considering the Riccati equation (1.1) with \( E = I_n \) as a system of nonlinear equations.

Such a system can be solved by Newton’s method [4, 9, 27, 44]. In this paper, we present an extension of this method to the PARE (1.2). Based on a preliminary draft of this paper, the Newton–Kleinman iteration for the PARE (1.2) was already employed in [36]. We are not aware of any other approach to solve PAREs.

Throughout the paper, the open left half-plane is denoted by \( \mathbb{C}_- \). The matrices \( A^T \) and \( A^* \) stand for the transpose and conjugate transpose of \( A \), respectively, and \( A^{-T} = (A^{-1})^T \). An identity matrix of order \( n \) is denoted by \( I_n \), or simply by \( I \). For symmetric matrices \( A, B \in \mathbb{R}^{n,n} \), we write \( A > B \) (\( A \geq B \)) if \( A - B \) is positive definite (semidefinite). We denote by \( \|A\|_2 \) and \( \|A\|_F \) the spectral and Frobenius matrix norms of \( A \in \mathbb{R}^{n,m} \).

This paper is organized as follows. In section 2, we discuss the solvability of the PARE (1.2). In section 3, we present Newton and Newton–Kleinman iterations for solving this equation. We also study the convergence of these methods and discuss the computation of a stabilizing initial guess. Section 4 contains low-rank versions of Newton-type methods. Finally, some results of numerical experiments for the presented algorithms are reported in section 5.

2. Preliminaries. In this section, we give basic definitions and some notations from matrix analysis and control theory that will be used in the following. We also study the solvability of the PARE (1.2).

Any regular pencil \( \lambda E - F \) can be transformed into its Weierstrass canonical form

\[
E = T_l \begin{bmatrix} I_{n_f} & 0 \\ 0 & E_\infty \end{bmatrix} T_r, \quad F = T_l \begin{bmatrix} F_f & 0 \\ 0 & I_{n_\infty} \end{bmatrix} T_r,
\]

where \( T_l \) and \( T_r \) are the left and right nonsingular transformation matrices, \( F_f \in \mathbb{R}^{n_f,n_f} \) and \( E_\infty \in \mathbb{R}^{n_\infty,n_\infty} \) are matrices in Jordan canonical form, and \( E_\infty \) is nilpotent with index of nilpotency \( \nu \); see [17]. The eigenvalues of \( F_f \) are the finite eigenvalues of \( \lambda E - F \), and \( E_\infty \) corresponds to an eigenvalue at infinity. The number \( \nu \) is called the index of \( \lambda E - F \). The pencil \( \lambda E - F \) is called stable if all its finite eigenvalues have negative real part. Using the Weierstrass canonical form (2.1), the spectral projectors \( P_r \) and \( P_l \) onto the right and left deflating subspaces of the pencil \( \lambda E - F \) corresponding to the finite eigenvalues can be represented as

\[
P_r = T_r^{-1} \begin{bmatrix} I_{n_f} & 0 \\ 0 & 0 \end{bmatrix} T_r, \quad P_l = T_l \begin{bmatrix} I_{n_\infty} & 0 \\ 0 & 0 \end{bmatrix} T_l^{-1}.
\]

A triple \((E, F, G)\) is called stabilizable if \( \text{rank} [\lambda E - F, G] = n \) for all \( \lambda \in \mathbb{C} \setminus \mathbb{C}_- \). A triple \((E, F, H)\) is called detectable if \( \text{rank} [\lambda E^T - F^T, H^T] = n \) for all \( \lambda \in \mathbb{C} \setminus \mathbb{C}_- \). A solution \( X_\ast \) of the PARE (1.2) is called stabilizing if \( X_\ast \) is symmetric and the pencil \( \lambda E - F - EX_\ast GP \) is stable. If this pencil has all finite eigenvalues in the closed left
half-plane, then the symmetric solution $X_*$ of (1.2) is called semistabilizing. The following theorem gives sufficient conditions for the existence of a unique stabilizing solution of (1.2).

**Theorem 2.1.** Consider the PARE (1.2) with $G = G^T \geq 0$ and $H = H^T \geq 0$. If $(E, F, G)$ is stabilizable and $(E, F, H)$ is detectable, then (1.2) has a unique stabilizing solution $X_*$. 

**Proof.** Let $\lambda E - F$ be in Weierstrass canonical form (2.1) and let the matrices

$$\begin{align*}
G &= T_r^T \begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix} T_r, \\
H &= T_l^T \begin{bmatrix} H_{11} & H_{12} \\ H_{12}^T & H_{22} \end{bmatrix} T_l^T, \\
X &= T_r^{-1} \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} T_r^{-T}
\end{align*}$$

(2.3)

be partitioned into blocks accordingly to $E$ and $F$. Since $G$ and $H$ are both symmetric and positive semidefinite, the matrices $G_{11}$ and $H_{11}$ are also symmetric and positive semidefinite. Substituting (2.1), (2.2), and (2.3) into the PARE (1.2), we obtain that any solution of (1.2) has the form

$$X = T_r^{-1} \begin{bmatrix} X_{11} & 0 \\ 0 & 0 \end{bmatrix} T_r^{-T},$$

(2.4)

where $X_{11}$ satisfies the standard Riccati equation

$$H_{11} + F_f X_{11} + X_{11} F_f^T + X_{11} G_{11} X_{11} = 0.$$  

(2.5)

Since $(E, F, G)$ is stabilizable, we have

$$n = \text{rank}[\lambda E - F, G] = \text{rank} \begin{bmatrix} \lambda I - F_f & 0 & G_{11} \\ 0 & \lambda E_\infty - I & G_{12} \\ G_{11}^T & G_{12}^T & G_{22} \end{bmatrix}$$

$$= n_\infty + \text{rank}[\lambda I - F_f, G_{11}, G_{12}]$$

for all $\lambda \in \mathbb{C} \setminus \mathbb{C}_-$. Then

$$\text{rank}[\lambda I - F_f, G_{11}, G_{12}] = n_f \quad \text{for all } \lambda \in \mathbb{C} \setminus \mathbb{C}_-. $$

(2.6)

We now show that $\text{rank}[\lambda I - F_f, G_{11}] = n_f$ for all $\lambda \in \mathbb{C} \setminus \mathbb{C}_-$, i.e., the pair $(F_f, G_{11})$ is stabilizable in the classical sense. For this purpose, consider the Cholesky factorization $G = G_1 G_1^T$. Then for $T_r^{-1} G_1 = [T_1^T, T_2^T]^T$, we have

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix} = T_r^{-T} G T_r^{-1} = \begin{bmatrix} T_1 T_1^T & T_1 T_2^T \\ T_2 T_1^T & T_2 T_2^T \end{bmatrix}.$$

Assume that $\text{rank}[\lambda_0 I - F_f, G_{11}] < n_f$ for some $\lambda_0 \in \mathbb{C} \setminus \mathbb{C}_-$. In this case, there exists $v \neq 0$ such that $v^* [\lambda_0 I - F_f, G_{11}] v = 0$. Then $0 = v^* G_{11} v = v^* T_1 T_1^T v$ implies $v^* T_1 = 0$ and, hence,

$$v^* [\lambda_0 I - F_f, G_{11}, G_{12}] = v^* [\lambda_0 I - F_f, T_1 T_1^T, T_1 T_2^T] = 0.$$

This contradicts (2.6).

Analogously, we can show that the detectability of $(E, F, H)$ implies the detectability of the pair $(F_f, H_{11})$ in the classical sense. In this case, the Riccati equation (2.5) has a unique symmetric solution $X_{11}$ such that all the eigenvalues
of $F_j + X_{11}G_{11}$ have negative real part; see [28, Chapter 8]. For this $X_{11}$, the matrix $X$ in (2.4) is symmetric and the pencil $\lambda E - F - EXGP_r$ is stable. Thus, the PARE (1.2) has a unique stabilizing solution.

Note that in Theorem 2.1 we make no assumptions about the index of the pencil $\lambda E - F$. Similar to the standard case with $E = I$ (see [28, section 8.5]), the detectability condition can be weakened. If only a semistabilizable solution of (1.2) is required, then also stabilizability can be relaxed. Moreover, for projected Riccati equations arising in passivity-preserving balanced truncation of structured passive circuit equations, these conditions can be removed at all [35].

3. Newton’s method. Observing that the first equation in (1.2) is a system of nonlinear equations, we can solve it using Newton’s method.

Let $P$ be a projector and let $S_P = \{ X \in \mathbb{R}^{n,n} : X = X^T \text{ and } X = PXPT \}$. Consider a Riccati operator $R : S_{P_r} \rightarrow S_{P_l}$ given by

$$R(X) = P_lHP_l^T + FXE^T + EXF^T + EXGE^T.$$  

The Frechét derivative of $R$ at $X \in S_{P_r}$ is a linear operator $R_X' : S_{P_r} \rightarrow S_{P_l}$ defined as

$$R_X'(N) = \lim_{\delta \rightarrow 0} \frac{1}{\delta} (R(X + \delta N) - R(X))$$

for $N \in S_{P_r}$. Taking into account that $N = P_rN = NP_r^T$, we have

$$R_X'(N) = (F + EXGP_r)NE^T + (F + EXGP_r)^T.$$  

Then Newton’s method for the PARE (1.2) can be written as

$$N_j = -(R_X')^{-1}(R(X_j)),$$

$$X_{j+1} = X_j + N_j.$$  

The standard formulation of this method is given in Algorithm 1.

\textbf{Algorithm 1.} Newton’s method.

\textbf{Input:} $E, F, G, H \in \mathbb{R}^{n,n}$, projectors $P_r$ and $P_l$, a stabilizing initial guess $X_0 \in S_{P_r}$.

\textbf{Output:} An approximate solution of the PARE (1.2).

\textbf{FOR} $j = 0, 1, 2, \ldots$

1. Compute $F_j = F + EX_jGP_r$.

2. Solve the projected algebraic Lyapunov equation (PALE)

$$F_jN_jE^T + EN_jF_j^T = -P_lR(X_j)P_l^T, \quad N_j = P_rN_jP_r^T.$$  

3. Compute $X_{j+1} = X_j + N_j$.

\textbf{END FOR}

As in the standard case [27], we can combine the second and third steps in Algorithm 1 and compute the new iterate $X_{j+1}$ directly from the projected Lyapunov equation as presented in Algorithm 2.

Although Algorithms 1 and 2 are mathematically equivalent, they behave differently in finite precision arithmetic and there are significant differences in their implementation especially for large-scale problems. We will compare the Newton and Newton–Kleinman methods in section 4.5.

\textbf{Remark 3.1.} Due to $X_j = P_rX_j$ and $P_rE = EP_r$, the matrices $F_j$ in Algorithms 1 and 2 satisfy $F_j = F + EX_jGP_r = F + P_lEX_jGP_r$. Then using the Weierstrass
canonical form (2.1) and representations (2.2), we obtain that \( P_l \) and \( P_r \) are the spectral projectors onto the left and right deflating subspaces corresponding to the finite eigenvalues not only of \( \lambda E - F \) but also of \( \lambda E - F_j \).

3.1. Convergence. First, we investigate the convergence of Algorithms 1 and 2. The following theorem establishes that the PALEs (3.1) and (3.2) are solvable and the iterate \( X_j \) converges in both algorithms to a stabilizing solution of the PARE (1.2) for every stabilizing \( X_0 \).

Theorem 3.2. Let \( X_0 \in \mathbb{S}_P \) be chosen such that the pencil \( \lambda E - F - EX_0GP_r \) is stable. Assume that the PARE (1.2) with \( G = G^T \geq 0 \) has a unique stabilizing solution \( X_* \).

Then for the iterate \( X_j \) in Algorithm 1 or 2, we have

(i) the pencil \( \lambda E - F_j \) with \( F_j = F + EX_jGP_r \) is stable for all \( j \geq 0 \);
(ii) \( X_1 \leq X_2 \leq \cdots \leq X_j \leq X_{j+1} \leq \cdots \leq X_* \);
(iii) \( \lim_{j \to \infty} X_j = X_* \) and \( \lim_{j \to \infty} R(X_j) = 0 \);
(iv) there exists a constant \( \gamma > 0 \) such that \( \|X_* - X_{j+1}\| \leq \gamma \|X_* - X_j\|^2 \) for \( j \geq 1 \), i.e., the iterate \( X_j \) converges globally and quadratic to \( X_* \).

Proof. This theorem can be proved in two different ways. The first approach is based on transforming the pencil \( \lambda E - F \) into the Weierstrass canonical form (2.1) and applying the classical convergence results [5, 44] to the standard Riccati equation (2.5). On the other hand, these results can be reformulated in terms of the original data. We choose the second approach.

(i) The stability of \( \lambda E - F_j \) with \( F_j = F + EX_jGP_r \), \( j = 0, 1, \ldots \), can be proved by induction. The pencil \( \lambda E - F_0 \) is stable by the choice of \( X_0 \). Assume now that \( \lambda E - F_j \) is stable. Then the PALE (3.1) has a unique symmetric solution \( N_j \) [40]. Since \( X_0 \) is symmetric, the iterate \( X_{j+1} = X_j + N_j \) is also symmetric. Subtracting the PALE (3.2) for \( X_{j+1} \) from the PARE (1.2) with \( X \) replaced by \( X_* \), we obtain that the difference \( D_{j+1} = X_* - X_{j+1} \) satisfies the PALE

\[
F_jD_{j+1}E^T + ED_{j+1}F_j^T = -P_lED_jGD_jE^TP_l^T, \quad D_{j+1} = P_rD_{j+1}P_r^T.
\]

Since \( \lambda E - F_j \) is stable and the right-hand side in the first equation in (3.3) is symmetric and negative semidefinite, then (3.3) has a unique symmetric, positive semidefinite solution \( D_{j+1} \); see [40]. Thus, \( X_{j+1} \leq X_* \).

We now show that \( \lambda E - F_{j+1} \) is stable. The first equation in (3.3) can be written as

\[
F_{j+1}D_{j+1}E^T + ED_{j+1}F_{j+1}^T = -ED_{j+1}GD_{j+1}E^T - EN_jGN_jE^T.
\]
Assume that $\lambda E - F_{j+1}$ has an eigenvalue $\lambda_0$ with $\text{Re}(\lambda_0) \geq 0$. Let $v = P_j v$ be a left eigenvector of $\lambda E - F_{j+1}$ corresponding to $\lambda_0$, i.e., $\lambda_0 v^* E = v^* F_{j+1}$. Premultiplying (3.4) by $v^*$ and postmultiplying by $v$, we obtain that

$$2\text{Re}(\lambda_0) v^* E D_{j+1} E^Tv = -v^* E D_{j+1} G D_{j+1} E^Tv - v^* E N_j G N_j E^Tv.$$ 

Since $D_{j+1}$ and $G$ are both symmetric, positive semidefinite and $\text{Re}(\lambda_0) \geq 0$, we have $GD_{j+1} E^Tv = 0$ and $GN_j E^Tv = 0$. Then

$$v^* F_j = v^* (F + E X_j G P_r) = v^* (F + EX_{j+1} G P_r) = v^* F_{j+1} = \lambda_0 v^* E.$$ 

This contradicts the stability of $\lambda E - F_j$. Thus, $\lambda E - F_{j+1}$ is stable.

(ii) It follows from the equations (1.2) with $X = X_*$ and (3.1) that

$$\mathcal{R}(X_j) = E N_{j-1} G N_{j-1} E^T, \quad j \geq 1.$$ 

Since $\lambda E - F_j$ is stable and $\mathcal{R}(X_j)$ is symmetric, positive semidefinite, the PALE (3.1) has a unique symmetric, positive semidefinite solution $N_j = X_{j+1} - X_j$. Hence, $X_j \leq X_{j+1}$ for $j \geq 1$.

(iii) Since the sequence $\{X_j\}_{j \geq 1}$ is nondecreasing and bounded above by $X_*$, this sequence is convergent, i.e., $\lim_{j \to \infty} X_j = X_\infty$ with symmetric $X_\infty \leq X_*$. Hence the relation (3.5) implies $\lim_{j \to \infty} \mathcal{R}(X_j) = 0$. Passing to the limit in (3.2), we obtain that $X_\infty$ solves the PARE (1.2). Replacing $X$ by $X_\infty$ and $X_*$ in (1.2) and considering the difference of the resulting equations, we find that

$$F_*(X_\infty - X_*) E^T + E (X_\infty - X_*) F_*^T = -P_1 E (X_\infty - X_*) G (X_\infty - X_*) E^T P_1^T,$$

$$X_\infty - X_* = P_1 (X_\infty - X_*) P_1^T,$$

with $F_* = F + E X_* G P_r$. Since $\lambda E - F_*$ is stable and $G$ is symmetric, positive semidefinite, the PALE (3.6) has a unique symmetric, positive semidefinite solution $X_\infty - X_*$, i.e., $X_\infty \geq X_*$. Thus, $X_\infty = X_*$.

(iv) Subtracting the PALE (3.2) for $X_{j+1}$ from the PARE (1.2) with $X$ replaced by $X_*$, we obtain that the difference $D_{j+1} = X_* - X_{j+1}$ satisfies the PALE

$$F_* D_{j+1} E^T + E D_{j+1} F_*^T = -P_1 E (N_j G N_j - D_{j+1} G E P_r^T) E^T P_1^T,$$

$$D_{j+1} = P_1 D_{j+1} P_1^T.$$ 

Since $\lambda E - F_*$ is stable, this equation has a unique solution given by

$$0 \leq D_{j+1} = \frac{1}{2\pi} \int_{-\infty}^{\infty} (i\omega E - F_*)^{-1} P_1 E (N_j G N_j - D_{j+1} G P_r^T) E^T P_1^T (-i\omega E - F_*)^{-T} d\omega;$$

see [40]. Taking norms, we get

$$\|X_* - X_{j+1}\| \leq \left\| \frac{1}{2\pi} \int_{-\infty}^{\infty} (i\omega E - F_*)^{-1} P_1 E N_j G N_j E^T P_1^T (-i\omega E - F_*)^{-T} d\omega \right\| \|N_j\| \|E\| \|G\| \left\| \frac{1}{2\pi} \int_{-\infty}^{\infty} (i\omega E - F_*)^{-1} P_1 P_1^T (-i\omega E - F_*)^{-T} d\omega \right\|$$

$$\leq \gamma \|X_{j+1} - X_j\|^2.$$ 

Since $0 \leq X_{j+1} - X_j \leq X_* - X_j$, we obtain that $\|X_* - X_{j+1}\| \leq \gamma \|X_* - X_j\|^2$. 

It follows from the proof of Theorem 3.2 that the stabilizing solution $X_*$ of the PARE (1.2) is minimal in the sense that $X_* \leq X$ for all symmetric solutions $X$ of
(1.2). Theorem 3.2 also shows that in each step of Algorithms 1 and 2, all iterates $X_j$ are stabilizing once the initial guess $X_0$ is chosen to be stabilizing. Such a matrix exists if, for example, $(E, F, G)$ is stabilizable. If this condition is violated, it is still possible, similarly to the standard case [7], to construct a convergent Newton–Kleinman iteration for computing a minimal solution of (1.2).

Note that the existence of a stabilizing solution of the PARE (1.2) guarantees a quadratic convergence of the Newton iteration. However, if (1.2) has only a semi-stabilizing solution $X_*$, then the quadratic convergence may be lost. For the standard Riccati equation with negative (semi)definite quadratic term ($G \leq 0$), a modification of the Newton iteration has been proposed in [20] which has a linear rate of convergence under assumption that the purely imaginary eigenvalues of $\lambda E - F - EX_*GP_r$ are semisimple. This result can also be extended to the PARE (1.2) with positive semidefinite quadratic term.

At each iteration step of Algorithms 1 and 2, we have to solve the PALEs (3.1) and (3.2), respectively. For small and medium-size problems, such equations can be solved using the generalized Schur–Bartels–Stewart method or the generalized Schur–Hammarling method [39]. For large dense problems, we can use the modified sign function method [41], whereas projected Lyapunov equations with large-scale sparse matrix coefficients can be solved using the generalized alternating direction implicit (ADI) method [42] or Krylov subspace methods [43]. We will discuss a combination of Lyapunov solvers with Newton iteration in section 4 in more detail.

3.2. Computing the stabilizing initial guess. The convergence of Algorithms 1 and 2 relies on a stabilizing initial guess $X_0$ that also satisfies $X_0 \in S_{Pr}$. If $\lambda E - F$ is stable, then $X_0 = 0$ trivially satisfies these demands. This is often the case in applications, but certainly not always. Thus, computing such $X_0$ is required in the unstable situations. The stabilization of descriptor systems using partial stabilization, i.e., computing $X_0$ such that the stable and infinite eigenvalues of $\lambda E - F$ remain unchanged and the unstable ones are moved to the open left half-plane, is considered in [6]. The suggested procedures basically use a numerically robust variant to compute a block form as in (2.1), where the nonzero blocks are not required to have a special structure. Then using the decomposition as in (2.3), the stabilization problem can be solved using the Bass algorithm or an algebraic Bernoulli equation as in the standard case. Though the symmetry condition $X_0 \in S_{Pr}$ is not considered in [6], using the same decomposition as in (2.3) with $G = 0$ and the resulting form of the solutions $X$ as in (2.4) of the Lyapunov or Bernoulli equations, it is easy to show that the associated $X_0$ matrices satisfy this symmetry equation. The solution of these Lyapunov or Bernoulli equations can also be obtained in factored forms as in [1, 2, 41, 42].


4.1. Computing the Cholesky factor of the stabilizing solution. In many control applications including positive real and bounded real balanced truncation model reduction, the matrices $G$ and $H$ in the PARE (1.2) are given in factored form $G = \tilde{G}^T \tilde{G}$ and $H = \tilde{H} \tilde{H}^T$, where $\tilde{G} \in \mathbb{R}^{p,n}$ and $\tilde{H} \in \mathbb{R}^{n,m}$. In this case, the stabilizing positive semidefinite solution of (1.2) can also be determined in factored form $X_* = \tilde{X}_* \tilde{X}_*^T$. Note that the computation of the Cholesky factorization $X_* = \tilde{X}_* \tilde{X}_*^T$ should be avoided because the computed Cholesky factor $\tilde{X}_*$ has usually lower accuracy than $X_*$. Fortunately, the factor $\tilde{X}_*$ of $X_*$ can be computed directly without calculating $X_*$ itself. This can be done in two different ways based on Algorithms 1
and 2, respectively. We consider first Newton’s method in Algorithm 1. It follows from $EP_r = P_rE$, $FP_r = P_rF$, and $X_j = P_rX_jP_r^T$ that $R(X_j) = P_rR(X_j)P_r^T$. Then taking into account (3.1), we obtain that

\begin{equation}
R(X_{j+1}) = R(X_j + N_j) = R(X_j) + F_jN_jE^T + EN_jF_j^T + EN_j\tilde{G}^T\tilde{G}N_jE^T = EN_j\tilde{G}^T\tilde{G}N_jE^T.
\end{equation}

In this case, the PALE (3.1) can be replaced by

\begin{equation}
F_jN_jE^T + EN_jF_j^T = -P_lK_jK_l^TP_l^T, \quad N_j = P_rN_jP_r^T,
\end{equation}

with $K_j = EN_{j-1}\tilde{G}^T$ for $j > 0$. Since $\lambda E - F_j$ is stable, (4.2) has a unique symmetric, positive semidefinite solution $N_j$ that can be factorized as $N_j = \tilde{N}_j\tilde{N}_j^T$.

For stable $\lambda E - F$, we can start with $X_0 = 0$ and solve the PALE

\begin{equation}
FN_0E^T + EN_0F^T = -P_l\tilde{H}\tilde{H}^TP_l^T, \quad N_0 = P_rN_0P_r^T,
\end{equation}

for a Cholesky factor $\tilde{N}_0$ of $N_0 = \tilde{N}_0\tilde{N}_0^T$. Otherwise, we first compute the Cholesky factorization $R(X_0) = K_0K_0^T$ for some stabilizing starting guess $X_0$ and determine the Cholesky factor of the solution of the PALE

\begin{equation}
F_0N_0E^T + EN_0F_0^T = -P_lK_0K_l^TP_l^T, \quad N_0 = P_rN_0P_r^T.
\end{equation}

with $F_0 = F + EX_0\tilde{G}^T\tilde{G}P_r$. If $R(X_0)$ is indefinite, one can compute the solution of (3.1) for $j = 0$ and then employ (4.2) starting with $j = 1$. Once we have $X_j = \tilde{X}_j\tilde{X}_j^T$ and $N_j = \tilde{N}_j\tilde{N}_j^T$, the next iterate can be obtained in factored form as

\begin{equation}
X_{j+1} = X_j + N_j = \tilde{X}_{j+1}\tilde{X}_{j+1}^T,
\end{equation}

where $\tilde{X}_{j+1} \in \mathbb{R}^{n,n}$ is computed from the LQ factorization $[\tilde{X}_j, \tilde{N}_j] = [\tilde{X}_{j+1}, 0]\tilde{Q}_j$ with orthogonal $\tilde{Q}_j$. Note that the Cholesky factors $\tilde{N}_j$ of the solutions of the PALEs (4.2)–(4.4) can be determined directly without computing $N_j$ itself by using the generalized Schur–Hammarling method [39]. This method does not require the preliminary computation of the projectors $P_r$ and $P_l$. It is based on reducing the pencil $\lambda E - F_j$ to the generalized Schur form [18] and solving the generalized Sylvester and Lyapunov equations. Using the fact that the pencils $\lambda E - F$ and $\lambda E - F_j$ have the same deflating subspaces corresponding to the finite eigenvalues, we do not need to compute the generalized Schur form of $\lambda E - F_j$ at every Newton iteration. It is enough to reduce $\lambda E - F$ into the generalized Schur form and solve the projected Lyapunov equations on the subspace corresponding to the finite eigenvalues. We summarize the resulting Newton’s method in Algorithm 3.

For computing the generalized Schur form (4.5), we can use the QZ algorithm [18, 48] or the GUPTRI algorithm [16]. To solve the generalized Sylvester equation (4.6) one can use the generalized Schur method [24]. The Cholesky factor of the solutions of the generalized Lyapunov equations (4.7) and (4.8) can be determined using the generalized Hammarling method [33]. The Newton–Schur–Hammarling method for the PARE (1.2) costs $O(n^3)$ flops and has the memory complexity $O(n^2)$. 

**Input:** $E, F \in \mathbb{R}^{n,n}$, $G \in \mathbb{R}^{p,n}$, $H \in \mathbb{R}^{n,m}$, and a stabilizing initial guess $X_0 \in S_P$.

**Output:** An approximate solution $\hat{X} \hat{X}^T$ of the PARE (1.2).

1. Compute the generalized Schur form

\[
    V^T E U = \begin{bmatrix} E_{11} & E_{12} \\ 0 & E_{22} \end{bmatrix}, \quad V^T F U = \begin{bmatrix} F_{11} & F_{12} \\ 0 & F_{22} \end{bmatrix},
\]

where $V$ and $U$ are orthogonal, $E_{11}$ is upper triangular, nonsingular and $E_{22}$ is upper triangular with zeros on the diagonal, $F_{11}$ is upper quasi-triangular and $F_{22}$ is upper triangular, nonsingular.

2. Solve the generalized Sylvester equation

\[
    E_{11} Y - Z E_{22} = -E_{12}, \quad F_{11} Y - Z F_{22} = -F_{12}.
\]

3. Compute the matrices

\[
    \hat{G} U = \begin{bmatrix} \hat{G}_1, \hat{G}_2 \end{bmatrix}, \quad V^T \hat{H} = \begin{bmatrix} \hat{H}_1 \\ \hat{H}_2 \end{bmatrix}, \quad U^T X_0 U = \begin{bmatrix} X_{11,0} & 0 \\ 0 & 0 \end{bmatrix}.
\]

4. Compute the Cholesky factorizations

\[
    X_{11,0} = \tilde{X}_{11,0} \tilde{X}_{11,0}^T, \quad R_1(X_{11,0}) = F_{11} X_{11,0} E_{11}^T + E_{11} X_{11,0} F_{11}^T + E_{11} X_{11,0} \hat{G}_1 E_{11} + (\hat{H}_1 - Z \hat{H}_2)(\hat{H}_1 - Z \hat{H}_2)^T = K_{1,0} K_{1,0}^T
\]

and the matrix $F_{11,0} = F_{11} + E_{11} X_{11,0} \hat{G}_1^T \hat{G}_1$.

5. Solve the generalized Lyapunov equation

\[
    F_{11,0} N_{11,0} E_{11}^T + E_{11} N_{11,0} F_{11,0}^T = -K_{1,0} K_{1,0}^T
\]

for the Cholesky factor $\tilde{N}_{11,0}$ of $N_{11,0} = \tilde{N}_{11,0} \tilde{N}_{11,0}^T$.

6. Compute the LQ factorization $[\tilde{X}_{11,0}, \tilde{N}_{11,0}] = [\tilde{X}_{11,1}, 0] Q_{1,1}$.

7. FOR $j = 1, 2, \ldots, j_{\text{max}} - 1$

   (a) Compute $K_{1,j} = E_{11} \tilde{N}_{11,j-1} \tilde{N}_{11,j-1}^T \hat{G}_1^T$ and $F_{11,j} = F_{11,j-1} + K_{1,j} \hat{G}_1$.

   (b) Solve the generalized Lyapunov equation

\[
    F_{11,j} N_{11,j} E_{11}^T + E_{11} N_{11,j} F_{11,j}^T = -K_{1,j} K_{1,j}^T
\]

for the Cholesky factor $\tilde{N}_{11,j}$ of $N_{11,j} = \tilde{N}_{11,j} \tilde{N}_{11,j}^T$.

   (c) Compute the LQ factorization $[\tilde{X}_{11,j}, \tilde{N}_{11,j}] = [\tilde{X}_{11,j+1}, 0] Q_{1,j}$.

END FOR

8. Compute

\[
    \hat{X} = U \begin{bmatrix} \tilde{X}_{11,j_{\text{max}}} \\ 0 \end{bmatrix}.
\]
4.2. Low-rank approximation to the stabilizing solution. If the eigenvalues of the stabilizing solution $X_\ast$ decay to zero very rapidly, then $X_\ast$ can be well approximated by a matrix of low rank. Such a low-rank approximation can be computed in factored form $X_\ast \approx \hat{X}\hat{X}^T$ with $\hat{X} \in \mathbb{R}^{n,k}$, $k \ll n$, by solving the PALEs (4.2) and (4.3) for low-rank approximate solutions. For this purpose, we can use an extension of the low-rank sign function method, the low-rank ADI method, or Krylov subspace methods to projected Lyapunov equations as presented in [41, 42, 43].

**Algorithm 4.** Low-rank Newton method.

**Input:** $E, F \in \mathbb{R}^{n,n}$ such that $\lambda E - F$ is stable, $G \in \mathbb{R}^{p,n}$, $\tilde{H} \in \mathbb{R}^{n,m}$, projectors $P_r$, $P_l$.

**Output:** A low-rank Cholesky factor of the stabilizing solution of the PARE (1.2).

1. Solve the PALE (4.3) for the low-rank Cholesky factor $\tilde{N}_0$ such that $N_0 \approx \tilde{N}_0\tilde{N}_0^T$.
2. Set $\hat{X}_1 = \tilde{N}_0$, $K_1 = EN_0\tilde{N}_0^T\hat{G}^T$, $F_1 = F + K_1\hat{G}P_r$.
3. FOR $j = 1, 2, \ldots$
   (a) Solve the PALE (4.2) for the low-rank Cholesky factor $\tilde{N}_j$ such that $N_j \approx \tilde{N}_j\tilde{N}_j^T$.
   (b) Compute $\hat{X}_{j+1} = [\hat{X}_j, \tilde{N}_j]$.
   (c) Compute $K_{j+1} = EN_j\tilde{N}_j^T\hat{G}^T$ and $F_{j+1} = F_j + K_{j+1}\hat{G}P_r$.
END FOR

The computation of the low-rank Cholesky factor of the stabilizing solution of the PARE (1.2) with stable $\lambda E - F$ is summarized in Algorithm 4. Note that in each iterative step in this algorithm the number of columns of the approximate Cholesky factor $\hat{X}_j$ of the solution of (1.2) increases by the number $k_j$ of columns of the approximate Cholesky factor $\tilde{N}_j$ of the solution of the PALE (4.3). In the case of large $k_j$ or slow convergence of the Newton iteration, a large workspace is required to store $\hat{X}_{j+1}$. In order to keep low-rank structure in $\hat{X}_{j+1}$, one can replace this iterate by its low-rank approximation computed via a rank-revealing QR decomposition

\begin{equation}
[\hat{X}_j, \tilde{N}_j]^T = Q_j \begin{bmatrix} R_{j,1} & R_{j,2} \\ 0 & R_{j,3} \end{bmatrix} \Pi_j^T,
\end{equation}

where $Q_j$ has orthogonal columns, $\Pi_j$ is a permutation matrix, $R_{j,1}$ has full row rank, and $\|R_{j,3}\|_F \leq \tau \|[\hat{X}_j, \tilde{N}_j]\|_F$ for some small tolerance $\tau$. Setting $R_{j,3} = 0$, we can proceed with the new iterate $\hat{X}_{j+1} = \Pi_j [R_{j,1}, R_{j,2}]^T$. Note that in (4.9) we do not need to accumulate the matrix $Q_j$.

Low-rank Cholesky factors of the stabilizing solution of the PARE (1.2) can also be computed using the Newton–Kleinman method in Algorithm 2. Such an approach has been considered previously for the case $E = I$ in [44]. An extension of this approach is given in Algorithm 5.

If $\lambda E - F$ is stable, in step 3 of Algorithm 5 we can solve the PALE

\begin{equation}
FXE^T + EXF^T = -P_l\tilde{H}\tilde{H}^TP_l^T, \quad X = P_rXP_r^T
\end{equation}

with $\tilde{H} = [\hat{H}, E(\hat{X}_{1,j_{\text{max}}}^T\hat{X}_{1,j_{\text{max}}} - \hat{X}_{2,j_{\text{max}}}^T\hat{X}_{2,j_{\text{max}}}^T)\hat{G}^T]$.

Note that in Algorithms 4 and 5, the projectors $P_l$ and $P_r$ are required in explicit form. The computation of these projectors is, in general, very expensive. Fortunately, in many applications including control of fluid flow, electrical circuit simulation, and
the Lyapunov residuals. Here, we extend this result to projected Lyapunov equations. A low-rank Cholesky factor of the stabilizing solution of the PARE (1.2).

1. Set \( \hat{X}_{1,0} = X_0 \) and \( \hat{X}_{2,0} = 0 \).
2. FOR \( j = 0, 1, 2, \ldots, J_{\text{max}} - 1 \):
   (a) Compute \( K_j = E(X_{1,j} \hat{X}_{1,j}^T - \hat{X}_{2,j} \hat{X}_{2,j}^T) \tilde{G}^T \) and \( F_j = F + K_j \tilde{G} P_r \).
3. Solve the PALE

\[
F_j X_{1,j+1} E^T + E X_{1,j+1} F_j^T = -P_t \tilde{H} \tilde{H}^T P_r^T, \quad X_{1,j+1} = P_r X_{1,j+1} P_r^T,
\]

\[
F_j X_{2,j+1} E^T + E X_{2,j+1} F_j^T = -P_l K_j K_j^T P_l^T, \quad X_{2,j+1} = P_r X_{2,j+1} P_r^T
\]

for the low-rank Cholesky factors \( \hat{X}_{1,j+1} \) and \( \hat{X}_{2,j+1} \) such that \( X_{1,j+1} \approx \hat{X}_{1,j+1} \hat{X}_{1,j+1}^T \) and \( X_{2,j+1} \approx \hat{X}_{2,j+1} \hat{X}_{2,j+1}^T \).

END FOR

3. Solve the PALE

\[
\tilde{F} X E^T + E \tilde{F} X^T = -P_t \tilde{H} \tilde{H}^T P_r^T, \quad X = P_r X P_r^T
\]

with \( \tilde{F} = F + \frac{1}{2} E(\tilde{X}_{1,max} \hat{X}_{1,max}^T - \tilde{X}_{2,max} \hat{X}_{2,max}^T) \tilde{G}^T \tilde{G} P_r \) for the low-rank Cholesky factor \( \hat{X} \) such that \( X \approx \hat{X} \hat{X}^T \).

Algorithm 5: Low-rank Newton-Kleinman method.

**INPUT:** \( E, F \in \mathbb{R}^{p,n}, G \in \mathbb{R}^{p,n}, \tilde{H} \in \mathbb{R}^{n,m} \), projectors \( P_l \) and \( P_r \), a low-rank matrix \( X_0 \) such that \( \hat{X}_0 = P_r \hat{X}_0 \) and \( \lambda E - F = -\hat{X}_0 \hat{X}_0^T \tilde{G}^T P_r \) is stable.

**OUTPUT:** A low-rank Cholesky factor of the stabilizing solution of the PARE (1.2).

**4.3. Low-rank ADI iteration for projected Lyapunov equations.** In this section, we briefly discuss the computation of approximate solutions to the PALE

\[
(4.13) \quad \tilde{F} Z E^T + E \tilde{F} Z^T = -P_t \tilde{H} \tilde{H}^T P_r^T, \quad Z = P_r Z P_r^T,
\]

with \( \tilde{F} \in \mathbb{R}^{n,n} \), \( \tilde{K} \in \mathbb{R}^{n,n} \) and unknown \( Z \in \mathbb{R}^{n,n} \). According to Remark 3.1 we can assume that \( P_l \) and \( P_r \) are the spectral projectors onto the left and right deflating subspaces of \( \lambda E - \tilde{F} \) corresponding to the finite eigenvalues. For solving the PALE (4.13), we use the ADI method. This method was first proposed for standard Lyapunov equations [6, 30, 34, 44] and then extended in [42] to projected Lyapunov equations. Recently, a more efficient version of the ADI iteration for standard Lyapunov equations was proposed in [8] which allows a cheap computation of the Lyapunov residuals. Here, we extend this result to projected Lyapunov equations.

The generalized ADI iteration for the PALE (4.13) is given by

\[
(4.14) \quad (E + \tau_k \tilde{F}) Z_{k-1/2} \tilde{F}^T + \tilde{F} Z_{k-1} (E - \tau_k \tilde{F})^T = -P_l \tilde{K} \tilde{K}^T P_l^T, \quad \tau_k \in \mathbb{C}_- \quad \text{with an initial matrix} \quad Z_0 = 0 \quad \text{and shift parameters} \quad \tau_1, \ldots, \tau_k \in \mathbb{C}_-.
\]

Solving the first equation for \( Z_{k-1/2} \) and the second equation for \( Z_k \), we obtain

\[
Z_k = (E + \tau_k \tilde{F})^{-1} (E - \tau_k \tilde{F}) Z_{k-1} (E - \tau_k \tilde{F})^T (E + \tau_k \tilde{F})^{-T} - 2 \text{Re}(\tau_k) (E + \tau_k \tilde{F})^{-1} P_l \tilde{K} \tilde{K}^T P_l^T (E + \tau_k \tilde{F})^{-T}.
\]
Since the solution of the PALE (4.13) also satisfies this equation, we get the following expression for the error:

\[
Z_k - Z = (E + \tau_k \hat{F})^{-1} (E - \tau_k \hat{F}) (Z_{k-1} - Z) (E - \tau_k \hat{F})^T (E + \tau_k \hat{F})^{-T} = \cdots = -A_k Z A_k^*,
\]

where \(A_k = (E + \tau_k \hat{F})^{-1} (E - \tau_k \hat{F}) \cdots (E + \tau_{i+1} \hat{F})^{-1} (E - \tau_{i+1} \hat{F})\). If the pencil \(\lambda E - \hat{F}\) is stable, then \(Z_k\) converges toward the solution of the PALE (4.13). The rate of convergence depends strongly on the choice of the shift parameters. The optimal shift parameters providing the superlinear convergence satisfy the generalized ADI minimax problem

\[
\{\tau_1, \ldots, \tau_q\} = \arg \min_{\{\tau_1, \ldots, \tau_q\} \in \mathbb{C}^q} \max_{e \in \text{Sp}_f(E, \hat{F})} \frac{|(1 - \tau_1 t) \cdots (1 - \tau_q t)|}{|(1 + \tau_1 t) \cdots (1 + \tau_q t)|},
\]

where \(\text{Sp}_f(E, \hat{F})\) denotes the finite spectrum of the pencil \(\lambda E - \hat{F}\). Similarly to [34], the suboptimal ADI parameters can be obtained from a set of largest and smallest in modulus approximate finite eigenvalues of \(\lambda E - \hat{F}\) computed by an Arnoldi procedure. Other parameter selection techniques developed for standard Lyapunov minimax problems [10, 38, 47] can also be used for the PALE (4.13).

The ADI iteration is terminated if the normalized residual satisfies the condition

\[
\theta_L(Z_k) = \frac{\|L(Z_k)\|_F}{\|P_1 \hat{F}^T P_k^T \|_F} \leq \text{tol},
\]

where \(\text{tol}\) is a user-defined tolerance, and

\[
L(Z_k) = \hat{F} Z_k E^T + E Z_k \hat{F}^T + P_l \hat{K} \hat{F}^T P_k^T
\]
is the Lyapunov residual. The following theorem shows that even though \(L(Z_k)\) is a large and dense matrix, it has a low rank.

**Theorem 4.1.** The Lyapunov residual at step \(k\) of the ADI iteration has the form

\[
L(Z_k) = \dot{A}_k P_1 \hat{K} \hat{F}^T P_k^T \dot{A}_k^*,
\]

where \(\dot{A}_k = (E - \tau_k \hat{F})(E + \tau_k \hat{F})^{-1} \cdots (E - \tau_{i+1} \hat{F})(E + \tau_{i+1} \hat{F})^{-1}\).

**Proof.** It follows from (4.13) and (4.15) that

\[
L(Z_k) = \hat{F} Z_k E^T + E Z_k \hat{F}^T + P_l \hat{K} \hat{F}^T P_k^T = \hat{F} (Z_k - Z) E^T + E (Z_k - Z) \hat{F}^T
\]

\[
= -\hat{F} A_k Z A_k^* E^T - E A_k Z A_k^* \hat{F}^T.
\]

Since \(P_l\) and \(P_k\) are the spectral projectors onto the right and left deflating subspaces of the pencil \(\lambda E - \hat{F}\) corresponding to the finite eigenvalues, one can show using the Weierstrass canonical form of \(\lambda E - \hat{F}\) that

\[
E (E + \tau_k \hat{F})^{-1} (E - \tau_k \hat{F}) = (E - \tau_k \hat{F})(E + \tau_k \hat{F})^{-1} E,
\]

\[
\hat{F} (E + \tau_k \hat{F})^{-1} (E - \tau_k \hat{F}) = (E - \tau_k \hat{F})(E + \tau_k \hat{F})^{-1} \hat{F}.
\]

Then we have \(\hat{F} A_k Z A_k^* E^T = \dot{A}_k \hat{F} Z E^T \dot{A}_k^*\) and, hence,

\[
L(Z_k) = -\dot{A}_k (\hat{F} Z E^T + E \hat{F}^T) \dot{A}_k^* = \dot{A}_k P_1 \hat{K} \hat{F}^T P_k^T \dot{A}_k^*.
\]

□
It follows from (4.17) that \( L(Z_k) \) is of rank at most \( g \) and its Frobenius norm can be computed as \( \| L(Z_k) \|_F = \| W Z_k \|_F = \| W_k^T Z_k \|_F \) with \( W_k = \tilde{A}_k P \tilde{K} \in \mathbb{R}^{n \times g} \). We now show that the matrix \( W_k \) can be obtained iteratively at low cost.

It has been shown in [42] that the iterate \( Z_k \) can be determined in factored form \( Z_k = \tilde{Z}_k \tilde{Z}_k^T \), where

\[
V_1 = (E + \tau_1 \tilde{F})^{-1} P \tilde{K}, \quad \tilde{Z}_1 = \sqrt{-2 \text{Re}(\tau_1)} V_1, \\
V_k = V_{k-1} - (\tilde{\tau}_{k-1} + \tau_k) (E + \tau_k \tilde{F})^{-1} \tilde{F} V_{k-1}, \quad \tilde{Z}_k = [ \tilde{Z}_{k-1} \sqrt{-2 \text{Re}(\tau_k)} V_k ].
\]

The iterate \( V_k \) can also be written as

\[
V_k = (E + \tau_k \tilde{F})^{-1} (E - \tilde{\tau}_{k-1} \tilde{F}) V_{k-1} \\
= \cdots = (E + \tau_k \tilde{F})^{-1} \tilde{A}_{k-1} P \tilde{K} = (E + \tau_k \tilde{F})^{-1} W_{k-1}
\]

for \( k \geq 2 \). Then we have

\[
W_k = \tilde{A}_k P \tilde{K} = (E - \tilde{\tau}_{k-1} \tilde{F})(E + \tau_k \tilde{F})^{-1} \tilde{A}_{k-1} P \tilde{K} \\
= (E - \tilde{\tau}_{k-1} \tilde{F}) V_k = (E - \tilde{\tau}_{k-1} \tilde{F})(E + \tau_k \tilde{F})^{-1} W_{k-1} \\
= (I - 2 \text{Re}(\tau_k) \tilde{F})(E + \tau_k \tilde{F})^{-1} W_{k-1} = W_{k-1} - 2 \text{Re}(\tau_k) \tilde{F} V_k.
\]

Summarizing, we obtain the following algorithm for computing a low-rank approximate solution of the PALE (4.13).

**Algorithm 6.** The generalized LR-ADI for the projected Lyapunov equation.

**INPUT:** \( E, F \in \mathbb{R}^{n \times n}, K \in \mathbb{R}^{n \times g}, \) projector \( P \), shift parameters \( \tau_1, \ldots, \tau_q \in \mathbb{C}, \) and tolerance \( \text{tol} \).

**OUTPUT:** A low-rank approximation \( Z \approx \tilde{Z}_k \tilde{Z}_k^T \) to the solution of the PALE (4.13).

1. \( V_1 = (E + \tau_1 \tilde{F})^{-1} P \tilde{K}, \)
2. \( \tilde{Z}_1 = \sqrt{-2 \text{Re}(\tau_1)} V_1, \)
3. \( W_1 = P \tilde{K} - 2 \text{Re}(\tau_1) \tilde{F} V_1, \) \( k = 2, \)
4. WHILE ( \( \| W_{k-1} \|_F \| \tilde{K} P \tilde{K} - P \tilde{K} \|_F \geq \text{tol} \) AND \( k \leq k_{\text{max}} \))
   \[
   (a) \quad V_k = (E + \tau_k \tilde{F})^{-1} W_{k-1}, \\
   (b) \quad \tilde{Z}_k = [ \tilde{Z}_{k-1} \sqrt{-2 \text{Re}(\tau_k)} V_k ], \\
   (c) \quad W_k = W_{k-1} - 2 \text{Re}(\tau_k) \tilde{F} V_k, \\
   (d) \quad k \leftarrow k + 1.
   \]

END

At each iteration we have \( \tilde{Z}_k = [ \sqrt{-2 \text{Re}(\tau_1)} V_1, \ldots, \sqrt{-2 \text{Re}(\tau_q)} V_q ] \in \mathbb{R}^{n \times gk} \). To keep the low-rank structure in \( \tilde{Z}_k \) for large \( gk \), we can compress the columns of \( \tilde{Z}_k \) using the rank-revealing QR factorization as described in [11]. Furthermore, in order to guarantee the factors \( Z_k \) to be real in case of complex shift parameters, we take these parameters in complex conjugate pairs \( \{ \tau_k, \tau_{k+1} = \overline{\tau}_k \} \) and compute \( \tilde{Z}_k \) as in (4.18) if \( \tau_k \) is real and

\[
\tilde{Z}_k = [ \tilde{Z}_{k-1} \sqrt{-\text{Re}(\tau_k)} \tilde{V} ], \qquad \tilde{Z}_{k+1} = [ \tilde{Z}_k \sqrt{-\text{Re}(\tau_{k+1})} (V_k - \tau_{k+1} \tilde{V}) ]
\]

if \( \tau_k \) is complex,

\[
\tau_{k+1} = \overline{\tau}_k,
\]

where \( \tilde{V} = (E + \tau_{k+1} \tilde{F})^{-1} \tilde{F} V_k \); see [42] for details.
Finally, note that if the LR-ADI method is used in the inner Newton iteration, then we need to compute the products \((E + \tau_k \tilde{F})^{-1} w\) with some vector \(w \in \mathbb{R}^n\) and \(E + \tau_k \tilde{F} = (E + \tau_k F) + \tau_k K_j \tilde{GP}_r\) with low-rank matrices \(\tilde{GP}_r \in \mathbb{R}^{p \times n}\) and \(K_j \in \mathbb{R}^{n \times p}\) depending on the Lyapunov equation to be solved. For this purpose, we can use the Sherman–Morrison–Woodbury formula \([18, \text{section 2.1.3}]\)

\[
(E + \tau \tilde{F})^{-1} w = w_1 - M_{K_j} \left( (I_p + \tilde{GP}_r M_{K_j} )^{-1} \tilde{GP}_r w_1 \right),
\]

where \(w_1 = (E + \tau_k F)^{-1} w\) and \(M_{K_j} = \tau_k (E + \tau_k F)^{-1} K_j\) can be determined either by computing (sparse) LU factorizations and forward/backward substitutions or by using iterative Krylov subspace methods \([37]\).

### 4.4. Stopping criteria for Newton’s method.

The iterations in Algorithms 4 and 5 can be stopped as soon as the normalized residual norm satisfies the condition

\[
\varrho_R(\tilde{X}_j \tilde{X}_j^T) = \frac{\| \mathcal{R}(\tilde{X}_j \tilde{X}_j^T) \|}{\| P_I H \tilde{H}^T P_I^T \|} \leq \text{tol},
\]

where \(\| \cdot \|\) is the spectral or Frobenius matrix norm and \(\text{tol}\) is a user-defined tolerance, or a stagnation of the residual norms is observed. Unfortunately, the computation of \(\| \mathcal{R}(\tilde{X}_j \tilde{X}_j^T) \|\) by forming the residual matrix \(\mathcal{R}(\tilde{X}_j \tilde{X}_j^T)\) is memory-intensive for large-scale problems. If \(\tilde{X}_j\) has a small number of columns \(n_j\), then the residual norm can be determined efficiently using a factorization based approach proposed in \([9]\) for standard Riccati equations. In Algorithm 4, we have

\[
\mathcal{R}(\tilde{X}_j \tilde{X}_j^T) = \begin{bmatrix} P_I H & F \tilde{X}_j & E \tilde{X}_j \end{bmatrix} \begin{bmatrix} I_m & 0 & 0 \\ 0 & 0 & I_{n_j} \\ 0 & I_{n_j} & \tilde{X}_j^T \tilde{G}^T \tilde{G} \tilde{X}_j \end{bmatrix} \begin{bmatrix} \tilde{H}^T P_I^T \\ \tilde{X}_j^T F^T \\ \tilde{X}_j^T E^T \end{bmatrix}.
\]

Then computing an “economy-size” QR decomposition

\[
\left[ \begin{array}{c} P_I H \\ F \tilde{X}_j \\ E \tilde{X}_j \end{array} \right] = Q^{(j)} R^{(j)},
\]

where \(Q^{(j)}\) has orthonormal columns and \(R^{(j)}\) has full row rank, we find

\[
\| \mathcal{R}(\tilde{X}_j \tilde{X}_j^T) \| = \| R^{(j)} \begin{bmatrix} I_m & 0 & 0 \\ 0 & I_{n_j} & \tilde{X}_j^T \tilde{G}^T \tilde{G} \tilde{X}_j \end{bmatrix} (R^{(j)})^T \|.
\]

A similar procedure can also be applied to determine \(\| \mathcal{R}(\tilde{X}_{1,j} \tilde{X}_{1,j}^T - \tilde{X}_{2,j} \tilde{X}_{2,j}^T) \|\) in Algorithm 5. Thus, the evaluation of the residual norms reduces to the computation of the norm of much smaller matrices. Though this is much cheaper than computing the residual matrix explicitly, the verification of the stopping criterion (4.20) can still be much more expensive than computing \(\tilde{X}_j\) itself.

Another approach for computing the Riccati residuals in Algorithm 4 is based on the relation (4.1). If the PALE (4.2) is solved for \(N_j\) exactly, then (4.1) leads to

\[
\| \mathcal{R}(X_j) \| = \| EN_j - 1 \tilde{G}^T \tilde{G} N_j - 1 E^T \| = \| K_j K_j^T \| = \| K_j^T K_j \|.
\]
with a small matrix $K_j^T K_j \in \mathbb{R}^{p \times p}$. However, if we solve the PALE (4.2) approximately using the LR-ADI method, then

\begin{equation}
\mathcal{R}(\tilde{X}_j \tilde{X}_j^T) = K_j K_j^T + F_{j-1} \tilde{N}_{j-1} \tilde{N}_{j-1}^T E^T + E \tilde{N}_{j-1} \tilde{N}_{j-1}^T F_{j-1}^T + \mathcal{R}(\tilde{X}_{j-1} \tilde{X}_{j-1}^T)
\end{equation}

\[= \cdots = K_j K_j^T + L_{j-1}(\tilde{N}_{j-1} \tilde{N}_{j-1}^T) + \cdots + L_0(\tilde{N}_0 \tilde{N}_0^T),\]

where $L_k(\tilde{N}_k \tilde{N}_k^T) = F_k \tilde{N}_k \tilde{N}_k^T E^T + E \tilde{N}_k \tilde{N}_k^T F_k^T + P_k K_k K_k^T P_k^T$ is the Lyapunov residual with $K_0 = \hat{H}$, $F_0 = F$, $K_k = E \tilde{N}_{k-1} \tilde{N}_{k-1}^T \hat{G}^T$, and $F_k = F_{k-1} + K_k \hat{G} P_r$ for $k = 1, \ldots, j - 1$. Thus, the Riccati residual can be estimated as

$$
\|\mathcal{R}(\tilde{X}_j \tilde{X}_j^T)\| \leq \|K_j K_j^T\| + \|L_{j-1}(\tilde{N}_{j-1} \tilde{N}_{j-1}^T)\| + \cdots + \|L_0(\tilde{N}_0 \tilde{N}_0^T)\|.
$$

Note that (4.22) implies that the Lyapunov residuals accumulate during the Newton iteration. This means that the tolerance for the Lyapunov residuals should be taken smaller than the tolerance for the Riccati residuals.

In Algorithm 4, the Newton iteration can also be stopped as soon as the changes in $\tilde{X}_j$ become small, i.e., $\|\tilde{N}_j\|_F / \|\tilde{X}_j\|_F \leq tol$.

### 4.5. Comparison of the Newton and Newton–Kleinman methods.

We now compare the low-rank formulations of the Newton and Newton–Kleinman iterations with respect to complexity and numerical robustness.

Consider first the case when $\lambda E - F$ is stable. While in each Newton iteration in Algorithm 4, only one PALE (4.2) has to be solved, the Newton–Kleinman iteration in Algorithm 5 involves solving two PALEs (4.11) and (4.12) in each iteration plus one PALE (4.10) at the end. Since (4.11) and (4.12) differ in the right-hand side only, they can be solved simultaneously. If we solve the PALE

\begin{equation}
F_j Z_{j+1} E^T + E Z_{j+1} F_j^T = -P_l \hat{H}, K_j \hat{H}, K_j^T P_l^T, \quad Z_{j+1} = P_r Z_{j+1} P_r^T,
\end{equation}

for the low-rank Cholesky factor $\hat{Z}_{j+1}$ such that $Z_{j+1} \approx \hat{Z}_{j+1} \hat{Z}_{j+1}^T$ using the generalized LR-ADI method in Algorithm 6, then the low-rank Cholesky factors $\hat{X}_{1,j+1}$ and $\hat{X}_{2,j+1}$ can be extracted from

$$
\hat{Z}_{j+1} = \begin{bmatrix}
Z_{11}, & Z_{12}, & \ldots, & Z_{k1}, & Z_{k2}
\end{bmatrix}_m^p
$$

as $\hat{X}_{1,j+1} = [Z_{11}, \ldots, Z_{k1}]$ and $\hat{X}_{2,j+1} = [Z_{12}, \ldots, Z_{k2}]$. Since the right-hand side in the PALE (4.2) has smaller rank than that in the PALE (4.23), Algorithm 4 is less expensive than Algorithm 5. Moreover, the computation of the normalized residual in Algorithm 4 is much cheaper than that in Algorithm 5. Finally, as numerical experiments show, each Newton (outer) iteration usually requires fewer (inner) ADI iterations compared with the Newton–Kleinman step.

It should be noted, however, that Algorithm 4 can be used only if the pencil $\lambda E - F$ is stable, whereas Algorithm 5 can also be applied to unstable problems provided a stabilizing initial guess is available. Furthermore, the inexact version of the Newton method may be unstable. As shown in section 4.4, due to the approximate solution of the PALE (4.2) in Algorithm 4, the residuals accumulate over the iterations that may even cause the iteration to diverge.
5. Numerical examples. In this section, we present some results of numerical experiments to demonstrate the properties of the presented methods for solving the PARE \((1.2)\). As mentioned earlier, projected Riccati equations arise in balancing-related model reduction of the descriptor system

\[(5.1) \quad E\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t),\]

where \(E, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, C \in \mathbb{R}^{p \times n}\) and \(x \in \mathbb{R}^n\) is the state, \(u \in \mathbb{R}^m\) is the input, and \(y \in \mathbb{R}^p\) is the output. A transfer function of \((5.1)\) is given by \(G(s) = C(sE - A)^{-1}B\).

It can be additively decomposed as \(G(s) = G_{sp}(s) + G_0(s)\), where \(G_{sp}(s)\) is a strictly proper part of \(G(s)\) satisfying \(\lim_{s \to \infty} G_{sp}(s) = 0\) and \(G_0(s) = M_0 + sM_1 + \cdots + s^dM_d\) is a polynomial part. In order to compute a reduced-order model for \((5.1)\), we have to solve the PARE \((1.2)\) (and also its dual), where the matrix coefficients have one of the forms depending on the applied balanced truncation approach [36]:

- positive real balanced truncation
  \[(5.2) \quad E = E, \quad F = A - P_1B(M_0 + M_0^T)^{-1}CP_r, \quad G = C^T(M_0 + M_0^T)^{-1}C, \quad H = B(M_0 + M_0^T)^{-1}B^T;\]

- bounded real balanced truncation
  \[(5.3) \quad E = E, \quad F = A + P_1B(I - M_0^T M_0)^{-1}M_0^T CP_r, \quad G = C^T(I - M_0 M_0^T)^{-1}C, \quad H = B(I - M_0^T M_0)^{-1}B^T;\]

- bounded real balanced truncation via a Moebius transformation
  \[(5.4) \quad E = E, \quad F = A - BC - 2\hat{P}_1B(I - \hat{M}_0^T \hat{M}_0)^{-1}\hat{M}_0^T CP_r, \quad G = 2C^T(I - \hat{M}_0 \hat{M}_0^T)^{-1}C, \quad H = 2B(I - \hat{M}_0^T \hat{M}_0)^{-1}B^T, \quad P_r = \hat{P}_r, \quad P_l = \hat{P}_l,\]

where \(\hat{P}_r\) and \(\hat{P}_l\) are the spectral projectors onto the left and right deflating subspaces of \(\lambda E - A + BC\) corresponding to the finite eigenvalues and \(\hat{M}_0 = I - 2\lim_{s \to \infty} C(sE - A + BC)^{-1}B\).

Note that the PARE \((1.2)\) with \((5.4)\) is just the bounded real PARE of the Moebius-transformed system \(\hat{G}(s) = (I - G(s))(I + G(s))^{-1} = C(\hat{E} - \hat{A})^{-1}\hat{B} + I\) with \(\hat{E} = E, \quad A = A - BC, \quad \hat{B} = -\sqrt{2}B, \quad \text{and} \quad \hat{C} = \sqrt{2}C\).

Example 5.1. The first example is a three-port RC circuit. This circuit is modeled by a descriptor system of index 1 in modified nodal analysis form. It has \(n = 207\) state variables and \(m = 3\) inputs and outputs. Under some conditions on circuit topology guaranteeing that the transfer function \(G\) is positive real, i.e., \(G\) is analytic in \(\mathbb{C}_+\) and \(G(s) + G(s)^* \geq 0\) for all \(s \in \mathbb{C}_+\), one can show that the positive real PARE \((1.2), (5.2)\) (PR-PARE for short) and also the PARE \((1.2), (5.4)\) (BR(M)-PARE for short) are solvable. We compute the semistabilizing solutions of these equations using the Newton–Schur–Hammarling method as in Algorithm 3.

Figure 1(a) shows the normalized residual \(\|\mathcal{R}(\hat{X}\hat{X}^T)\|_F / \|P_1HP_1^T\|_F\). One can see that for both Riccati equations, the Newton iteration has a linear convergence only. This can be explained by the fact that \(\mathcal{R}_X^N\) is singular. In Figure 1(b), we present the condition number \(\kappa_2(E, F_j)\) of the operator \(\mathcal{R}_X^N\) or, equivalently, of the PALE \((3.1)\), which is defined as \(\kappa_2(E, F_j) = 2\|E\|_2\|F_j\|_2\|Z_j\|_2\), where \(Z_j\) solves the PALE.
Fig. 1. RC circuit: (a) the convergence history of the Newton–Schur–Hammarling method; (b) the condition number \( \kappa_{2}(E,F) \) of the operator \( R_{X_{j}} \).

Table 1

<table>
<thead>
<tr>
<th># outer</th>
<th>( g_{R}(X_{j}) )</th>
<th># inner</th>
<th>( g_{L}(Z_{j}Z_{T}^{T}) )</th>
<th># outer</th>
<th>( g_{R}(X_{j}) )</th>
<th># inner</th>
<th>( g_{L}(Z_{j}Z_{T}^{T}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.363e−02</td>
<td>27</td>
<td>9.920e−13</td>
<td>1</td>
<td>5.363e−02</td>
<td>27</td>
<td>9.920e−13</td>
</tr>
<tr>
<td>2</td>
<td>3.912e−04</td>
<td>23</td>
<td>7.289e−13</td>
<td>2</td>
<td>3.912e−04</td>
<td>25</td>
<td>4.472e−13</td>
</tr>
<tr>
<td>3</td>
<td>3.487e−08</td>
<td>20</td>
<td>8.043e−13</td>
<td>3</td>
<td>3.487e−08</td>
<td>25</td>
<td>4.457e−13</td>
</tr>
<tr>
<td>4</td>
<td>1.032e−12</td>
<td>20</td>
<td>4.238e−13</td>
<td>4</td>
<td>4.910e−13</td>
<td>25</td>
<td>4.457e−13</td>
</tr>
</tbody>
</table>

see [39]. As expected, in both cases, the condition number increases, as \( X_{j} \) approaches \( X_{*} \).

Example 5.2. Consider the two-dimensional instationary Stokes equation that describes the flow of an incompressible fluid in a domain. The spatial discretization of this equation by the finite volume method on a uniform staggered grid leads to the descriptor system (5.1) of index 2. The transfer function of this system is bounded real, i.e., \( G \) is analytic in \( \mathbb{C}_{+} \) and \( I - G(s)G(s)^{*} \geq 0 \) for all \( s \in \mathbb{C}_{+} \), which guarantees the solvability of the bounded real PARE (1.2), (5.3). We compute the low-rank approximations to the solution of this equation using the low-rank Newton (LR-N for short) and low-rank Newton–Kleinman (LR-NK for short) methods as in Algorithms 4 and 5, respectively, combined with the LR-ADI iteration. The inner ADI iterations have been stopped as soon as the normalized residuals for the PALE (4.13) satisfy (4.16) with \( tol = 10^{-13} \). In Table 1, we present the number \( j \) of outer Newton iterations (# outer), the normalized Riccati residuals \( g_{R}(X_{j}) \) given in (4.20), the number of the inner ADI iterations (# inner), and the reached normalized Lyapunov residuals \( g_{L}(Z_{j}Z_{T}^{T}) \) as in (4.16) for the low-rank Newton and Newton–Kleinman methods. The problem dimensions are \( n = 10679 \) and \( m = p = 5 \). Figure 2(a) shows the normalized residuals \( g_{R}(X_{j}) \), whereas in Figure 2(b) we present the number of ADI iterations for both methods.

Example 5.3. Consider a constrained damped mass-spring system from [32]. The vibration of this system is described by the single-input single-output descriptor system (5.1) which is of index 3 and has a bounded real transfer function. We compute the low-rank approximations to the solution of the bounded real PARE
6. Conclusions. In this paper, we have presented efficient and reliable numerical methods for solving projected Riccati equations as they arise in positive real and bounded real balanced truncation of descriptor systems. These methods are based on the Newton and Newton–Kleinman iterations. We have also considered the computation of the Cholesky factors and low-rank Cholesky factors of the stabilizing, positive semidefinite solutions of projected Riccati equations. The convergence analysis has been presented for Newton’s iteration. The numerical experiments for different types of descriptor systems and different forms of projected Riccati equations illustrate the properties of the presented numerical algorithms.
In [49], a quadratic ADI method has been proposed for standard Riccati equations. An extension of this method to projected Riccati equations remains for future work.

Acknowledgment. The authors would like to thank Jens Saak for helpful discussions.

REFERENCES


