THE PROJECTED NEWTON-KLEINMAN METHOD FOR THE ALGEBRAIC RICCATI EQUATION *

DAVIDE PALITTA†

Abstract. The numerical solution of the algebraic Riccati equation is a challenging task especially for very large problem dimensions. In this paper we present a new algorithm that combines the very appealing computational features of projection methods with the convergence properties of the inexact Newton-Kleinman procedure equipped with a line search. In particular, the Newton scheme is completely merged in a projection framework with a single approximation space. Moreover, the line search that guarantees the convergence of the inexact procedure turns out to be exact in our setting. Several numerical results are reported to illustrate the potential of our novel approach.

Key words. Riccati equation, Newton-Kleinman method, projection methods, large-scale matrix equations

AMS subject classifications. 65F30, 15A24, 49M15, 39B42, 40C05

1. Introduction.

We are interested in the numerical solution of the algebraic Riccati equation[1]

$$\mathcal{R}(X) := AX + XA^T - XBB^TX + C^TC = 0, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is of very large dimension, and $B \in \mathbb{R}^{n\times p}$ and $C \in \mathbb{R}^{q\times n}$ are such that $p+q \ll n$. This equation is of great interest in many applications, such as linear-quadratic optimal control problems for parabolic PDEs and balancing based model order reduction of large linear systems. See, e.g., [2, 37].

The solution X to (1.1) is usually dense and it cannot be stored in case of large scale problems. Under certain assumptions on the coefficient matrices, the singular values of the solution present a very fast decay and the matrix X can thus be well-approximated by a low rank matrix $SS^T \approx X$, $S \in \mathbb{R}^{n\times t}$, $t \ll n$, so that only the low-rank factor S needs to be computed and stored. See, e.g., [7].

Many efficient numerical methods for the solution of (1.1) have been developed in the last decades. For instance, the Newton-Kleinman method and many of its variants [10, 11, 14, 20, 28], projection methods [24, 27, 43, 45], subspace iteration methods [1, 7, 34] and the very recent RADI [9]. See also the survey article [8].

In this paper we focus on the inexact version of the Newton-Kleinman method where, at each iteration, a Lyapunov matrix equation needs to be solved. In standard implementations, these linear equations are tackled independently from each other so that the solution of the $(k+1)$-th Lyapunov equation does not exploit the information generated for computing the solution of the previous ones at all. Here we show how the solution of the $(k+1)$-th equation can actually profit from the computational efforts made to solve the first k ones. In particular, all the Lyapunov equations of the Newton-Kleinman scheme can be solved by employing the same approximation space and this observation leads to a remarkable speed-up of the entire algorithm maintaining the convergence properties of the Newton-Kleinman method.

The most common approximation spaces used in the solution of matrix equations by projection are the extended Krylov subspace

$$\mathbf{EK}^{\square}_m(A, C^T) := \text{Range}([C^T, AC^T, A^{-1}C^T, \dots, A^{m-1}C^T, A^{-m}C^T]), \tag{1.2}$$

see, e.g., [29, 42], and the more general rational Krylov subspace

$$\mathbf{K}^{\square}_m(A, C^T, \mathbf{s}) := \text{Range}([C^T, (A - s_2I)^{-1}C^T, \dots, \prod_{i=2}^{m}(A - s_iI)^{-1}C^T]), \tag{1.3}$$

where $\mathbf{s} = [s_2, \dots, s_m]^T \in \mathbb{C}^{m-1}$. See, e.g., [17–19]. We thus consider only these spaces in our analysis.

The following is a synopsis of the paper. In section 2 we revise the Newton-Kleinman method and its inexact variant presented in [11]. In section 3 we show that all the iterates computed by these algorithms lie

*Version of January 30, 2019.

†Research Group Computational Methods in Systems and Control Theory (CSC), Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany. palitta@mpi-magdeburg.mpg.de

[1]The Riccati equation is usually reported as $A^TX + XA - XBB^TX + C^TC = 0$. Here we prefer to use A in place of A^T for sake of simplicity in the presentation of the Krylov subspace approach.

on the same subspace whose definition depends on the choice of the initial guess X_0 in the Newton sequence. In particular, in section 3.1 we present the main result of the paper and the complete implementation of the new iterative procedure is illustrated in section 3.2. For the sake of simplicity, only the extended Krylov subspace (1.2) is considered in the discussion presented in section 3.1-3.2 but in section 3.3 we show how to easily adapt our new strategy when the rational Krylov subspace (1.3) is adopted as approximation space. In section 4 we further generalize the approach and we present its natural extension for the solution of generalized Riccati equations. In section 5 several numerical examples illustrate the effectiveness of the novel framework and our conclusions are given in section 6.

Throughout the paper we adopt the following notation. The matrix inner product is defined as $\langle X, Y\rangle_F = \text{trace}(Y^T X)$ so that the induced norm is $\|X\|_F^2 = \langle X, X\rangle_F$. The Kronecker product is denoted by $\otimes$ while I_n and $O_{n\times m}$ denote the identity matrix of order n and the $n \times m$ zero matrix respectively. Only one subscript is used for a square zero matrix, i.e., $O_{n\times n} = O_n$, and the subscript is omitted whenever the dimension of I and O is clear from the context. Moreover, E_i will denote the i-th block of ℓ columns of an identity matrix whose dimension depends on the adopted approximation space. More precisely, when the extended Krylov subspace (1.2) is employed, $\ell = 2q$ and $E_i \in \mathbb{R}^{2qm\times 2q}$ while $\ell = q$, $E_i \in \mathbb{R}^{qm\times q}$, when the rational Krylov subspace (1.3) is selected. The brackets $[\cdot]$ are used to concatenate matrices of conforming dimensions. In particular, a Matlab-like notation is adopted and $[M, N]$ denotes the matrix obtained by putting M and N one next to the other whereas $[M; N]$ the one obtained by stacking M and N one of top of each other, i.e., $[M; N] = [M^T, N^T]^T$. The notation $\text{diag}(M, N)$ is used to denote the block diagonal matrix with diagonal blocks M and N and we write $A < 0$ if the matrix A is negative definite, i.e., if its field of values $W(A) := \{\lambda \in \mathbb{C} \text{ s.t. } z^*(A - \lambda I)z = 0,\ z \in \mathbb{C}^n,\ \|z\|_F = 1\}$, z^* conjugate transpose of z, is contained in the left half plane $\mathbb{C}_-$.

Given a suitable space $\mathcal{K}_m$[2], we will always assume that a matrix $V_m \in \mathbb{R}^{n\times \ell}$, $\text{Range}(V_m) = \mathcal{K}_m$, has orthonormal columns and it is full rank so that $\dim(\mathcal{K}_m) = \ell$. Indeed, if this is not the case, deflation strategies to overcome the possible linear dependence of the basis vectors can be adopted as it is customary in block Krylov methods. See, e.g., [22, Section 8].

2. The (inexact) Newton-Kleinman method. In this section we recall the Newton-Kleinman method and its inexact counterpart for the solution of (1.1).

DEFINITION 2.1. *Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, and $C \in \mathbb{R}^{q\times n}$. The pair (A, B) is called stabilizable if there exists a feedback matrix $K \in \mathbb{R}^{n\times p}$ such that $A - KB^T$ is stable, i.e., all the eigenvalues of $A - KB^T$ lie on the left half complex plane $\mathbb{C}_-$. The pair (A, C) is called detectable if (A^T, C^T) is stabilizable.*

What follows is an assumption that will hold throughout the paper.

ASSUMPTION 2.2. *The coefficient matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, and $C \in \mathbb{R}^{q\times n}$ in (1.1) are such that (A, B) is stabilizable and (A, C) detectable.*

If Assumption 2.2 holds, there exists a unique symmetric positive semidefinite solution X to (1.1) which is also the unique stabilizing solution, i.e., X is such that the matrix $A - XBB^T$ is stable. See, e.g., [32].

For a given X_0 such that $A - X_0BB^T$ is stable[3], the $(k+1)$-th iteration of the Newton method is defined as

$$\mathcal{R}'[X](X_{k+1} - X_k) = -\mathcal{R}(X_k), \quad k \geqslant 0,$$

where $\mathcal{R}'[X]$ denotes the Fréchet derivative of $\mathcal{R}$ at a given symmetric X. For the Riccati operator we have

$$\mathcal{R}'[X](Y) = AY + YA^T - YBB^TX - XBB^TY = (A - XBB^T)Y + Y(A - XBB^T)^T,$$

and therefore the $(k+1)$-th iterate of the Newton-Kleinman method is given by the solution of the Lyapunov equation

$$(A - X_kBB^T)X_{k+1} + X_{k+1}(A - X_kBB^T)^T = -X_kBB^TX_k - C^TC. \tag{2.1}$$

If the Lyapunov equations (2.1) are solved exactly, the Newton-Kleinman method computes a sequence of symmetric positive semidefinite matrices $\{X_k\}_{k\geqslant 0}$ such that $X_k \geqslant X_{k+1}$ for any $k \geqslant 0$. Moreover, $\{X_k\}_{k\geqslant 0}$

[2] $\mathcal{K}_m$ as in (1.2) or (1.3).

[3] Such an X_0 exists thanks to Assumption 2.2.

converges to the stabilizing solution X with a convergence rate that is quadratic for k large enough. See, e.g., [28]. Moreover, Benner and Byers showed in [10] that a line search can significantly improve the performance of the Newton-Kleinman method during the first Newton steps.

In our setting, due to the large problem dimensions, equations (2.1) have to be iteratively solved by one of the many efficient methods for Lyapunov equations present in the literature like projection methods [18,42], low-rank ADI [13,33] or low-rank sign-function method [5,6]. See also [44] and the references therein.

The iterative solution of (2.1) introduces some inexactness in the Newton scheme, and this leads to the so-called inexact Newton-Kleinman method whose convergence has been proved in [20]. However, the conditions considered in [20] seem difficult to meet in practice and in [11] the authors showed that a specific line search guarantees the convergence of the inexact Newton-Kleinman method. In particular, given a symmetric X_k, $\alpha > 0$, $\eta_k \in (0,1)$, we want to compute a matrix Z_k such that

$$\|\mathcal{R}'[X_k](Z_k) + \mathcal{R}(X_k)\|_F \leqslant \eta_k \|\mathcal{R}(X_k)\|_F. \tag{2.2}$$

Then we define

$$X_{k+1} = X_k + \lambda_k Z_k, \tag{2.3}$$

where the step size $\lambda_k > 0$ is such that

$$\|\mathcal{R}(X_k + \lambda_k Z_k)\|_F \leqslant (1 - \lambda_k \alpha)\|\mathcal{R}(X_k)\|_F, \tag{2.4}$$

without λ_k being too small.

If $L_{k+1} := \mathcal{R}'[X_k](Z_k) + \mathcal{R}(X_k)$, equation (2.2) is equivalent to $\|L_{k+1}\|_F \leqslant \eta_k \|\mathcal{R}(X_k)\|_F$ and we can write

$$L_{k+1} = (A - X_k BB^T)(X_k + Z_k) + (X_k + Z_k)(A - X_k BB^T) + X_k BB^T X_k + C^T C.$$

This means that the matrix $\widetilde{X}_{k+1} := X_k + Z_k$ is the solution of the Lyapunov equation

$$(A - X_k BB^T)\widetilde{X}_{k+1} + \widetilde{X}_{k+1}(A - X_k BB^T)^T = -X_k BB^T X_k - C^T C + L_{k+1}. \tag{2.5}$$

Clearly the matrix L_{k+1} is never computed and the notation in (2.5) is used only to indicate that $\widetilde{X}_{k+1}$ is an inexact solution to equation (2.1) such that the residual norm $\|L_{k+1}\|$ satisfies (2.2). Once $\widetilde{X}_{k+1}$ is computed, we recover Z_k by $Z_k = \widetilde{X}_{k+1} - X_k$.

We now want to compute a step-size $\lambda_k > 0$ such that (2.4) holds and define X_{k+1} as in (2.3). The Riccati residual at $X_{k+1} = X_k + \lambda_k Z_k$ can be expressed as

$$\mathcal{R}(X_k + \lambda_k Z_k) = (1 - \lambda_k)\mathcal{R}(X_k) + \lambda_k L_{k+1} - \lambda_k^2 Z_k BB^T Z_k,$$

so that, if $\eta_k \leqslant \bar{\eta} < 1$ and $\alpha \in (0, 1 - \bar{\eta})$, the sufficient decrease condition (2.4) is satisfied for every $\lambda_k \in (0, (1 - \alpha - \bar{\eta}) \cdot \frac{\|\mathcal{R}(X_k)\|_F}{\|Z_k BB^T Z_k\|_F}]$.

In [11], two choices for the forcing parameter η_k and for the actual computation of the step size λ_k are proposed and in [11, Theorem 10] the authors showed the convergence of the inexact iterative scheme.

The Newton-Kleinman method can be formulated in different ways. For instance, in the exact setting, if X_1 is such that $(A - X_0 BB^T)X_1 + X_1(A - X_0 BB^T)^T + C^T C = 0$, then a matrix δX_k given by the solution of the Lyapunov equation,

$$(A - X_k BB^T)\delta X_k + \delta X_k (A - X_k BB^T)^T = \delta X_{k-1} BB^T \delta X_{k-1}, \quad k \geqslant 1, \quad \delta X_0 = X_1 - X_0, \tag{2.6}$$

can be computed; the next iterate of the Newton-Kleinman scheme is then defined as $X_{k+1} = X_k + \delta X_k$. See, e.g., [3]. In the recent literature, this reformulation has been shown to be very appealing when $C^T C$ is supposed to be a *hierarchical* matrix and not simply low-rank. See [31]. In our problem setting, the solution of (2.6) may be computationally advantageous if p is significantly smaller than q. However, we prefer to deal with equations of the form (2.1) as suggested in [11,20].

In Algorithm 2.1 the inexact Newton-Kleinman method is summarized.

The effectiveness of Algorithm 2.1 is strictly related to the efficiency of the Lyapunov solves in line 3. In [11], the low-rank ADI method is employed for solving equations (2.1) whereas in [45] numerical results

Algorithm 2.1 Inexact Newton-Kleinman method with line search.

input : $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{q\times n}$, $X_0 \in \mathbb{R}^{n\times n}$, s.t. $A - X_0BB^T$ is stable, $\epsilon > 0$, $\overline{\eta} \in (0,1)$, $\alpha \in (0, 1-\overline{\eta})$.
output: $X_k \in \mathbb{R}^{n\times n}$ approximate solution to (1.1).

for $k = 0, 1, \ldots,$ *till convergence* **do**
  if $\|\mathcal{R}(X_k)\|_F < \epsilon \cdot \|\mathcal{R}(X_0)\|_F$ **then**
1 **Stop** and return X_k
  end
2 Select $\eta_k \in (0, \overline{\eta})$
3 Compute $\widetilde{X}_{k+1}$ s.t.

$$(A - X_kBB^T)\widetilde{X}_{k+1} + \widetilde{X}_{k+1}(A - X_kBB^T)^T = -X_kBB^TX_k - C^TC + L_{k+1}$$

  where $\|L_{k+1}\|_F \leqslant \eta_k\|\mathcal{R}(X_k)\|_F$
4 Set $Z_k = \widetilde{X}_{k+1} - X_k$
5 Compute $\lambda_k > 0$ s.t. $\|\mathcal{R}(X_k + \lambda_k Z_k)\|_F \leqslant (1 - \lambda_k\alpha)\|\mathcal{R}(X_k)\|_F$
6 Set $X_{k+1} = X_k + \lambda_k Z_k$
end

are reported where the extended Krylov subspace method (EKSM) is used as inner solver. See [42] for an implementation of EKSM called K-PIK. In both cases the inexact Newton-Kleinman method is not competitive when compared to other methods as the Newton-Kleinman method with Galerkin acceleration [14], projection methods [45] and the very recent RADI [9]. See, e.g., [8]. As already mentioned, the main disadvantage of the inexact Newton-Kleinman scheme is that, at each iteration $k+1$, equation (2.1) is solved independently from the previous ones. For instance, if a projection method like EKSM is used as Lyapunov solver, a new subspace has to be computed from scratch at each Newton iteration. However, in the next section, we show that all the iterates computed by the Newton-Kleinman scheme (2.1) lie on the same space. This means that only one space needs to be constructed leading to remarkable reductions in the computational efforts.

3. A new iterative framework. In this section we show that all the iterates computed by the Newton-Kleinman method lie in the same subspace if a projection method is employed in the solution of the Lyapunov equations (2.1).

Given a Lyapunov equation $AW + WA^T + C^TC = 0$ where $C \in \mathbb{R}^{q\times n}$ is low-rank, projection methods compute an approximate solution of the form $W_m = V_mY_mV_m^T$ where the orthonormal columns of V_m are a basis of a suitable subspace $\mathcal{K}_m$, i.e., $\mathcal{K}_m = \text{Range}(V_m)$, and Y_m is a small square matrix computed, e.g., by imposing a Galerkin condition on the residual matrix $AW_m + W_mA^T + C^TC$. In Algorithm 3.1 the general framework of projection methods for Lyapunov equations is reported. See also [44] for more details.

To ensure the solvability of the projected problems in line 5 of Algorithm 3.1, the matrix A is usually supposed to be negative definite as this is a sufficient condition for having a stable $T_m = V_m^TAV_m$. However, also T_m stable is only a sufficient condition for the well-posedness of the projected equation in line 5 and projection methods work in practice even with a coefficient matrix A that is stable but not necessarily negative definite. See, e.g., [44, Section 5.2.1]. For the Lyapunov equations (2.1), it is easy to show that the coefficient matrices $A - X_kBB^T$ are stable for all $k \geqslant 0$ when the exact Newton-Kleinman method is employed. See, e.g., [28]. However, this is no longer straightforward in the inexact setting and in [11, Theorem 10] the authors have to assume that $A - X_kBB^T$ is stable for $k > k_0$.

In line 9, the matrix $\widehat{Y}$ denotes a low-rank factor of Y_m. If Y_m is not numerically low-rank, $\widehat{Y}$ amounts to its Cholesky factor and $t = \dim(\mathcal{K}_m)$.

The performance of projection methods mainly depends on the quality of the approximation space $\mathcal{K}_m$ employed. As already mentioned, one of the most popular choice is the block extended Krylov subspace (1.2) which leads to EKSM presented in [42] for the solution of large-scale Lyapunov equations.

If EKSM is employed as inner solver in the Newton-Kleinman method, at each Newton step (2.1) we have to build a new extended Krylov subspace $\mathbf{EK}_m^{\square}(A - X_kBB^T, [C^T, X_kB])$ and this is not feasible in practice. However, in the next section we show that all the iterates X_{k+1} in (2.1) belong to the same extended

Algorithm 3.1 Galerkin projection method for the Lyapunov matrix equation.

input : $A \in \mathbb{R}^{n\times n}$, A negative definite, $C \in \mathbb{R}^{q\times n}$, $\epsilon > 0$
output: $S_m \in \mathbb{R}^{n\times t}$, $t \leqslant \dim(\mathcal{K}_m)$, $S_m S_m^T = W_m \approx W$

Set $\beta = \|CC^T\|_F$
Perform economy-size QR of C^T, $C^T = V_1\boldsymbol{\gamma}$
for $m = 1, 2, \ldots,$ *till convergence,* **do**
  Compute next basis block $\mathcal{V}_{m+1}$ and set $V_{m+1} = [V_m, \mathcal{V}_{m+1}]$
  Update $T_m = V_m^T A V_m$
  Solve $T_m Y_m + Y_m T_m^T + E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T = 0, \quad E_1 \in \mathbb{R}^{\dim(\mathcal{K}_m)\times \ell}$
  Compute $\|R_m\|_F = \|A(V_m Y_m V_m^T) + (V_m Y_m V_m^T)A^T + C^T C\|_F$
  if $\|R_m\|_F/\beta < \epsilon$ **then**
    Break and go to **9**
  end
end
Compute the eigendecomposition of Y_m and retain $\widehat{Y} \in \mathbb{R}^{\dim(\mathcal{K}_m)\times t}$, $t \leqslant \dim(\mathcal{K}_m)$
Set $S_m = V_m \widehat{Y}$

Krylov subspace whose definition depends on the choice of the initial guess X_0. This means that only one approximation space needs to be constructed to compute all the necessary iterates of the Newton-Kleinman method. We first show this result supposing that the equations (2.1) are solved with very high accuracy, i.e., $\|L_{k+1}\|_F \approx \texttt{eps}$ in (2.5) for any k where eps denotes machine precision. Then we generalize the result to the case of inexact solves equipped with the necessary line search.

3.1. The extended Krylov subspace. If A in (1.1) is negative definite, we can choose the initial guess of the Newton-Kleinman method to be zero[4]. This means that the first equation to be solved in (2.1) is

$$AX_1 + X_1A^T = -C^TC,$$

and the extended Krylov subspace $\mathbf{EK}^{\square}_{m_1}(A, C^T)$ is constructed.

The solution computed by EKSM is of the form $X_1 = V_{m_1} Y_{m_1} V_{m_1}^T$, $V_{m_1} \in \mathbb{R}^{n\times 2qm_1}$, $\mathrm{Range}(V_{m_1}) = \mathbf{EK}^{\square}_{m_1}(A, C^T)$, $Y_{m_1} \in \mathbb{R}^{2qm_1\times 2qm_1}$. The second equation $(A - X_1BB^T)X_2 + X_2(A - X_1BB^T)^T = -X_1BB^TX_1 - C^TC$ can thus be rewritten as

$$(A - V_{m_1}\Theta_1 B^T)X_2 + X_2(A - V_{m_1}\Theta_1 B^T)^T = -(V_{m_1}\Theta_1)(V_{m_1}\Theta_1)^T - C^TC, \quad \Theta_1 := Y_{m_1}V_{m_1}^T B \in \mathbb{R}^{2qm_1\times p}.$$

Therefore, the second space to be constructed is $\mathbf{EK}^{\square}_{m_2}(A - V_{m_1}\Theta_1 B^T, [C^T, V_{m_1}\Theta_1])$.

In the following theorem we show that $\mathbf{EK}^{\square}_{m_2}(A - V_{m_1}\Theta_1 B^T, [C^T, V_{m_1}\Theta_1])$ is a subspace of $\mathbf{EK}^{\square}_{\overline{m}_2}(A, C^T)$ for a sufficiently large $\overline{m}_2$ and this happens also for the spaces related to all the other equations of the Newton-Kleinman scheme.

THEOREM 3.1. *Let $X_{k+1} = S_{m_{k+1}} S_{m_{k+1}}^T$ be the solution to (2.1) computed by EKSM. Suppose that also all the previous Lyapunov equations of the Newton-Kleinman scheme have been solved by means of EKSM as well. Then*

$$\mathit{Range}(S_{m_{k+1}}) \subset \mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T),$$

for a sufficiently large $\overline{m}_{k+1}$ and $\overline{m}_{k+1} \leqslant \sum_{j=1}^{k+1} m_j + 2$.

Proof. We are going to prove the statement by induction on k.

For $k = 0$, the equation $AX_1 + X_1A^T = -C^TC$ needs to be solved. By applying EKSM we obtain a solution of the form $X_1 = S_1S_1^T$ such that $\mathrm{Range}(S_1) \subset \mathbf{EK}^{\square}_{m_1}(A, C^T)$, and we can set $\overline{m}_1 = m_1 \leqslant m_1 + 2$.

We now assume that $\mathrm{Range}(S_k) \subset \mathbf{EK}^{\square}_{\overline{m}_k}(A, C^T)$, $\overline{m}_k \leqslant \sum_{j=1}^{k} m_j + 2$, for a certain $k > 0$ and we show the statement for $k + 1$. If $X_k = S_kS_k^T = V_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T$ where $V_{\overline{m}_k} \in \mathbb{R}^{n\times 2q\overline{m}_k}$ denotes the orthonormal basis of

[4]To have a well-defined Newton sequence $\{X_k\}_{k\geqslant 0}$ for $X_0 = O$ is sufficient to have a stable A. Here we suppose $A < 0$ in order to apply a projection method.

$\mathbf{EK}^{\square}_{\overline{m}_k}(A, C^T)$, the Lyapunov equation which defines X_{k+1} in (2.1) can be written as

$$(A - V_{\overline{m}_k}\Theta_k B^T)X_{k+1} + X_{k+1}(A - V_{\overline{m}_k}\Theta_k B^T)^T = -(V_{\overline{m}_k}\Theta_k)(V_{\overline{m}_k}\Theta_k)^T - C^T C, \tag{3.1}$$

where $\Theta_k := Y_{\overline{m}_k} V_{\overline{m}_k}^T B \in \mathbb{R}^{2q\overline{m}_k \times p}$. Then, the extended Krylov subspace to solve (3.1) has the form $\mathbf{EK}^{\square}_{m_{k+1}}(A - V_{\overline{m}_k}\Theta_k B^T, [C^T, V_{\overline{m}_k}\Theta_1])$ and we show that this is a subspace of $\mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T)$ for a sufficiently large $\overline{m}_{k+1}$ with $\overline{m}_{k+1} \leqslant \sum_{j=1}^{k+1} m_j + 2$.

A matrix $H \in \mathbb{R}^{n\times(p+q)}$ such that $\mathrm{Range}(H) \subset \mathbf{EK}^{\square}_{m_{k+1}}(A - V_{\overline{m}_k}\Theta_k B^T, [C^T, V_{\overline{m}_k}\Theta_k])$ can be written as

$$H = \sum_{j=0}^{m_{k+1}-1} (A - V_{\overline{m}_k}\Theta_k B^T)^j [C^T, V_{\overline{m}_k}\Theta_k]\xi_j + \sum_{j=1}^{m_{k+1}} (A - V_{\overline{m}_k}\Theta_k B^T)^{-j} [C^T, V_{\overline{m}_k}\Theta_k]\nu_j,$$

where $\xi_j, \nu_j \in \mathbb{R}^{(q+p)\times(q+p)}$.

We first focus on the polynomial part and show that

$$\mathrm{Range}\left(\sum_{j=0}^{m_{k+1}-1} (A - V_{\overline{m}_k}\Theta_k B^T)^j [C^T, V_{\overline{m}_k}\Theta_k]\xi_j\right) \subset \mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T),$$

by induction on j. For $j = 0$ we have $[C^T, V_{\overline{m}_k}\Theta_k]\xi_0$ whose range is clearly a subset of $\mathbf{EK}^{\square}_{\overline{m}_k}(A, C^T)$ as $\mathrm{Range}(C) \subset \mathbf{EK}^{\square}_{\overline{m}_k}(A, C^T)$ by definition and $V_{\overline{m}_k}$ is a basis of $\mathbf{EK}^{\square}_{\overline{m}_k}(A, C^T)$.

We now assume that

$$\mathrm{Range}((A - V_{\overline{m}_k}\Theta_k B^T)^j [C^T, V_{\overline{m}_k}\Theta_k]\xi_j) \subset \mathbf{EK}^{\square}_{\overline{m}_k+j}(A, C^T), \text{ for a certain } j > 0, \tag{3.2}$$

and we show that $\mathrm{Range}((A - V_{\overline{m}_k}\Theta_k B^T)^{j+1} [C^T, V_{\overline{m}_k}\Theta_k]\xi_{j+1}) \subset \mathbf{EK}^{\square}_{\overline{m}_k+j+1}(A, C^T)$. We have

$$\begin{aligned}(A - V_{\overline{m}_k}\Theta_k B^T)^{j+1} [C^T, V_{\overline{m}_k}\Theta_k]\xi_{j+1} &= (A - V_{\overline{m}_k}\Theta_k B^T)(A - V_{\overline{m}_k}\Theta_k B^T)^j [C^T, V_{\overline{m}_k}\Theta_k]\xi_{j+1} \\ &= A(A - V_{\overline{m}_k}\Theta_k B^T)^j [C^T, V_{\overline{m}_k}\Theta_k]\xi_{j+1} \\ &\quad - V_{\overline{m}_k}\left(\Theta_k B^T (A - V_{\overline{m}_k}\Theta_k B^T)^j [C^T, V_{\overline{m}_k}\Theta_k]\xi_{j+1}\right).\end{aligned} \tag{3.3}$$

Since (3.2) holds,

$$\mathrm{Range}(A(A - V_{\overline{m}_k}\Theta_k B^T)^j [C^T, V_{\overline{m}_k}\Theta_k]\xi_j) \subset A \cdot \mathbf{EK}^{\square}_{\overline{m}_k+j}(A, C^T) \subset \mathbf{EK}^{\square}_{\overline{m}_k+j+1}(A, C^T).$$

Moreover, the second term in the right-hand side of (3.3) is just a linear combination of the columns of $V_{\overline{m}_k}$ and its range is thus contained in $\mathbf{EK}^{\square}_{\overline{m}_k+j}(A, C^T)$. Therefore,

$$\mathrm{Range}\left(\sum_{j=0}^{m_{k+1}-1} (A - V_{\overline{m}_k}\Theta_k B^T)^j [C^T, V_{\overline{m}_k}\Theta_k]\xi_j\right) \subset \mathbf{EK}^{\square}_{\overline{m}_k+m_{k+1}-1}(A, C^T).$$

The exact same arguments together with the Sherman-Morrison-Woodbury (SMW) formula [21, Equation (2.1.4)],

$$(A - V_{\overline{m}_k}\Theta_k B^T)^{-1} = A^{-1} + A^{-1}V_{\overline{m}_k}\Theta_k(I - B^T A^{-1} V_{\overline{m}_k}\Theta_k)^{-1} B^T A^{-1} = A^{-1}(A + V_{\overline{m}_k}\Upsilon_k)A^{-1}, \tag{3.4}$$

where $\Upsilon_k := \Theta_k(I - B^T A^{-1} V_{\overline{m}_k}\Theta_k)^{-1} B^T \in \mathbb{R}^{2q\overline{m}_k \times p}$, show that

$$\mathrm{Range}\left(\sum_{j=1}^{m_{k+1}} (A + V_{\overline{m}_k}\Theta_k B^T)^{-j} [C^T, V_{\overline{m}_k}\Theta_k]\nu_j\right) \subset \mathbf{EK}^{\square}_{\overline{m}_k+m_{k+1}+2}(A, C^T).$$

Since $\mathbf{EK}^{\square}_{\overline{m}_k+m_{k+1}-1}(A, C^T) \subset \mathbf{EK}^{\square}_{\overline{m}_k+m_{k+1}+2}(A, C^T)$, we can define $\overline{m}_{k+1} = \overline{m}_k + m_{k+1} + 2$ and get the result. ∎

Theorem 3.1 says that we can use the same space $\mathbf{EK}^{\square}_m(A, C^T)$ for solving all the Lyapunov equations of the Newton-Kleinman scheme (2.1). Moreover, for the $(k+1)$-th Lyapunov equation we do not have to recompute the space from scratch but we can reuse the space already computed for the previous equations and just keep expanding it.

The idea of embedding approximation spaces related to different problems in one single, possibly larger, space is not new. For instance, in the context of the solution of shifted linear systems, it has been shown in [41] how the total number of iteration of (restarted) FOM applied to a sequence of shifted linear systems simultaneously is equal to the one achieved by applying (restarted) FOM to the single linear system with the slowest convergence rate. Similarly, in our context, the number of iteration needed to solve all the Lyapunov equations of the Newton-Kleinman scheme is equal to the one necessary to solve the last equation. The *feedback invariant* property of $\mathbf{EK}^{\square}_m(A, C^T)$ may be reminiscent of the shifted invariant feature of the Krylov subspace exploited in the solution of sequences of shifted linear systems. However, all the linear systems are solved at the same time in the latter problem setting while we have to solve the Lyapunov equations in (2.1) serially as the $(k+1)$-th equation depends on the solution of the previous one.

A result similar to the one stated in Theorem 3.1 can be shown also in the case of the inexact solves (2.5) equipped with the line search (2.3).

COROLLARY 3.2. *Let $\widetilde{X}_{k+1} = \widetilde{S}_{m_{k+1}}\widetilde{S}^T_{m_{k+1}}$ be the inexact solution to (2.1) computed by EKSM. Suppose that also all the previous Lyapunov equations of the Newton-Kleinman scheme have been solved inexactly by means of EKSM as well. Then $X_{k+1} = X_k + \lambda_k Z_k = P_{k+1}P^T_{k+1}$, P_{k+1} low-rank, is such that*

$$\mathit{Range}(P_{k+1}) \subset \mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T),$$

for a sufficiently large $\overline{m}_{k+1}$, $\overline{m}_{k+1} \leqslant \sum_{j=1}^{k+1} m_j + 2$.

Proof. We again prove the statement by induction on k. For $k = 1$, we compute the matrix $\widetilde{X}_1 = \widetilde{S}_{m_1}\widetilde{S}^T_{m_1}$, $\mathrm{Range}(\widetilde{S}_{m_1}) \subset \mathbf{EK}^{\square}_{m_1}(A, C^T)$, which is an inexact solution of the equation $AX + XA^T = -C^TC$. Then, since $X_0 = O$, we define the first iterate of the Newton sequence as $X_1 = \lambda_1\widetilde{X}_1$ so that X_1 can be written as $X_1 = P_1P_1^T$, $P_1 = \sqrt{\lambda_1}\widetilde{S}_{m_1}$ and $\mathrm{Range}(P_1) \subset \mathbf{EK}^{\square}_{m_1}(A, C^T)$; $\overline{m}_1 = m_1 \leqslant m_1 + 2$.

We now suppose that the statement has been proven for a certain $k > 1$ and we show it for $k+1$. Let $\widetilde{X}_{k+1} = \widetilde{S}_{m_{k+1}}\widetilde{S}^T_{m_{k+1}}$ be the approximate solution of the equation $(A - X_kBB^T)X + X(A - X_kBB^T)^T = -X_kBB^TX_k - C^TC$ computed by the EKSM. Since $X_k = P_kP_k^T$ is such that $\mathrm{Range}(P_k) \subset \mathbf{EK}^{\square}_{\overline{m}_k}(A, C^T)$, $\overline{m}_k \leqslant \sum_{j=1}^{k} m_j + 2$, by inductive hypothesis, with the same argument of Theorem 3.1 we can show that $\mathrm{Range}(\widetilde{S}_{m_{k+1}}) \subset \mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T)$ for a sufficiently large $\overline{m}_{k+1}$, $\overline{m}_{k+1} \leqslant \sum_{j=1}^{k+1} m_j + 2$. Then, following section 2 we define $Z_k = \widetilde{X}_{k+1} - X_k$, so that the $(k+1)$-th iterate of the Newton sequence is

$$\begin{aligned} X_{k+1} &= X_k + \lambda_k Z_k = (1-\lambda_k)X_k + \lambda_k\widetilde{X}_{k+1} = (1-\lambda_k)P_kP_k^T + \lambda_k\widetilde{S}_{m_{k+1}}\widetilde{S}^T_{m_{k+1}} \\ &= [(1-\lambda_k)P_k, \lambda_k\widetilde{S}_{m_{k+1}}][P_k, \widetilde{S}_{m_{k+1}}]^T = P_{k+1}P^T_{k+1}, \end{aligned}$$

where $P_{k+1} \coloneqq [\sqrt{1-\lambda_k}P_k, \sqrt{\lambda_k}\widetilde{S}_{m_{k+1}}]$ is such that $\mathrm{Range}(P_{k+1}) \subset \mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T)$, $\overline{m}_{k+1} \leqslant \sum_{j=1}^{k+1} m_j + 2$, as $\mathrm{Range}(P_k) \subset \mathbf{EK}^{\square}_{\overline{m}_k}(A, C^T) \subset \mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T)$. ∎

The estimate on the number of iterations $\overline{m}_{k+1}$ given in Theorem 3.1 and Corollary 3.2 is very rough and it is provided only for showing that the dimension of $\mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T)$ is bounded by a constant which is smaller than n if m_j is moderate for $j = 1, \ldots, k+1$.

For a given tolerance ϵ, the actual dimension of $\mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T)$ to achieve $\|(A - X_kBB^T)X_{k+1} + X_{k+1}(A - X_kBB^T)^T + X_kBB^TX_k + C^TC\|_F \leqslant \epsilon$ is in general much smaller than $2q\left(\sum_{j=1}^{k+1} m_j + 2\right)$. See, e.g., Example 5.1.

3.2. Implementation details. In this section we present how to fully exploit Theorem 3.1 and Corollary 3.2 by merging the (inexact) Newton-Kleinman method in a projection procedure. Also here we present our strategy by first assuming $\|L_{k+1}\|_F \approx \texttt{eps}$ in (2.5) and then we generalize the approach to the case of inexact solves with line search.

We start by solving the equation $AX_1 + X_1A^T = -C^TC$ by projection onto the extended Krylov subspace $\mathbf{EK}^{\square}_{m_1}(A, C^T) = \mathrm{Range}(V_{m_1})$. As outlined in Algorithm 3.1, if $T_{m_1} \coloneqq V^T_{m_1}AV_{m_1}$, at each iteration of EKSM we have to solve the projected equation

$$T_{m_1}Y + YT^T_{m_1} + E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^TE_1^T = 0, \tag{3.5}$$

where $E_1 \in \mathbb{R}^{2qm_1 \times 2q}$ and $\boldsymbol{\gamma} \in \mathbb{R}^{2q\times q}$, $C^T = V_1\boldsymbol{\gamma}$. Since equation (3.5) is of small dimension, decomposition based methods as the Bartels-Stewart method [4] or the Hammarling method [23] can be employed for its solution.

If at iteration $\overline{m}_1$ the Lyapunov residual norm $\|R_{\overline{m}_1}\|_F = \|A(V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^T) + (V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^T)A^T + C^TC\|_F$ is sufficiently small, we define $Y_{\overline{m}_1} := Y$ and check the residual norm of the Riccati equation, namely $\|\mathcal{R}(V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^T)\|_F$. If this is sufficiently small we have completed the procedure and $V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^T$ is the sought approximated solution to (1.1), otherwise we pass to solve the second equation of the Newton scheme. We can write

$$(A - V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^TBB^T)X_2 + X_2(A - V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^TBB^T)^T = -V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^TBB^TV_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^T - C^TC, \tag{3.6}$$

and Theorem 3.1 says that $\mathbf{EK}^{\square}_{\overline{m}_2}(A, C^T)$, $\overline{m}_2 \geqslant \overline{m}_1$, is still a good approximation space for solving it. We thus start by projecting (3.6) onto the already computed space $\mathbf{EK}^{\square}_{\overline{m}_1}(A, C^T)$ getting

$$(T_{\overline{m}_1} - Y_{\overline{m}_1}B_{\overline{m}_1}B_{\overline{m}_1}^T)Y + Y(T_{\overline{m}_1} - Y_{\overline{m}_1}B_{\overline{m}_1}B_{\overline{m}_1}^T)^T = -Y_{\overline{m}_1}B_{\overline{m}_1}B_{\overline{m}_1}^TY_{\overline{m}_1} - E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^TE_1^T, \tag{3.7}$$

where $B_{\overline{m}_1} = V_{\overline{m}_1}^TB$. Notice that $B_{\overline{m}_1}$ can be computed on the fly performing $2q$ inner products per iteration.

It may happen that $\mathbf{EK}^{\square}_{\overline{m}_1}(A, C^T)$ is already a good approximation space for equation (3.6), that is, the Lyapunov residual norm $\|R_{\overline{m}_2}\|_F = \|(A - V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^TBB^T)V_{\overline{m}_1}YV_{\overline{m}_1}^T + V_{\overline{m}_1}YV_{\overline{m}_1}^T(A - V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^TBB^T)^T + V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^TBB^TV_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^T + C^TC\|_F$, where Y is the solution to (3.7), is sufficiently small. If this is the case we set $\overline{m}_2 = \overline{m}_1$, $Y_{\overline{m}_2} := Y$ and check $\|\mathcal{R}(V_{\overline{m}_2}Y_{\overline{m}_2}V_{\overline{m}_2}^T)\|_F$. Otherwise, we expand the space computing the next basis block $\mathcal{V}_{\overline{m}_1+1} \in \mathbb{R}^{n\times 2q}$ such that $V_{\overline{m}_1+1} := [V_{\overline{m}_1}, \mathcal{V}_{\overline{m}_1+1}]$ has orthonormal columns and $\mathrm{Range}(V_{\overline{m}_1+1}) = \mathbf{EK}^{\square}_{\overline{m}_1+1}(A, C^T)$.

In the next proposition we show how to easily compute the projection of the current Lyapunov equation once the subspace has been expanded and a cheap computation of the residual norm.

PROPOSITION 3.1. *Let $X_k = V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T$, $\mathrm{Range}(V_{\overline{m}_k}) = \mathbf{EK}^{\square}_{\overline{m}_k}(A, C^T)$ be the solution of the k-th Lyapunov equation of the Newton-Kleinman scheme. Then, the projection of the $(k+1)$-th equation (2.1) onto $\mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T) = \mathrm{Range}(V_{\overline{m}_{k+1}})$, $\overline{m}_{k+1} \geqslant \overline{m}_k$, is given by*

$$Q_{\overline{m}_{k+1}}Y + YQ_{\overline{m}_{k+1}}^T = -\mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})B_{\overline{m}_{k+1}}B_{\overline{m}_{k+1}}^T\,\mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) - E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^TE_1^T, \tag{3.8}$$

where $Q_{\overline{m}_{k+1}} = T_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})B_{\overline{m}_{k+1}}B_{\overline{m}_{k+1}}^T$, $T_{\overline{m}_{k+1}} = V_{\overline{m}_{k+1}}^TAV_{\overline{m}_{k+1}}$, $B_{\overline{m}_{k+1}} = V_{\overline{m}_{k+1}}^TB$ and $C^T = V_1\boldsymbol{\gamma}$. Moreover, the solution $Y_{\overline{m}_{k+1}}$ to (3.8) is such that

$$\begin{aligned}\|(A - X_kBB^T)(V_{\overline{m}_{k+1}}Y_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T) + (V_{\overline{m}_{k+1}}Y_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T)(A - X_kBB^T)^T + X_kBB^TX_k + C^TC\|_F = \\ = \sqrt{2}\|Y_{\overline{m}_{k+1}}\underline{T}_{\overline{m}_{k+1}}^TE_{\overline{m}_{k+1}+1}\|_F,\end{aligned} \tag{3.9}$$

where $\underline{T}_{\overline{m}_{k+1}} = V_{\overline{m}_{k+1}+1}^TAV_{\overline{m}_{k+1}}$.

Proof. We show the statements by induction on k. Since we suppose A negative definite, for $k = 0$ we have $X_0 = O$ and denoting by $Y_{\overline{m}_0} := V_1^TX_0V_1 = O_{2q}$, $\mathrm{Range}(V_1) = \mathbf{EK}^{\square}_1(A, C^T)$, we can write the projection of $AX + XA^T = -C^TC$ onto $\mathbf{EK}^{\square}_{\overline{m}_1}(A, C^T)$, $m_1 \geqslant 1$, as

$$Q_{\overline{m}_1}Y + YQ_{\overline{m}_1}^T = \; = -\mathrm{diag}(Y_{\overline{m}_0}, O_{2q(\overline{m}_1-1)})B_{\overline{m}_1}B_{\overline{m}_1}^T\,\mathrm{diag}(Y_{\overline{m}_0}, O_{2q(\overline{m}_1-1)}) - E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^TE_1^T, \tag{3.10}$$

with $Q_{\overline{m}_1} = T_{\overline{m}_1} - \mathrm{diag}(Y_{\overline{m}_0}, O_{2q(\overline{m}_1-1)})B_{\overline{m}_1}B_{\overline{m}_1}^T$. Moreover,

$$\|A(V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^T) + (V_{\overline{m}_1}Y_{\overline{m}_1}V_{\overline{m}_1}^T)A^T + C^TC\|_F = \sqrt{2}\|Y_{\overline{m}_1}\underline{T}_{\overline{m}_1}^TE_{\overline{m}_1+1}\|_F,$$

where $Y_{\overline{m}_1}$ denotes the solution to (3.10). See, e.g., [26].

We now suppose that the statements hold for a certain $k > 0$ and we show them for $k + 1$. If $X_k = V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T$, $\mathrm{Range}(V_{\overline{m}_k}) = \mathbf{EK}^{\square}_{\overline{m}_k}(A, C^T)$, then we can write the $(k+1)$-th equation of the Newton scheme as

$$(A - V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^TBB^T)X + X(A - V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^TBB^T)^T = -V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^TBB^TV_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T - C^TC. \tag{3.11}$$

If $\mathrm{Range}(V_{\overline{m}_{k+1}}) = \mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T)$ for $\overline{m}_{k+1} \geqslant \overline{m}_k$, then $V_{\overline{m}_{k+1}}^T V_{\overline{m}_k} = [I_{2q\overline{m}_k}; O_{2q(\overline{m}_{k+1}-\overline{m}_k)\times 2qm_k}]$. Therefore, pre and post multiplying equation (3.11) by $V_{\overline{m}_{k+1}}^T$ and $V_{\overline{m}_{k+1}}$ respectively, we get

$$\begin{aligned}(T_{\overline{m}_{k+1}} - [Y_{\overline{m}_k}; O_{2q(\overline{m}_{k+1}-\overline{m}_k)\times 2qm_k}]V_{\overline{m}_k}^T BB_{\overline{m}_{k+1}}^T)Y + Y(T_{\overline{m}_{k+1}} - [Y_{\overline{m}_k}; O_{2q(\overline{m}_{k+1}-\overline{m}_k)\times 2qm_k}]V_{\overline{m}_k}^T BB_{\overline{m}_{k+1}}^T)^T = \\ = -[Y_{\overline{m}_k}; O_{2q(\overline{m}_{k+1}-\overline{m}_k)\times 2qm_k}]V_{\overline{m}_k}^T BB^T V_{\overline{m}_k}[Y_{\overline{m}_k}; O_{2q(\overline{m}_{k+1}-\overline{m}_k)\times 2qm_k}]^T - E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T,\end{aligned}$$

and noticing that $[Y_{\overline{m}_k}; O_{(2q\overline{m}_{k+1}-\overline{m}_k)\times 2qm_k}]V_{\overline{m}_k}^T = \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})V_{\overline{m}_{k+1}}^T$ we have the result.

In conclusion, if $Y_{\overline{m}_{k+1}}$ denotes the solution to (3.8), it is easy to show that the residual norm of the Lyapunov equation $(A - X_kBB^T)(V_{\overline{m}_{k+1}}Y_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T) + (V_{\overline{m}_{k+1}}Y_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T)(A - X_kBB^T)^T = -X_kBB^TX_k - C^TC$ can be computed very cheaply as in (3.9). The proof follows the same line of the proof in [42, Proposition 3.3] recalling that $X_k = V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T = V_{\overline{m}_{k+1}}\mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})V_{\overline{m}_{k+1}}^T$. ∎

A similar result can be shown also in case of the inexact Newton-Kleinman method. Indeed, at the $(k+1)$-th iteration, we define X_{k+1} as in (2.5) where $Z_k = \widetilde{X}_{k+1} - X_k$ and $\widetilde{X}_{k+1}$ is the inexact solution to (2.1). Assuming $X_k = V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T$, $\mathrm{Range}(V_{\overline{m}_k}) = \mathbf{EK}^{\square}_{\overline{m}_k}(A, C^T)$, the projection of

$$(A - X_kBB^T)X + X(A - X_kBB^T)^T = -X_kBB^TX_k - C^TC,$$

onto $\mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, C^T) = \mathrm{Range}(V_{\overline{m}_{k+1}})$, $\overline{m}_{k+1} \geqslant \overline{m}_k$, is still of the form (3.8). This means that the residual norm can be still computed as in (3.9). Once $\|L_{k+1}\|_F = \sqrt{2}\|\widetilde{Y}_{\overline{m}_{k+1}}\underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}\|_F \leqslant \eta_k\|\mathcal{R}(X_k)\|_F$, where $\widetilde{Y}_{\overline{m}_{k+1}}$ is the solution to (3.8), we define $\widetilde{X}_{k+1} := V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T$, and we have

$$\begin{aligned}X_{k+1} &= X_k + \lambda_k Z_k = (1-\lambda_k)X_k + \lambda_k\widetilde{X}_{k+1} \\ &= (1-\lambda_k)V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T + \lambda_k V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T \\ &= (1-\lambda_k)V_{\overline{m}_{k+1}}\mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})V_{\overline{m}_{k+1}}^T + \lambda_k V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T \\ &= V_{\overline{m}_{k+1}}\left((1-\lambda_k)\cdot\mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) + \lambda_k\widetilde{Y}_{\overline{m}_{k+1}}\right)V_{\overline{m}_{k+1}}^T \\ &= V_{\overline{m}_{k+1}}Y_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T,\end{aligned}$$

where $Y_{\overline{m}_{k+1}} := (1-\lambda_k)\cdot\mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) + \lambda_k\widetilde{Y}_{\overline{m}_{k+1}}$.

We are thus left with showing that the line search, i.e., the computation of the λ_k's, can be cheaply carried out on the current subspace with no need to go back to $\mathbb{R}^n$. In [11], the authors show that $\|\mathcal{R}(X_k + \lambda Z_k)\|_F^2$ is a quartic polynomial in λ of the form

$$p_k(\lambda) = \|\mathcal{R}(X_k + \lambda Z_k)\|_F^2 = (1-\lambda)^2\alpha_k + \lambda^2\beta_k + \lambda^4\delta_k + 2\lambda(1-\lambda)\gamma_k - 2\lambda^2(1-\lambda)\epsilon_k - 2\lambda^3\zeta_k, \tag{3.12}$$

where

$$\begin{aligned}&\alpha_k = \|\mathcal{R}(X_k)\|_F^2, && \delta_k = \|Z_kBB^TZ_k\|_F^2, \\ &\beta_k = \|L_{k+1}\|_F^2, && \epsilon_k = \langle\mathcal{R}(X_k), Z_kBB^TZ_k\rangle_F, \\ &\gamma_k = \langle\mathcal{R}(X_k), L_{k+1}\rangle_F, && \zeta_k = \langle L_{k+1}, Z_kBB^TZ_k\rangle_F.\end{aligned} \tag{3.13}$$

If $\|L_{k+1}\|_F = 0$, the polynomial in (3.12) has a local minimizer $\lambda_k \in (0, 2]$ such that, if $A - X_kBB^T$ is stable and X_{k+1} is computed by using such a λ_k, also $A - X_{k+1}BB^T$ is stable. See [10]. However, in [11] the authors state that, in general, this no longer holds if $\|L_{k+1}\|_F \neq 0$. Nevertheless, we show that, in our particular framework, $p_k(\lambda)$ still has a local minimizer in $(0, 2]$ and we can thus compute the step-size as

$$\lambda_k = \operatorname*{argmin}_{(0,2]} p_k(\lambda). \tag{3.14}$$

We first derive new expressions for the coefficients (3.13) that will help us to prove the existence of a local minimizer of $p_k(\lambda)$ in $(0, 2]$.

PROPOSITION 3.2. *The coefficients in* (3.13) *are such that*

$$\alpha_k = \|T_{\overline{m}_k} Y_{\overline{m}_k} + Y_{\overline{m}_k} T_{\overline{m}_k}^T - Y_{\overline{m}_k} B_{\overline{m}_k} B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1 \boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T\|_F^2 + 2\|Y_{\overline{m}_k} \underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}\|_F^2,$$

$$\beta_k = 2\|\widetilde{Y}_{\overline{m}_{k+1}} \underline{T}_{\overline{m}_{k+1}} E_{\overline{m}_{k+1}+1}\|_F^2,$$

$$\gamma_k = \begin{cases} \langle E_{\overline{m}_k+1}^T \underline{T}_{\overline{m}_k} Y_{\overline{m}_k} + Y_{\overline{m}_k} \underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}, E_{\overline{m}_{k+1}+1}^T \underline{T}_{\overline{m}_{k+1}} \widetilde{Y}_{\overline{m}_{k+1}} \widetilde{Y}_{\overline{m}_{k+1}} \underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1} \rangle_F, & \text{if } \overline{m}_{k+1} = \overline{m}_k, \\ 0, \quad \text{if } \overline{m}_{k+1} > \overline{m}_k, \end{cases}$$

$$\delta_k = \| \left(\widetilde{Y}_{\overline{m}_{k+1}} - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) \right) B_{\overline{m}_{k+1}} B_{\overline{m}_{k+1}}^T \left(\widetilde{Y}_{\overline{m}_{k+1}} - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) \right) \|_F^2,$$

$$\epsilon_k = \begin{cases} \langle T_{\overline{m}_k} Y_{\overline{m}_k} + Y_{\overline{m}_k} T_{\overline{m}_k}^T - Y_{\overline{m}_k} B_{\overline{m}_k} B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1 \boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T, \\ \left(\widetilde{Y}_{\overline{m}_{k+1}} - Y_{\overline{m}_k} \right) B_{\overline{m}_{k+1}} B_{\overline{m}_{k+1}}^T \left(\widetilde{Y}_{\overline{m}_{k+1}} - Y_{\overline{m}_k} \right) \rangle_F, & \text{if } \overline{m}_{k+1} = \overline{m}_k, \\ \\ \langle T_{\overline{m}_k} Y_{\overline{m}_k} + Y_{\overline{m}_k} T_{\overline{m}_k}^T - Y_{\overline{m}_k} B_{\overline{m}_k} B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1 \boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T, \\ [I_{2q\overline{m}_k}, O_{2q\overline{m}_k \times 2q(\overline{m}_{k+1}-\overline{m}_k)}] \left(\widetilde{Y}_{\overline{m}_{k+1}} - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) \right) B_{\overline{m}_{k+1}} B_{\overline{m}_{k+1}}^T \left(\widetilde{Y}_{\overline{m}_{k+1}} \right. \\ \left. - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) \right) [I_{2q\overline{m}_k}; O_{2q(\overline{m}_{k+1}-\overline{m}_k)\times 2q\overline{m}_k}] \rangle_F \\ + \langle E_{\overline{m}_k+1} E_{\overline{m}_k+1}^T \underline{T}_{\overline{m}_k} [Y_{\overline{m}_k}, O_{2q\overline{m}_k \times 2q}] + [Y_{\overline{m}_k}; O_{2q\times 2q\overline{m}_k}] \underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1} E_{\overline{m}_k+1}^T, & \text{if } \overline{m}_{k+1} > \overline{m}_k, \\ [I_{2q(\overline{m}_k+1)}, O_{2q(\overline{m}_k+1)\times 2q(\overline{m}_{k+1}-\overline{m}_k-1)}] \left(\widetilde{Y}_{\overline{m}_{k+1}} - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) \right) B_{\overline{m}_{k+1}} B_{\overline{m}_{k+1}}^T \left(\widetilde{Y}_{\overline{m}_{k+1}} \right. \\ \left. - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) \right) [I_{2q(\overline{m}_k+1)}; O_{2q(\overline{m}_{k+1}-\overline{m}_k-1)\times 2q(\overline{m}_k+1)}] \rangle_F. \end{cases}$$

and

$$\zeta_k = 0.$$

See the Appendix for the proof.

Proposition 3.2 shows how only matrices of size (at most) $2q(\overline{m}_{k+1}+1)$ are actually involved in the computation of the coefficients (3.13) and we just need information that are available in the current subspace $\mathbf{EK}_{\overline{m}_{k+1}}^{\square}(A, C^T)$ to define $p_k(\lambda)$ without any backward transformations to $\mathbb{R}^n$.

By exploiting the expressions in Proposition 3.2 we are now able to show the existence of a local minimizer $\lambda_k \in (0,2]$ of $p_k(\lambda)$.

PROPOSITION 3.3. *If $\overline{m}_{k+1} > \overline{m}_k$, the polynomial $p_k(\lambda)$ has a local minimizer $\lambda_k \in (0,2]$.*

Proof. If $\overline{m}_{k+1} > \overline{m}_k$, by exploiting the expressions is Proposition 3.2, the polynomial $p_k(\lambda)$ in (3.12) can be written as

$$p_k(\lambda) = (1-\lambda)^2 \alpha_k + \lambda^2 \beta_k + \lambda^4 \delta_k - 2\lambda^2 (1-\lambda) \epsilon_k,$$

whose first derivative is

$$p_k'(\lambda) = -2(1-\lambda)\alpha_k + 2\lambda\beta_k + 4\lambda^3 \delta_k - 4\lambda\epsilon_k + 6\lambda^2 \epsilon_k.$$

Therefore, $p_k'(0) = -2\alpha_k < 0$ as $\alpha_k = \|\mathcal{R}(X_k)\|_F^2 > 0$. Notice that if $\alpha_k = 0$, this means that X_k is the exact solution to (1.1) and we do not need to compute any step-size λ_k.

Moreover,

$$p_k'(2) = 2\alpha_k + 4\beta_k + 32\delta_k + 16\epsilon_k \geqslant 2\alpha_k + 32\delta_k + 16\epsilon_k = 2\|\mathcal{R}(X_k) + 4Z_k B B^T Z_k\|_F^2,$$

as $\beta = \|L_{k+1}\|_F^2 \geqslant 0$. Since $\|\mathcal{R}(X_k) + 4Z_k B B^T Z_k\|_F^2 \geqslant 0$, also $p_k'(2) \geqslant 0$ and there exists a local minimizer λ_k of p_k in $(0,2]$. ∎

In our numerical experience it is very rare to have $\overline{m}_{k+1} = \overline{m}_k$. Indeed, it is unlikely that the space used for solving the k-th Lyapunov equation in the Newton sequence contains enough spectral information to solve also the $(k+1)$-th one. This may happen for the very first couple of Lyapunov equations, i.e., it has happened that $\overline{m}_2 = \overline{m}_1$. Since for the subsequent equations we have to expand the space anyway, we suggest to perform an extra iteration when $\overline{m}_{k+1} = \overline{m}_k$, so that $\overline{m}_{k+1} > \overline{m}_k$, and then compute λ_k as in (3.14).

Once λ_k is computed, defining $Y_{\overline{m}_{k+1}} := (1-\lambda_k)\cdot \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) + \lambda_k \widetilde{Y}_{\overline{m}_{k+1}}$, we can cheaply evaluate the residual norm of the new approximate solution $X_{k+1} := V_{\overline{m}_{k+1}} Y_{\overline{m}_{k+1}} V_{\overline{m}_{k+1}}^T$ to the Riccati equation by

$$\begin{aligned}\|\mathcal{R}(X_{k+1})\|_F^2 =& \|\mathcal{R}(V_{\overline{m}_{k+1}} Y_{\overline{m}_{k+1}} V_{\overline{m}_{k+1}}^T)\|_F^2 \\ =& \|T_{\overline{m}_{k+1}} Y_{\overline{m}_{k+1}} + Y_{\overline{m}_{k+1}} T_{\overline{m}_{k+1}}^T - Y_{\overline{m}_{k+1}} B_{\overline{m}_{k+1}} B_{\overline{m}_{k+1}}^T Y_{\overline{m}_{k+1}} + E_1 \boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T\|_F^2 + 2\|Y_{\overline{m}_{k+1}} \underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}\|_F^2. \end{aligned} \tag{3.15}$$

Moreover, this value can be used in the next iteration as α_{k+1} if necessary.

The complete implementation of our new iterative framework is summarized in Algorithm 3.2[5] where the residual norms of the Riccati operator $\|\mathcal{R}(\cdot)\|_F$ are cheaply computed as in (3.15). Moreover, as suggested

Algorithm 3.2 Projected Newton-Kleinman method with extended Krylov (PNK_EK).

input : $A \in \mathbb{R}^{n\times n}$, $A<0$, $B\in\mathbb{R}^{n\times p}$, $C\in\mathbb{R}^{q\times n}$, $m_{\max}$, $\epsilon>0$, $\overline{\eta}\in(0,1)$ $\alpha\in(0,1-\overline{\eta})$.
output: $P_{k+1}\in\mathbb{R}^{n\times t}$, $t\ll n$, $P_{k+1}P_{k+1}^T = X_{k+1}\approx X$ approximate solution to (1.1).

1 Set $Y_0 = O_{2q}$, $\overline{m}=1$ and $k=0$
2 Perform economy-size QR, $[C^T, A^{-1}C^T] = [\mathcal{V}_1^{(1)}, \mathcal{V}_1^{(2)}][\boldsymbol{\gamma},\boldsymbol{\theta}]$, $\boldsymbol{\gamma},\boldsymbol{\theta}\in\mathbb{R}^{2q\times q}$
3 Set $V_1 = [\mathcal{V}_1^{(1)}, \mathcal{V}_1^{(2)}]$
4 Select $\eta_0\in(0,\overline{\eta})$
for $m=1,2,\dots,$ *till* $m_{\max}$ **do**
5 Compute next basis block $\mathcal{V}_{m+1}$ as in [42] and set $V_{m+1}=[V_m,\mathcal{V}_{m+1}]$
6 Update $T_m = V_m^T A V_m$ as in [42] and $B_m = V_m^T B$
7 Set $Q_m := T_m - \mathrm{diag}(Y_k, O_{2q(m-\overline{m})})B_m B_m^T$
8 Solve

$$Q_m\widetilde{Y} + \widetilde{Y}Q_m^T = -\mathrm{diag}(Y_k, O_{2q(m-\overline{m})})B_m B_m^T \mathrm{diag}(Y_k, O_{2q(m-\overline{m})}) - E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T$$

  if $\sqrt{2}\|\widetilde{Y}\underline{T}_m^T E_{m+1}\|_F \leqslant \eta_k \|\mathcal{R}(V_{\overline{m}} Y_k V_{\overline{m}}^T)\|_F$ **then**
9 Compute the coefficients (3.13) as in Proposition 3.2
10 Compute λ_k as in (3.14)
11 Set $Y_{k+1} = (1-\lambda_k)\cdot\mathrm{diag}(Y_k, O_{2q(m-\overline{m})}) + \lambda_k\widetilde{Y}$
  if $\|\mathcal{R}(V_m Y_{k+1} V_m^T)\|_F < \epsilon\cdot\|C^TC\|_F$ **then**
12 **Break** and go to **15**
  end
13 Set $k=k+1$ and $\overline{m}=m$
14 Select $\eta_k\in(0,\overline{\eta})$
  end
end
15 Factorize Y_{k+1} and retain $\widehat{Y}_{k+1}\in\mathbb{R}^{2mq\times t}$, $t\leqslant 2mq$
16 Set $P_{k+1} = V_m\widehat{Y}_{k+1}$

in [11], the parameter η_k is given by $\eta_k = 1/(k^3+1)$ or $\eta_k = \min\{0.1, 0.9\cdot\|\mathcal{R}(V_m Y_k V_m^T)\|_F\}$. These values lead to superlinear convergence and quadratic convergence, respectively.

To reduce the computational efforts of Algorithm 3.2, one can solve the projected equation in line 8 only periodically, say every $d\geqslant 1$ iterations. From our numerical experience, we think that this strategy may pay off if implemented only for large k, e.g., $k>3$, when $\eta_k\|\mathcal{R}(V_m Y_k V_m^T)\|_F$ is small and a quite large space is in general necessary to reach the accuracy prescribed for the current Lyapunov equation. For small k, very few iterations are sufficient for the solution of the related equations and performing line 8, and thus checking the Lyapunov residual norm, only periodically can lead to the execution of unnecessary iterations with a consequent waste of computational efforts in the solution of the linear systems for the basis generation. However, in all the reported results in section 5 we solve the projected equation at each iteration, i.e., $d=1$.

[5] Many subscripts have been omitted to make the algorithm more readable.

If the coefficient matrix A is neither negative definite nor stable, we need an initial guess X_0 such that $A - X_0BB^T$ is stable. Such an X_0 exists thanks to Assumption 2.2 and the first equation to be solved in the Newton sequence (2.1) is

$$(A - X_0BB^T)X_1 + X_1(A - X_0BB^T)^T = -X_0BB^TX_0 - C^TC. \tag{3.16}$$

Once again, in order to apply a projection method to equation (3.16), we need to suppose that the matrix $A - X_0BB^T$ is negative definite, or at least that its projected version is stable.

Supposing that such an X_0 is given and low-rank, i.e., $X_0 = S_0S_0^T$, the same argument of Theorem 3.1 shows that the $(k+1)$-th iterate of the Newton-Kleinman method can be approximated by a matrix $X_{k+1} = S_{k+1}S_{k+1}^T$ such that $\mathrm{Range}(S_{k+1}) \subset \mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, [C^T, S_0])$. Similarly, for the inexact Newton-Kleinman method, with the notation of Corollary 3.2, we can show that $X_{k+1} = P_{k+1}P_{k+1}^T$ is such that $\mathrm{Range}(P_{k+1}) \subset \mathbf{EK}^{\square}_{\overline{m}_{k+1}}(A, [C^T, S_0])$. Therefore, we can still use our new projection framework and the only modification to Algorithm 3.2 consists in replacing the starting block C^T with $[C^T, S_0]$.

3.3. The rational Krylov subspace. The same results stated in Theorem 3.1 and Corollary 3.2 can be shown also when the rational Krylov subspace (1.3) is employed as approximation space. The proofs follow the same line of the proofs of Theorem 3.1 and Corollary 3.2. The only technical difference is the presence of the shifts $\mathbf{s} = (s_2, \dots, s_m)^T$. To show that $\mathbf{K}^{\square}_{m_{k+1}}(A - V_{\overline{m}_k}\Theta_k B^T, [C^TV_{\overline{m}_k}\Theta_k], \mathbf{s}) \subset \mathbf{K}^{\square}_{\overline{m}_{k+1}}(A, C^T, \mathbf{s})$ for a sufficiently large $\overline{m}_{k+1}$, the shifts employed in the construction of the two spaces must be the same. This can be done by assuming that, e.g., the shifts are given and fixed. Also the results in Proposition 3.1 are still valid except for the computation of the residual norm as the Arnoldi relation (6.1) no longer holds. Indeed, for the rational Krylov subspace, we have

$$\begin{aligned} AV_{\overline{m}_{k+1}} &= V_{\overline{m}_{k+1}}T_{\overline{m}_{k+1}} + \mathcal{V}_{\overline{m}_{k+1}+1}E^T_{\overline{m}_{k+1}+1}\underline{H}_{\overline{m}_{k+1}}(\mathrm{diag}(s_2, \dots, s_{\overline{m}_{k+1}+1}) \otimes I_q)H^{-1}_{\overline{m}_{k+1}} \\ &\quad -(I - V_{\overline{m}_{k+1}}V^T_{\overline{m}_{k+1}})A\mathcal{V}_{\overline{m}_{k+1}+1}E^T_{\overline{m}_{k+1}+1}\underline{H}_{\overline{m}_{k+1}}H^{-1}_{\overline{m}_{k+1}}, \end{aligned} \tag{3.17}$$

where the matrix $\underline{H}_{\overline{m}_{k+1}} \in \mathbb{R}^{q(\overline{m}_{k+1}+1)\times q\overline{m}_{k+1}}$ collects the orthonormalization coefficients stemming from the orthogonalization steps and $H_{\overline{m}_{k+1}} \in \mathbb{R}^{q\overline{m}_{k+1}\times q\overline{m}_{k+1}}$ is its principal square submatrix. See, e.g., [18, 39]. Nevertheless, the residual norm can be still computed at low cost as it is shown in the next proposition.

PROPOSITION 3.4. *Consider $V_{\overline{m}_{k+1}} \in \mathbb{R}^{n\times qm_{k+1}}$, $\mathrm{Range}(V_{\overline{m}_{k+1}}) = \mathbf{K}^{\square}_{\overline{m}_{k+1}}(A, C^T, \mathbf{s})$, and let $Y_{\overline{m}_{k+1}}$ be the solution of the projected equation (3.8). Then*

$$\begin{aligned} \|(A - X_kBB^T)(V_{\overline{m}_{k+1}}Y_{\overline{m}_{k+1}}V^T_{\overline{m}_{k+1}}) + (V_{\overline{m}_{k+1}}Y_{\overline{m}_{k+1}}V^T_{\overline{m}_{k+1}})(A - X_kBB^T)^T + X_kBB^TX_k + C^TC\|_F = \\ = \|F_{\overline{m}_{k+1}}JF^T_{\overline{m}_{k+1}}\|_F, \end{aligned} \tag{3.18}$$

where $F_{\overline{m}_{k+1}}$ is the $2q \times 2q$ upper triangular matrix in the "skinny" QR factorization of

$$U_{\overline{m}_{k+1}} = [V_{\overline{m}_{k+1}}Y_{\overline{m}_{k+1}}H^{-T}_{\overline{m}_{k+1}}\underline{H}^T_{\overline{m}_{k+1}}E_{\overline{m}_{k+1}+1}, \mathcal{V}_{\overline{m}_{k+1}+1}s_{\overline{m}_{k+1}+1} - (I - V_{\overline{m}_{k+1}}V^T_{\overline{m}_{k+1}})A\mathcal{V}_{\overline{m}_{k+1}+1}],$$

and

$$J = \begin{bmatrix} O_q & I_q \\ I_q & O_q \end{bmatrix}.$$

Proof. The proof is the same of [18, Proposition 4.2]. ∎

Also the derivation of the efficient computation of the line search coefficients in Proposition 3.2 exploits the Arnoldi relation (6.1) so that new expressions are needed if the rational Krylov subspace is employed.

PROPOSITION 3.5. *Let the matrix $\widetilde{F}_{\overline{m}_{k+1}}$ be the $2q \times 2q$ upper triangular matrix in the "skinny" QR factorization of*

$$\widetilde{U}_{\overline{m}_{k+1}} = [V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}H^{-T}_{\overline{m}_{k+1}}\underline{H}^T_{\overline{m}_{k+1}}E_{\overline{m}_{k+1}+1}, \mathcal{V}_{\overline{m}_{k+1}+1}s_{\overline{m}_{k+1}+1} - (I - V_{\overline{m}_{k+1}}V^T_{\overline{m}_{k+1}})A\mathcal{V}_{\overline{m}_{k+1}+1}],$$

and the columns of $G_{\overline{m}_k} \in \mathbb{R}^{n\times 2q}$ *be an orthogonal basis for the range of* $U_{\overline{m}_k}$. *If the rational Krylov subspace is employed in the solution of* (1.1), *then the coefficients in* (3.13) *are such that*

$$\alpha_k = \|T_{\overline{m}_k} Y_{\overline{m}_k} + Y_{\overline{m}_k} T_{\overline{m}_k}^T - Y_{\overline{m}_k} B_{\overline{m}_k} B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1 \boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T\|_F^2 + \|F_{\overline{m}_k} J F_{\overline{m}_k}^T\|_F^2,$$

$$\beta_k = 2\|\widetilde{F}_{\overline{m}_{k+1}} J \widetilde{F}_{\overline{m}_{k+1}}^T\|_F^2,$$

$$\gamma_k = \begin{cases} \langle F_{\overline{m}_k} J F_{\overline{m}_k}^T, \widetilde{F}_{\overline{m}_{k+1}} J \widetilde{F}_{\overline{m}_{k+1}}^T\rangle_F, & \text{if } \overline{m}_{k+1} = \overline{m}_k, \\ 0, & \text{if } \overline{m}_{k+1} > \overline{m}_k, \end{cases}$$

$$\delta_k = \| \left(\widetilde{Y}_{\overline{m}_{k+1}} - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right) B_{\overline{m}_{k+1}} B_{\overline{m}_{k+1}}^T \left(\widetilde{Y}_{\overline{m}_{k+1}} - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)\|_F^2,$$

$$\epsilon_k = \begin{cases} \langle T_{\overline{m}_k} Y_{\overline{m}_k}, +Y_{\overline{m}_k} T_{\overline{m}_k}^T - Y_{\overline{m}_k} B_{\overline{m}_k} B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1 \boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T, \\ \left(\widetilde{Y}_{\overline{m}_{k+1}} - Y_{\overline{m}_k}\right) B_{\overline{m}_{k+1}} B_{\overline{m}_{k+1}}^T \left(\widetilde{Y}_{\overline{m}_{k+1}} - Y_{\overline{m}_k}\right)\rangle_F & \text{if } \overline{m}_{k+1} = \overline{m}_k, \\ \langle T_{\overline{m}_k} Y_{\overline{m}_k} + Y_{\overline{m}_k} T_{\overline{m}_k}^T - Y_{\overline{m}_k} B_{\overline{m}_k} B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1 \boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T, \\ [I_{2q\overline{m}_k}, O_{2q\overline{m}_k \times 2q(\overline{m}_{k+1}-\overline{m}_k)}] \left(\widetilde{Y}_{\overline{m}_{k+1}} - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right) B_{\overline{m}_{k+1}} B_{\overline{m}_{k+1}}^T \left(\widetilde{Y}_{\overline{m}_{k+1}}\right. \\ \left. - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right) [I_{2q\overline{m}_k}; O_{2q(\overline{m}_{k+1}-\overline{m}_k)\times 2q\overline{m}_k}]\rangle_F & \text{if } \overline{m}_{k+1} > \overline{m}_k, \\ + \langle F_{\overline{m}_k} J F_{\overline{m}_k}^T, G_{\overline{m}_k}^T V_{\overline{m}_{k+1}} \left(\widetilde{Y}_{\overline{m}_{k+1}} - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right) B_{\overline{m}_{k+1}} B_{\overline{m}_{k+1}}^T \left(\widetilde{Y}_{\overline{m}_{k+1}}\right. \\ \left. - diag(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right) V_{\overline{m}_{k+1}}^T G_{\overline{m}_k}\rangle_F, \end{cases}$$

and

$$\zeta_k = 0.$$

Proof. The proof follows the same line of the proof of Proposition 3.2. In the latter we deeply exploit the orthogonality of $\mathcal{V}_{\overline{m}_{k+1}+1}$ with respect to $V_{\overline{m}_{k+1}}$. Here we do the same noticing the space spanned by $\mathcal{V}_{\overline{m}_{k+1}+1} E_{\overline{m}_{k+1}+1}^T \underline{H}_{\overline{m}_{k+1}} (\mathrm{diag}(s_2, \ldots, s_{\overline{m}_{k+1}+1}) \otimes I_q) H_{\overline{m}_{k+1}}^{-1} - (I - V_{\overline{m}_{k+1}} V_{\overline{m}_{k+1}}^T) A \mathcal{V}_{\overline{m}_{k+1}+1} E_{\overline{m}_{k+1}+1}^T \underline{H}_{\overline{m}_{k+1}} H_{\overline{m}_{k+1}}^{-1}$ is orthogonal to $\mathrm{Range}(V_{\overline{m}_{k+1}})$. ∎

Even though the coefficients are computed in a different manner, Proposition 3.3 still holds as $\gamma_k = \zeta_k = 0$ for $\overline{m}_{k+1} > \overline{m}_k$. Therefore, the exact line search (3.14) can be carried out also when the rational Krylov subspace is employed.

The projected Newton-Kleinman method with rational Krylov as approximation subspace differs from Algorithm 3.2 only in the basis construction, the update of the matrix T_m and the computation of the residual norms and the line search coefficients. See Algorithm 3.3.

In practice the shifts defining the rational Krylov subspace can be computed on the fly following the approach presented in [18, Section 2]. This approach requires two values $s_0^{(1)}$ and $s_0^{(2)}$, and their complex conjugates, which define a rough approximation of the mirrored spectral region of $A - X_0 B B^T$. At the m-th iteration, if $\lambda_1, \ldots, \lambda_{qm}$ denote the eigenvalues of $T_m - \mathrm{diag}(Y_k, O_{q(m-\overline{m})}) B_m B_m^T$, the $(m+1)$-th shift is computed as

$$s_{m+1} = \operatorname*{argmax}_{s\in\mathfrak{S}_m} \frac{1}{|r_m(s)|},$$

where $r_m(s) = \prod_{j=1}^{qm} \prod_{i=2}^{m} \frac{s-\lambda_j}{s-s_i}$ and $\mathfrak{S}_m$ denotes the convex hull of $\{-\lambda_1, \ldots, -\lambda_{qm}, s_0^{(1)}, \bar{s}_0^{(1)}, s_0^{(2)}, \bar{s}_0^{(2)}\}$.

The employment of the eigenvalues of $T_m - \mathrm{diag}(Y_k, O_{q(m-\overline{m})}) B_m B_m^T$ in the computation of s_{m+1} is natural in our framework as, at the m-th iteration, we are actually trying to solve a Lyapunov equation of the same form of (2.1). We think that this is somehow related to the analysis Simoncini presented in [43] where a "pure" rational Krylov subspace method for the solution of (1.1) is studied. In [43], if V_m denotes the orthonormal basis of $\mathbf{K}_m^{\square}(A, C^T, \mathbf{s})$, the eigenvalues of $V_m^T A V_m - Y(V_m^T B)(B^T V_m)$ are employed for computing the $(m+1)$-th shift where Y denotes the solution of the projection of (1.1) onto the current subspace, namely Y is such that

$$(V_m^T A V_m) Y + Y (V_m^T A V_m)^T - Y (V_m^T B)(B^T V_m) Y + (V_m^T C^T)(C V_m) = 0.$$

Algorithm 3.3 Projected Newton-Kleinman method with rational Krylov (PNK_RK).

input : $A \in \mathbb{R}^{n\times n}$, $A < 0$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{q\times n}$, $m_{\max}$, $\epsilon > 0$, $\overline{\eta} \in (0,1)$, $\alpha \in (0, 1-\overline{\eta})$, $\{s_2, \ldots, s_{m_{\max}}\}$.
output: $P_{k+1} \in \mathbb{R}^{n\times t}$, $t \ll n$, $P_{k+1}P_{k+1}^T = X_{k+1} \approx X$ approximate solution to (1.1).

Set $Y_0 = O_q$, $\overline{m} = 1$ and $k = 0$
Perform economy-size QR of C^T, $C^T = V_1\boldsymbol{\gamma}$. Set $\mathcal{V}_1 \equiv V_1$
Select $\eta_0 \in (0, \overline{\eta})$
for $m = 1, 2, \ldots,$ *till* $m_{\max}$ **do**
  Compute next basis block $\mathcal{V}_{m+1}$ as in [18] and set $V_{m+1} = [V_m, \mathcal{V}_{m+1}]$
  Collect the orthonormalization coefficients in $\underline{H}_m \in \mathbb{R}^{q(m+1)\times qm}$
  Update $T_m = V_m^T A V_m$ as in [18] and $B_m = V_m^T B$
  Set $Q_m := T_m - \mathrm{diag}(Y_k, O_{q(m-\overline{m})})B_m B_m^T$
  Solve

$$Q_m\widetilde{Y} + \widetilde{Y}Q_m^T = -\mathrm{diag}(Y_k, O_{q(m-\overline{m})})B_m B_m^T \mathrm{diag}(Y_k, O_{q(m-\overline{m})}) - E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T$$

  Perform economy-size QR, $\widetilde{U} = \widetilde{G}\widetilde{F}$ of

$$\widetilde{U} = [V_m\widetilde{Y}H_m^{-T}\underline{H}_m^T E_{m+1}, \mathcal{V}_{m+1}s_{m+1} - (I - V_m V_m^T)A\mathcal{V}_{m+1}]$$

  if $\|\widetilde{F}J\widetilde{F}^T\|_F \leqslant \eta_k \|\mathcal{R}(V_{\overline{m}}Y_k V_{\overline{m}}^T)\|_F$ **then**
    Compute the coefficients (3.13) as in Proposition 3.5
    Compute λ_k as in (3.14)
    Set $Y_{k+1} = (1-\lambda_k)\cdot \mathrm{diag}(Y_k, O_{q(m-\overline{m})}) + \lambda_k\widetilde{Y}$
    if $\|\mathcal{R}(V_m Y_{k+1} V_m^T)\|_F < \epsilon \cdot \|C^T C\|_F$ **then**
      Break and go to **16**
    end
    Set $k = k+1$ and $\overline{m} = m$
    Select $\eta_k \in (0, \overline{\eta})$
  end
end
Factorize Y_{k+1} and retain $\widehat{Y}_{k+1} \in \mathbb{R}^{mq\times t}$, $t \leqslant mq$
Set $P_{k+1} = V_m\widehat{Y}_{k+1}$

It may be interesting to study the connection between the two approaches and this may help to better understand the convergence properties of projection methods for algebraic Riccati equations. Indeed, to the best of our knowledge, no proof about their convergence to the unique stabilizing solution is available in the literature even though, in practice, they often produce a solution X such that $A - XBB^T$ is stable. See, e.g., [27, Section 4]. However, this is beyond the scope of this paper.

4. The generalized Riccati equation. In this section we show how our new projection framework can be used also to deal with generalized Riccati equations of the form

$$\mathcal{R}_{\mathrm{gen}}(X) := AXM^T + MXA^T - MXBB^T XM^T + C^T C = 0, \tag{4.1}$$

where we suppose the mass matrix $M \in \mathbb{R}^{n\times n}$ to be symmetric positive definite (SPD) as this is the case in many applications. See, e.g., [30]. In principle, given an initial guess X_0 such that the matrix $A - MX_0BB^T$ is stable, one can apply a Newton-Kleinman-like method to (4.1) and determine the $(k+1)$-th iterate as the solution to the generalized Lyapunov equation

$$(A - MX_kBB^T)X_{k+1}M^T + MX_{k+1}(A - MX_kBB^T)^T = -MX_kBB^T X_k M^T - C^T C. \tag{4.2}$$

See, e.g., [36]. However, we prefer to pursue a different path. If $L \in \mathbb{R}^{n\times n}$ denotes the Cholesky factor of M, i.e., $M = LL^T$, we can transform the generalized equation (4.1) in a standard algebraic Riccati equation of

the form

$$\widehat{\mathcal{R}}(\widehat{X}) := \widehat{A}\widehat{X} + \widehat{X}\widehat{A}^T - \widehat{X}\widehat{B}\widehat{B}^T\widehat{X} + \widehat{C}^T\widehat{C} = 0, \tag{4.3}$$

where $\widehat{A} = L^{-1}AL^{-T}$, $\widehat{X} = L^TXL$, $\widehat{B} = L^{-1}B$ and $\widehat{C}^T = L^{-1}C^T$. Therefore, we can solve equation (4.1) by simply running Algorithm 3.2-3.3 with $\widehat{A}$, $\widehat{B}$ and $\widehat{C}$ in place of A, B and C. Notice that the matrix $\widehat{A}$ does not need to be computed but we can compute its action on a vector v by performing $\widehat{A}v = L^{-1}(A(L^{-T}v))$ and $\widehat{A}^{-1}v = L^T(A^{-1}(Lv))$ as suggested in [42, Example 5.4]. Moreover, in the construction of the basis of the rational Krylov subspace $\mathbf{K}^{\square}_m(\widehat{A}, \widehat{C}^T, \mathbf{s})$ the shifted linear systems $(\widehat{A} - s_jI)^{-1}v$ can be solved by performing $L^T\left((A - s_jM)^{-1}(Lv)\right)$.

Once we get the solution $\widehat{X}_{k+1} = \widehat{P}_{k+1}\widehat{P}_{k+1}^T$ to (4.3), we recover the low-rank solution to (4.1) as $X_{k+1} = (L^{-T}\widehat{P}_{k+1})(L^{-T}\widehat{P}_{k+1})^T$.

Clearly the one we employed is not the only possible transformation to obtain a standard Riccati equation from (4.1). For instance, one can pre and postmultiply equation (4.1) by M^{-1} and M^{-T} respectively getting an equation of the form (4.3) where $\widehat{A} = M^{-1}A$, $\widehat{X} = X$, $\widehat{B} = B$ and $\widehat{C}^T = M^{-1}C^T$. However, we think that the transformation based on the Cholesky factor L presents some advantages. For instance, it preserves symmetry in case of a symmetric A. More remarkably, the matrix L is in general much more well-conditioned than M so that solving linear systems with L may be preferable especially in case of large $\kappa(M)$. Furthermore, if

$$W(A, M) = \{\lambda \in \mathbb{C} \text{ s.t. } z^*(A - \lambda M)z = 0,\ z \in \mathbb{C}^n,\ \|z\|_F = 1\},$$

denotes the field of values of the matrix pencil (A, M) [38], then $W(A, M) = W(L^{-1}AL^{-T})$. This means that, given a generalized equation (4.1) where the matrix pencil (A, M) is negative definite, namely $W(A, B) \subset \mathbb{C}_-$, then the transformed equation (4.3) is defined by a coefficient matrix that is also negative definite if $\widehat{A} = L^{-1}AL^{-T}$. This may be no longer true if $\widehat{A} = M^{-1}A$ since $W(M^{-1}A)$ can be significantly larger than $W(A, M)$. See, e.g., [25].

Even though the generalization of our approach to equations of the form (4.1) seems straightforward, a naive application of Algorithm 3.2-3.3 to equation (4.3) should not be performed as it may happen that the computed solution $\widehat{X}_{k+1}$ is accurate for (4.3) but the recovered matrix X_{k+1} consists of a poor approximation of the solution to (4.1). In particular, for the projected Newton-Kleinman method with extended Krylov, i.e., when Algorithm 3.2 is applied to (4.3), the residual norms should be computed as shown in the following proposition.

PROPOSITION 4.1. *Suppose that Algorithm 3.2 is employed to solve the transformed equation (4.3) and let* $\widehat{X}_k = \widehat{V}_{\overline{m}_k}\widehat{Y}_{\overline{m}_k}\widehat{V}_{\overline{m}_k}^T$, $Range(\widehat{V}_{\overline{m}_k}) = \mathbf{EK}^{\square}_{\overline{m}_k}(\widehat{A}, \widehat{C}^T)$ *be the solution of the k-th transformed Lyapunov equation of the Newton-Kleinman scheme*

$$(\widehat{A} - \widehat{X}_{k-1}\widehat{B}\widehat{B}^T)\widehat{X}_k + \widehat{X}_k(\widehat{A} - \widehat{X}_{k-1}\widehat{B}\widehat{B}^T)^T = -\widehat{X}_{k-1}\widehat{B}\widehat{B}^T\widehat{X}_{k-1} - \widehat{C}^T\widehat{C}, \tag{4.4}$$

such that $X_k = L^{-T}\widehat{X}_kL^{-1}$ *solves the corresponding generalized Lyapunov equation. Suppose also that* $\widehat{Y}_{\overline{m}_{k+1}}$ *is the solution of the projected equation*

$$\widehat{Q}_{\overline{m}_{k+1}}\widehat{Y} + \widehat{Y}\widehat{Q}_{\overline{m}_{k+1}}^T = -diag(\widehat{Y}_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\widehat{B}_{\overline{m}_{k+1}}\widehat{B}_{\overline{m}_{k+1}}^T diag(\widehat{Y}_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) - E_1\widehat{\gamma}\widehat{\gamma}^TE_1^T, \tag{4.5}$$

where $\widehat{Q}_{\overline{m}_{k+1}} = \widehat{T}_{\overline{m}_{k+1}} - diag(\widehat{Y}_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\widehat{B}_{\overline{m}_{k+1}}\widehat{B}_{\overline{m}_{k+1}}^T$, $\widehat{T}_{\overline{m}_{k+1}} = \widehat{V}_{\overline{m}_{k+1}}^T\widehat{A}\widehat{V}_{\overline{m}_{k+1}}$, $\widehat{B}_{\overline{m}_{k+1}} = \widehat{V}_{\overline{m}_{k+1}}^T\widehat{B}$ *and* $\widehat{C}^T = \widehat{V}_1\widehat{\gamma}$, $\widehat{V}_{\overline{m}_{k+1}} \in \mathbb{R}^{n\times 2q\overline{m}_{k+1}}$, $Range(\widehat{V}_{\overline{m}_{k+1}}) = \mathbf{EK}^{\square}_{\overline{m}_{k+1}}(\widehat{A}, \widehat{C}^T)$, $\overline{m}_{k+1} \geqslant \overline{m}_k$.

Then, if $X_{k+1} = L^{-T}(\widehat{V}_{\overline{m}_{k+1}}\widehat{Y}_{\overline{m}_{k+1}}\widehat{V}_{\overline{m}_{k+1}}^T)L^{-1}$, *we have*

$$\|(A - MX_kBB^T)X_{k+1}M^T + MX_{k+1}(A - MX_kBB^T)^T + MX_kBB^TX_kM^T + C^TC\|_F = \|N_{\overline{m}_{k+1}}KN_{\overline{m}_{k+1}}^T\|_F,$$

where $N_{\overline{m}_{k+1}}$ *denotes the* $4q \times 4q$ *upper triangular matrix in the "skinny" QR factorization of*

$$D_{\overline{m}_{k+1}} = L\,[\widehat{V}_{\overline{m}_{k+1}}\widehat{Y}_{\overline{m}_{k+1}}\widehat{T}_{\overline{m}_{k+1}}^T\widehat{E}_{\overline{m}_{k+1}+1}, \widehat{\mathcal{V}}_{\overline{m}_{k+1}+1}],$$

$$K = \begin{bmatrix} O_{2q} & I_{2q} \\ I_{2q} & O_{2q} \end{bmatrix},$$

and $\widehat{\underline{T}}_{\overline{m}_{k+1}} = [\widehat{V}_{\overline{m}_{k+1}}, \widehat{\mathcal{V}}_{\overline{m}_{k+1}+1}]^T \widehat{A}\widehat{V}_{\overline{m}_{k+1}}$. *Moreover,*

$$\|\mathcal{R}_{gen}(X_{k+1})\|_F = \|\Pi_{\overline{m}_{k+1}} \Phi_{\overline{m}_{k+1}} \Pi_{\overline{m}_{k+1}}^T\|_F \tag{4.6}$$

where $\Pi_{\overline{m}_{k+1}}$ *denotes the* $(q(4\overline{m}_{k+1}+5)+p) \times (q(4\overline{m}_{k+1}+5)+p)$ *upper triangular matrix in the "skinny" QR factorization of*

$$W_{\overline{m}_{k+1}} = [LV_{\overline{m}_{k+1}} [\widehat{T}_{\overline{m}_{k+1}}, \widehat{Y}_{\overline{m}_{k+1}}, \widehat{Y}_{\overline{m}_{k+1}} \widehat{B}_{\overline{m}_{k+1}}, E_1 \widehat{\boldsymbol{\gamma}}], D_{\overline{m}_{k+1}}],$$

and

$$\Phi_{\overline{m}_{k+1}} = \begin{bmatrix} O_{2q\overline{m}_{k+1}} & I_{2q\overline{m}_{k+1}} & & & \\ I_{2q\overline{m}_{k+1}} & O_{2q\overline{m}_{k+1}} & & & \\ & & -I_p & & \\ & & & I_q & \\ & & & & K \end{bmatrix}.$$

Proof. For the extended Krylov subspace $\mathbf{EK}^{\square}_{\overline{m}_{k+1}}(\widehat{A}, \widehat{C}^T)$, the following Arnoldi relation holds

$$(L^{-1}AL^{-T})\widehat{V}_{\overline{m}_{k+1}} = \widehat{V}_{\overline{m}_{k+1}} \widehat{T}_{\overline{m}_{k+1}} + \widehat{\mathcal{V}}_{\overline{m}_{k+1}+1} E_{\overline{m}_{k+1}+1}^T \widehat{\underline{T}}_{\overline{m}_{k+1}},$$

so that

$$AL^{-T}\widehat{V}_{\overline{m}_{k+1}} = L\widehat{V}_{\overline{m}_{k+1}} \widehat{T}_{\overline{m}_{k+1}} + L\widehat{\mathcal{V}}_{\overline{m}_{k+1}+1} E_{\overline{m}_{k+1}+1}^T \widehat{\underline{T}}_{\overline{m}_{k+1}}.$$

If $X_{k+1} = L^{-T}(\widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T) L^{-1}$, we have

$$\begin{aligned}&(A - MX_kBB^T)X_{k+1}M^T + MX_{k+1}(A - MX_kBB^T)^T + MX_kBB^TX_kM^T + C^TC = \\ &(A - MX_kBB^T)L^{-T}(\widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T)L^T + L(\widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T)L^{-1}(A - MX_kBB^T)^T + MX_kBB^TX_kM^T + C^TC,\end{aligned}$$

and, recalling that $X_k = L^{-T}\widehat{X}_k L^{-1} = L^{-T}(\widehat{V}_{\overline{m}_k} \widehat{Y}_{\overline{m}_k} \widehat{V}_{\overline{m}_k}^T)L^{-1}$, the right-hand side in the above expression can be written as

$$\begin{aligned}&L\big((\widehat{V}_{\overline{m}_{k+1}} \widehat{T}_{\overline{m}_{k+1}} - \widehat{V}_{\overline{m}_k} \widehat{Y}_{\overline{m}_k} \widehat{B}_{\overline{m}_k} \widehat{B}_{\overline{m}_{k+1}}^T) \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T + \widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} (\widehat{V}_{\overline{m}_{k+1}} \widehat{T}_{\overline{m}_{k+1}} - \widehat{V}_{\overline{m}_k} \widehat{Y}_{\overline{m}_k} \widehat{B}_{\overline{m}_k} \widehat{B}_{\overline{m}_{k+1}}^T)^T \\ &+ \widehat{V}_{\overline{m}_k} \widehat{Y}_{\overline{m}_k} \widehat{B}_{\overline{m}_k} \widehat{B}_{\overline{m}_k}^T \widehat{Y}_{\overline{m}_k} \widehat{V}_{\overline{m}_k}^T + \widehat{C}^T\widehat{C} + \widehat{\mathcal{V}}_{\overline{m}_{k+1}+1} E_{\overline{m}_{k+1}+1}^T \widehat{\underline{T}}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T + \widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{\underline{T}}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1} \widehat{\mathcal{V}}_{\overline{m}_{k+1}+1}^T\big) L^T.\end{aligned}$$

Defining $\widehat{Q}_{\overline{m}_{k+1}} = \widehat{T}_{\overline{m}_{k+1}} - \mathrm{diag}(\widehat{Y}_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}) \widehat{B}_{\overline{m}_{k+1}} \widehat{B}_{\overline{m}_{k+1}}^T$ and recalling that $\widehat{Y}_{\overline{m}_{k+1}}$ solves (4.5), we can write

$$\begin{aligned}&(A - MX_kBB^T)X_{k+1}M^T + MX_{k+1}(A - MX_kBB^T)^T + MX_kBB^TX_kM^T + C^TC = \\ &\quad = L\left(\widehat{\mathcal{V}}_{\overline{m}_{k+1}+1} E_{\overline{m}_{k+1}+1}^T \widehat{\underline{T}}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T + \widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{\underline{T}}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1} \widehat{\mathcal{V}}_{\overline{m}_{k+1}+1}^T\right) L^T = D_{\overline{m}_{k+1}} K D_{\overline{m}_{k+1}}^T,\end{aligned}$$

and get the result.

Similarly, for the residual of the generalized Riccati equation (4.3), we have

$$\begin{aligned}\|\mathcal{R}_{\mathrm{gen}}(X_{k+1})\|_F \quad = \quad &\|AL^{-T}\widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T L^T + L\widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T L^{-1}A^T \\ &-L\widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T \widehat{B}\widehat{B}^T \widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T L^T + C^TC\|_F \\ \\ &\|L\big(\widehat{V}_{\overline{m}_{k+1}} \big(\widehat{T}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} + Y_{\overline{m}_{k+1}} \widehat{T}_{\overline{m}_{k+1}}^T - \widehat{Y}_{\overline{m}_{k+1}} \widehat{B}_{\overline{m}_{k+1}} \widehat{B}_{\overline{m}_{k+1}}^T \widehat{Y}_{\overline{m}_{k+1}} \\ &+E_1\widehat{\boldsymbol{\gamma}}\widehat{\boldsymbol{\gamma}}^T E_1^T\big) \widehat{V}_{\overline{m}_{k+1}}^T + \widehat{\mathcal{V}}_{\overline{m}_{k+1}+1} E_{\overline{m}_{k+1}+1}^T \widehat{\underline{T}}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T \\ &+\widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{\underline{T}}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1} \widehat{\mathcal{V}}_{\overline{m}_{k+1}+1}^T\big) L^T\|_F.\end{aligned}$$

Since this matrix can be written as $W_{\overline{m}_{k+1}} \Phi_{\overline{m}_{k+1}} W_{\overline{m}_{k+1}}^T$ we get the result. ∎

A similar result holds also when the rational Krylov space is adopted as approximation space. In this case, in addition to proper dimensions of K and $\Phi_{\overline{m}_{k+1}}$, the matrix $D_{\overline{m}_{k+1}}$ in Proposition 4.1 must be replaced by the matrix

$$L[\widehat{V}_{\overline{m}_{k+1}} \widehat{Y}_{\overline{m}_{k+1}} \widehat{H}_{\overline{m}_{k+1}}^{-T} \widehat{\underline{H}}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}, \widehat{\mathcal{V}}_{\overline{m}_{k+1}+1} s_{\overline{m}_{k+1}+1} - (I - \widehat{V}_{\overline{m}_{k+1}} \widehat{V}_{\overline{m}_{k+1}}^T)\widehat{A}\widehat{\mathcal{V}}_{\overline{m}_{k+1}+1}],$$

where $\underline{\widehat{H}}_{\overline{m}_{k+1}}$ collects the orthonormalization coefficients computed during the construction of $\mathbf{K}_m^{\square}(\widehat{A}, \widehat{C}^T, \mathbf{s})$.

The extra work for computing the actual residual norms of the generalized Lyapunov equation (4.2) and the generalized Riccati equation (4.1) is necessary to check if the computed quantities are really what we are looking for. Indeed,

$$\frac{\|\mathcal{R}_{\text{gen}}(X_{k+1})\|_F}{\|\mathcal{R}_{\text{gen}}(X_0)\|_F} \leqslant \kappa_F(L)^2 \frac{\|\widehat{\mathcal{R}}(\widehat{X}_{k+1})\|_F}{\|\widehat{\mathcal{R}}(\widehat{X}_0)\|_F},$$

and in a naive application of Algorithm 3.2 to (4.3), only $\|\widehat{\mathcal{R}}(\widehat{X}_{k+1})\|_F / \|\widehat{\mathcal{R}}(\widehat{X}_0)\|_F$ would be computed to check convergence. This may be misleading for the solution of the generalized Riccati equation (4.1) especially if the condition number $\kappa_F(L) := \|L\|_F \|L^{-1}\|_F$ is large.

We must mention that the results in Proposition 3.2-3.3-3.5 are still valid for the transformed equation (4.3) but this does not imply that such a line search is exact also for the generalized equation (4.1).

The computation of the coefficients in (3.13) would require different expressions from the ones in Proposition 3.2-3.5. The presence of the matrix L in the computation of the residual norms does not let us exploit the orthogonality of the basis $V_{\overline{m}_k}$ and the new derivation may be cumbersome. We thus decide to study in depth this topic in a future work as we think that an exact line search can be derived also for the inexact Newton-Kleinman method applied to generalized Riccati equations if non standard inner products are considered.

Roughly speaking, what we suggest to do for solving equation (4.1) is the following. The Lyapunov equation (4.4) is iteratively solved until $\|N_{\overline{m}_{k+1}} K N_{\overline{m}_{k+1}}^T\|_F \leqslant \eta_k \|\mathcal{R}_{\text{gen}}(X_k)\|_F$. Then we perform an exact line search for the transformed equation (4.3) to compute the next iterate $\widehat{X}_{k+1}$. If $X_{k+1} = L^{-T} \widehat{X}_{k+1} L^{-1}$ provides a sufficiently small residual norm $\|\mathcal{R}_{\text{gen}}(X_{k+1})\|_F$ we stop the process, otherwise we expand the space and pass to solve the $(k+1)$-th (standard) Lyapunov equation.

5. Numerical examples. In this section we compare Algorithm 3.2-3.3 with state-of-the-art methods for the solution of large-scale algebraic Riccati equations. In particular, our new procedures are compared with the inexact Newton-Kleinman method with ADI as inner solver (iNK+ADI) [11], the Newton-Kleinman method with Galerkin acceleration (NK+GP) [14], projection methods with extended (EKSM) and rational (RKSM) Krylov subspace and RADI [9]. The two variants of the Newton-Kleinman method are available in the M-M.E.S.S. package [40]. A Matlab implementation of projection methods for Riccati equations can be found on the web page of Simoncini[6] while we thank Jens Saak for providing us with the RADI code[7].

The performances of the algorithms are compared in terms of memory requirements and CPU time. For the former we report the maximum number of vectors of length n that need to be stored. For instance, in our framework the storage demand consists in the dimension of the computed subspace. The same for the "pure" projection procedures. For the other methods, the memory requirements amount to the number of columns of the low-rank factor of the solution and we thus check this value at each step, before any low-rank truncation is performed. See [9,11,14] for further details. We also report the rank of the computed solution, the relative residual norm achieved with such a solution and the number of (outer) iterations performed to converge. In iNK+ADI and in NK+GP, ADI is employed as inner solver for the Lyapunov equations stemming from the Newton scheme. We thus report also the average number of ADI iterations. See, e.g., [33] for further details about ADI.

The RADI method, as its linear counterpart ADI, requires the computation of effective shifts. In [9] several kinds of shifts s_j are proposed and the performance achieved with different s_j's seems to be highly problem dependent. However, the residual Hamiltonian shifts, denoted in [9, Section 5] by "Ham, $\ell = 2p$"[8], provide very good performance in all the experiments reported in [9]. We thus employ the same RADI shifts. When ADI is used as inner solver in iNK+ADI and in NK+GP, the default setting of the M-M.E.S.S. package is used for computing the ADI shifts.

In [11], two values for the forcing parameter η_k are proposed: $\eta_k = 1/(1+k^3)$ and $\eta_k = \max\{0.1, 0.9 \cdot \|\mathcal{R}(V_m Y_k V_m)\|_F\}$. These values lead to a superlinear and a quadratic convergence of the Newton scheme respectively. However, in all our numerical experiments, we notice a remarkable increment in both the CPU time and the memory requirements of our new procedure when $\eta_k = \max\{0.1, 0.9 \cdot \|\mathcal{R}(V_m Y_k V_m)\|_F\}$ is

[6] `http://www.dm.unibo.it/~simoncin/software.html`

[7] A Matlab implementation will be available in the next version of the M-M.E.S.S. package.

[8] In our notation it would be "Ham, $\ell = 2q$" as $p = \text{rank}(C^T)$ in [9].

employed. Indeed, the quadratic convergence obtained is in terms of the number k of Lyapunov equations we need to solve. Even though we have to solve less equations, each of them requires to be more accurately solved and, in general, this means that a larger subspace has to be generated and more computational efforts are thus demanded. Therefore, in all the reported experiments, $\eta_k = 1/(1+k^3)$.

The tolerance for the final relative residual norm is always set to 10^{-8}.

All results were obtained with Matlab R2017b [35] on a Dell machine with two 2GHz processors and 128 GB of RAM.

EXAMPLE 5.1. In the first example we consider a matrix A in (1.1) stemming from the centered finite difference discretization of the 3D lapalcian $\mathcal{L}(u) = \Delta u$ on the unit cube with zero Dirichlet boundary condition. In particular, if $T = 1/(n_0-1)^2 \cdot \text{tridiag}(1, \underline{-2}, 1)$ denotes the matrix representing the discrete operator associated to the 1D lapalcian, then

$$A = T \otimes I_{n_0} \otimes I_{n_0} + I_{n_0} \otimes T \otimes I_{n_0} + I_{n_0} \otimes I_{n_0} \otimes T.$$

The matrix A is thus symmetric negative definite. Since all the methods we compare require solving many linear systems with A - or a shifted A - we reorder the entries of this matrix by the Matlab function `amd`.

The low-rank matrices $B \in \mathbb{R}^{n\times p}$ and $C \in \mathbb{R}^{q\times n}$ have random entries that have been scaled by the mesh size $1/(n_0-1)^2$ to match the magnitude of the components of A. In particular, $B = 1/(n_0-1)^2 \cdot \texttt{rand}(n,p)$ where $n = n_0^3$. Similarly for C. The same reordering applied to A has been applied also to B and C.

In Table 5.1 we report the results for $n = 125000$ and different values of p and q.

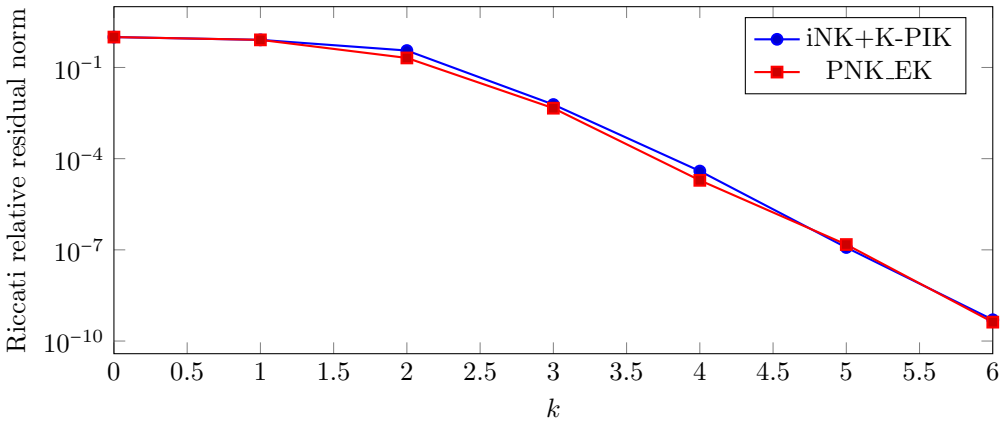
Table 5.1: Example 5.1. Results for different values of p and q.

	It. (inner It.)	Mem.	rank(X)	Rel. Res	Time (secs)
	$p = q = 1$				
PNK_EK	15 (-)	30	18	4.18e-10	10.19
PNK_RK	17 (-)	17	17	7.35e-12	28.10
EKSM	13 (-)	26	24	4.16e-9	10.23
RKSM	12 (-)	12	12	4.58e-9	20.26
iNK+ADI	2 (7.5)	16	13	6.29e-9	27.85
NK+GP	2 (30.5)	64	25	3.07e-15	120.72
RADI	12 (-)	12	12	4.86e-9	20.55
	$p = q = 10$				
PNK_EK	14 (-)	280	186	1.36e-9	37.62
PNK_RK	15 (-)	150	150	1.13e-9	33.65
EKSM	13 (-)	260	234	5.15e-9	35.85
RKSM	14 (-)	140	140	6.93e-9	31.13
iNK+ADI	3 (14)	340	177	6.23e-10	99.51
NK+GP	32.5 (2)	700	237	3.60e-11	226.52
RADI	14 (-)	140	140	2.98e-9	33.79
	$p = 10,\ q = 1$				
PNK_EK	14 (-)	28	16	1.36e-10	9.96
PNK_RK	12 (-)	12	12	3.02e-9	21.46
EKSM	11 (-)	22	19	3.93e-9	9.53
RKSM	11 (-)	11	11	4.62e-9	19.74
iNK+ADI	3 (9)	154	18	2.45e-10	53.71
NK+GP	2 (29.5)	330	21	2.06e-11	139.58
RADI	12 (-)	12	12	1.60e-9	22.91

In this example, the iNK+ADI and the NK+GP are not very competitive in terms of both memory requirements and CPU time when compared to the other methods.

The procedures based on the extended Krylov subspace, i.e., PNK_EK and EKSM, are very fast. Indeed,

Fig. 5.1: Example 5.1. Relative residual norms produced by iNK+K-PIK and PNK_EK for $p = q = 1$.



they need few iterations to converge and the precomputation of the LU factors[9] of A makes the linear solves very cheap. Very few iterations are needed also in PNK_RK and RKSM but these are computationally more expensive due to the presence of different shifts in the linear systems. The gains coming from the precomputation of the LU factors are less outstanding in the case $p = q = 10$ to the point that PNK_RK and RKSM turn out to be faster than the corresponding procedures based on the extended Krylov subspace. This is mainly due to the cost of the inner solves. Indeed, the solution of the projected equation[10] grows cubically with the space dimension and in PNK_EK and EKSM a quite large space is constructed when $p = q = 10$.

The methods based on the rational Krylov subspace demand little storage and provide a very low-rank solution. This is typical also when projection methods are applied to Lyapunov equations.

For all the tested values of p and q, both PNK_EK and PNK_RK implicitly solve six Lyapunov equations of the Newton scheme (2.1).

The RADI method is very competitive in terms of both memory requirements and CPU time and its performance is very similar to the ones achieved by PNK_RK and RKSM.

For $p = q = 1$, we also compare PNK_EK with the inexact Newton-Kleinman method where each Lyapunov equations of the scheme is solved by K-PIK [42]. Such a procedure is called iNK+K-PIK in the following and it solves the $(k+1)$-th Lyapunov equation (2.5) by projection onto the extended Krylov subspace $\mathbf{EK}^{\square}_m(A - X_kBB^T, [C^T, X_kB])$. In both PNK_EK and iNK+K-PIK we need to solve six Lyapunov equations to achieve $\|\mathcal{R}(X_{k+1})\|_F/\|C^TC\|_F \leqslant 10^{-8}$ and the relative residual norms produced by the two methods have a very similar trend. See Figure 5.1.

We want to compare the dimension of the subspaces constructed by iNK+K-PIK to solve the $k+1$ equations of the Newton scheme (2.5) with the corresponding dimension of $\mathbf{EK}^{\square}_m(A, C^T)$, i.e., with $2qm = \dim(\mathbf{EK}^{\square}_m(A, C^T))$ such that $\sqrt{2}\|\widetilde{Y}\underline{T}^T_m E_{m+1}\|_F \leqslant \eta_k \|\mathcal{R}(V_{\overline{m}} Y_k V_{\overline{m}})\|_F$ in Algorithm 3.2 for $k = 0, \ldots, 5$. The results are reported in Table 5.2.

Table 5.2: Example 5.1, $p = q = 1$. Comparison between the memory consumption of iNK+K-PIK and PNK_EK.

	k					
	0	1	2	3	4	5
$\dim(\mathbf{EK}^{\square}_m(A - X_kBB^T, [C^T, X_kB]))$ (It.)	4 (2)	8 (2)	8 (2)	12 (3)	24 (6)	36 (9)
$\dim(\mathbf{EK}^{\square}_m(A, C^T))$ (It.)	4 (2)	6 (3)	8 (4)	12 (6)	20 (10)	30 (15)

[9] The time for the computation of the LU factorization is included in all the reported results, also for the next examples.

[10] This amounts to a Lyapunov equation in case of PNK_EK and PNK_RK, and a Riccati equation in case of EKSM and RKSM.

If $k > 0$, the dimension of $\mathbf{EK}^{\square}_m(A - X_k BB^T, [C^T, X_k B])$ grows faster than $\dim(\mathbf{EK}^{\square}_m(A, C^T))$ as four new basis vectors are added to the current space at each iteration instead of only two. This may lead to some redundancy in $\mathbf{EK}^{\square}_m(A - X_k BB^T, [C^T, X_k B])$ and, at least for this example, a smaller subspace can be constructed to achieve the same level of accuracy in the solution of the $(k+1)$-th equation. For instance, in the solution of the second Lyapunov equation ($k = 1$) only one iteration of K-PIK is not sufficient to achieve the prescribed level of accuracy and a second iteration is performed so that the algorithm necessarily ends up constructing a subspace of dimension 8. On the other hand, since only two basis vectors are added to $\mathbf{EK}^{\square}_m(A, C^T)$ at each iteration, PNK_EK manages to realize that a space of dimension 6 contains already enough spectral information to solve the second equation. Moreover, the final dimension of $\mathbf{EK}^{\square}_m(A, C^T)$ is much smaller than the one predicted by Corollary 3.2. Indeed, for this example, the latter amounts to 52 but a subspace of dimension 15 is sufficient to solve the Riccati equation.

Notice that iNK+K-PIK and PNK_EK are not comparable from a CPU time perspective. Indeed, iNK+K-PIK constructs $\mathbf{EK}^{\square}_m(A - X_k BB^T, [C^T, X_k B])$ from scratch for all $k = 0, \ldots, 5$ and the computation of the last space $\mathbf{EK}^{\square}_m(A - X_5 BB^T, [C^T, X_5 B])$ is more expensive than the overall PNK_EK procedure.

EXAMPLE 5.2. We now consider the matrix $T \in \mathbb{R}^{n \times n}$, $n = 109460$, denominated `lung` in the UF Sparse Matrix Collection [16]. This unsymmetric matrix has been used in [9, Example 6] as coefficient matrix of the Riccati equation (1.1). However, T is *anti-stable*, i.e., the spectrum of T is contained in the right half plane. We thus consider $-T$ to our purpose. Even though $-T$ is stable, it is indefinite as $\overline{\lambda} := \max_j(\lambda_j(-T - T^T)/2) > 0$. Since we are not aware of any low-rank X_0 such that $-T - X_0 BB^T < 0$, we prefer to shift $-T$ and consider the negative definite matrix $A := -T - (\overline{\lambda} + 1)I$. Moreover, the entries of A have been reordered by means of the Matlab function `symrcm`. As before, the matrices $B \in \mathbb{R}^{n \times p}$ and $C \in \mathbb{R}^{q \times n}$ have random entries which have been reordered by the same permutations applied to A.

In Table 5.3 we report the results for different values of p and q.

Table 5.3: Example 5.2. Results for different values of p and q.

	It. (inner it.)	Mem.	rank(X)	Rel. Res	Time
	$p = q = 1$				
PNK_EK	36 (-)	72	50	9.82e-9	3.69
PNK_RK	29 (-)	29	29	7.62e-9	7.25
EKSM	36 (-)	72	60	9.83e-9	3.40
RKSM	29 (-)	29	29	9.07e-9	7.36
iNK+ADI	4 (10.75)	50	31	8.01e-9	8.38
NK+GP	2 (62)	128	55	4.40e-15	23.12
RADI	33 (-)	33	33	5.61e-9	5.81
	$p = q = 10$				
PNK_EK	26 (-)	520	369	3.97e-11	24.24
PNK_RK	28 (-)	280	280	2.07e-9	25.72
EKSM	24 (-)	480	433	5.19e-9	24.90
RKSM	26 (-)	260	260	6.01e-9	22.53
iNK+ADI	4 (13.25)	600	318	5.06e-9	40.75
NK+GP	2 (73.5)	1640	388	1.08e-14	107.35
RADI	31 (-)	310	310	2.03e-9	24.54
	$p = 10,\ q = 1$				
PNK_EK	48 (-)	96	46	1.02e-10	5.62
PNK_RK	30 (-)	30	30	3.72e-9	7.89
EKSM	33 (-)	66	55	9.42e-9	3.36
RKSM	28 (-)	28	28	8.80e-9	7.28
iNK+ADI	4 (12.25)	286	34	7.65e-9	24.80
NK+GP	2 (61.5)	671	55	2.69e-14	52.36
RADI	27 (-)	27	27	6.10e-9	8.16

PNK_EK and EKSM are still among the fastest methods, especially for small q, and PNK_RK, RKSM

and RADI exhibit similar results. For all the tested values of p and q, both PNK_EK and PNK_RK implicitly solve four Lyapunov equations.

We would like to underline how the computational cost of our new procedures does not really depend on p. More precisely, in Algorithm 3.2-3.3 we only solve q linear systems per iteration, similarly to what is done in EKSM and RKSM. This does not hold for iNK+ADI, RADI and NK+GP. Indeed, in these methods, linear systems of the form $(A+\theta_j I+UV^T)Z=W$, $W\in\mathbb{R}^{n\times\ell}$, $U,V\in\mathbb{R}^{n\times p}$, have to be solved at each (inner) iteration. The number of columns ℓ of the right-hand side W depends on the selected method. In particular, for iNK+ADI and NK+GP, $\ell=p+q$ so that, by employing the SMW formula, we solve $2p+q$ linear system at each inner iteration. In RADI, $W\in\mathbb{R}^{n\times q}$ and $p+q$ linear systems are solved at each iteration. See, e.g., [8] for more details. Therefore, if p is large compared to q, the computational cost of iNK+ADI, RADI and NK+GP may dramatically increase while it remains almost constant in PNK_EK and PNK_RK. For instance, if we compare the performance of PNK_RK for the cases $p=q=1$ and $p=10$, $q=1$ we obtain a similar number of iterations and basically the same CPU time. On the other hand, the time of iNK+ADI and NK+GP is more than the double when $p=10$, $q=1$ compared to the case $p=q=1$. Also the CPU time of RADI increases when $p=10$ and $q=1$ even though we perform less iterations compared to the case $p=q=1$.

Example 5.3. In the last example we consider the RAIL benchmark problem in [30]. This consists of a semidiscretized heat transfer problem for optimal cooling of steel profiles. The algebraic problem amounts to a generalized Riccati equation of the form (4.1) where $A\in\mathbb{R}^{n\times n}$ is symmetric negative definite, $M\in\mathbb{R}^{n\times n}$ is SPD, $B\in\mathbb{R}^{n\times 7}$ and $C\in\mathbb{R}^{6\times n}$. For PNK_EK, PNK_RK, EKSM and RKSM, the generalized Riccati equation (4.1) is solved by performing the transformation based on the Cholesky factor of M as described in section 4[11]. RADI, iNK+ADI and NK+GP directly work on the generalized Riccati equation (4.1).

In Table 5.4 we report the results for different values of n.

Table 5.4: Example 5.3. Results for different values of n.

	It. (inner it.)	Mem.	rank(X)	Rel. Res	Time
	$n=20209$				
PNK_EK	67 (-)	804	585	7.73e-11	176.52
PNK_RK	33 (-)	198	198	1.63e-10	9.11
EKSM	41 (-)	492	458	8.45e-9	51.63
RKSM	27 (-)	162	162	2.54e-9	4.41
iNK+ADI	4 (11.57)	286	177	8.72e-9	4.10
NK+GP	2 (62.5)	858	219	2.61e-11	11.09
RADI	31 (-)	186	165	3.47e-9	3.11
	$n=79841$				
PNK_EK	98 (-)	1176	802	6.29e-10	1418.95
PNK_RK	35 (-)	210	210	5.17e-10	48.50
EKSM	55 (-)	660	621	8.19e-9	577.06
RKSM	29 (-)	174	174	7.40e-9	21.76
iNK+ADI	4 (13.75)	390	204	3.94e-10	19.74
NK+GP	2 (69.5)	910	250	1.21e-10	47.85
RADI	35 (-)	210	177	9.82e-9	16.6

From the timings in Table 5.4 it is clear that our new procedures are not competitive with the other methods. This is mainly due to the expensive residual norm computation in (4.6). For instance, for $n=20209$, PNK_EK implicitly solves twelve Lyapunov equations so that the residual norm of the generalized Riccati operator (4.6) has been evaluated 12 times. This operation takes 120.7 seconds that is about the 68% of the overall computational time.

We believe that the performances of our new projection framework applied to generalized Riccati equa-

[11] The EKSM and RKSM codes downloaded from `http://www.dm.unibo.it/~simoncin/software.html` have been modified to compute the actual residual norm of (4.1).

tions can be improved by, e.g., employing non-standard inner products, but this requires further study. We plan to do this in the near future.

6. Conclusions. A novel and effective approach for solving large-scale algebraic Riccati equations has been developed. The inexact Newton-Kleinman method has been combined with projection techniques that rely on timely approximation spaces as the extended and the rational Krylov subspaces. In our approach, only one approximation space is constructed as in the "pure" projection methods for matrix equations making our algorithm very efficient. The projected Newton-Kleinman procedures PNK_EK and PNK_RK perform very similarly to EKSM and RKSM respectively, in terms of both memory requirements and computational time. Moreover, the convergence to the unique stabilizing solution is guaranteed in our new framework as the well-established properties of the inexact Newton-Kleinman method are preserved thanks to the employment of a line search which turns out to be exact in our setting. The robust convergence theory our new schemes are based on represents the main advantage of the proposed methods when compared to the pure projection counterparts.

The numerical results show how our new algorithms are very competitive also with state-of-the-art procedures which are not based on projection. The only exception is when we have to deal with generalized Riccati equations. We believe that the performances of the projected Newton-Kleinman method applied to generalized Riccati equations can be largely improved. This will be the topic of future works.

Another research direction is the solution of nonsymmetric Riccati equations [12,15]. Our new algorithms can be easily adapted to handle nonsymmetric problems and the solution process only require the construction of a *right* and a *left* subspace, in agreement with standard procedures for Sylvester equations. See, e.g., [44, Section 4.4.1].

Acknowledgments. We wish to thank Valeria Simoncini, Stefano Massei, Leonardo Robol and Patrick Kürschner for insightful conversations about earlier versions of the manuscript. Their helpful suggestions are greatly appreciated.

The author is a member of the Italian INdAM Research group GNCS.

Appendix. Here we report the proof of Proposition 3.2.

Proof. In this proof we only need the Arnoldi relation

$$AV_{\overline{m}_{k+1}} = V_{\overline{m}_{k+1}} T_{\overline{m}_{k+1}} + \mathcal{V}_{\overline{m}_{k+1}+1} E_{\overline{m}_{k+1}+1}^T \underline{T}_{\overline{m}_{k+1}}, \tag{6.1}$$

and the cyclic property of the trace operator, i.e., $\mathrm{trace}(ABCD) = \mathrm{trace}(DABC)$ for A, B, C, D matrices of conforming dimensions.

We have

$$\begin{aligned}
\alpha_k &= \|\mathcal{R}(X_k)\|_F^2 = \|\mathcal{R}(V_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T)\|_F^2 \\
&= \|AV_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T + V_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T A^T - V_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T BB^T V_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T + C^T C\|_F^2 \\
&= \|V_{\overline{m}_k} T_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T + V_{\overline{m}_k} Y_{\overline{m}_k} T_{\overline{m}_k}^T V_{\overline{m}_k}^T - V_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T BB^T V_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T + C^T C \\
&\quad + \mathcal{V}_{\overline{m}_k+1} E_{\overline{m}_k+1}^T \underline{T}_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T + V_{\overline{m}_k} Y_{\overline{m}_k} \underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1} \mathcal{V}_{\overline{m}_k+1}^T \|_F^2 \\
&= \|V_{\overline{m}_k} \left(T_{\overline{m}_k} Y_{\overline{m}_k} + Y_{\overline{m}_k} T_{\overline{m}_k}^T - Y_{\overline{m}_k} B_{\overline{m}_k} B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1 \boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T \right) V_{\overline{m}_k}^T \\
&\quad + \mathcal{V}_{\overline{m}_k+1} E_{\overline{m}_k+1}^T \underline{T}_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T + V_{\overline{m}_k} Y_{\overline{m}_k} \underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1} \mathcal{V}_{\overline{m}_k+1}^T \|_F^2.
\end{aligned}$$

Since $\langle \mathcal{V}_{\overline{m}_k+1}, V_{\overline{m}_k} \rangle_F = 0$ by construction, we have

$$\begin{aligned}
\alpha_k &= \|T_{\overline{m}_k} Y_{\overline{m}_k} + Y_{\overline{m}_k} T_{\overline{m}_k}^T - Y_{\overline{m}_k} B_{\overline{m}_k} B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1 \boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T \|_F^2 \\
&\quad + \|\mathcal{V}_{\overline{m}_k+1} E_{\overline{m}_k+1}^T \underline{T}_{\overline{m}_k} Y_{\overline{m}_k} V_{\overline{m}_k}^T + V_{\overline{m}_k} Y_{\overline{m}_k} \underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1} \mathcal{V}_{\overline{m}_k+1}^T \|_F^2 \\
&= \|T_{\overline{m}_k} Y_{\overline{m}_k} + Y_{\overline{m}_k} T_{\overline{m}_k}^T - Y_{\overline{m}_k} B_{\overline{m}_k} B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1 \boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T \|_F^2 + 2\|Y_{\overline{m}_k} \underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}\|_F^2.
\end{aligned}$$

Moreover, recalling that $\|L_{k+1}\|_F = \sqrt{2}\|\widetilde{Y}_{\overline{m}_{k+1}} \underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}\|_F$, it holds

$$\beta_k = \|L_{k+1}\|_F^2 = 2\|\widetilde{Y}_{\overline{m}_{k+1}} \underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}\|_F^2.$$

Then

$$\begin{aligned}
\gamma_k &= \langle \mathcal{R}(X_k), L_{k+1}\rangle_F \\
&= \langle AV_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T + V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T A^T - V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T BB^T V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T + C^TC, \\
&\quad (A - V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T BB^T)V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T + V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T(A - V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T BB^T)^T \\
&\quad + V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T BB^T V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T + C^TC\rangle_F \\
&= \langle V_{\overline{m}_k}\left(T_{\overline{m}_k}Y_{\overline{m}_k} + Y_{\overline{m}_k}T_{\overline{m}_k}^T - Y_{\overline{m}_k}B_{\overline{m}_k}B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T\right)V_{\overline{m}_k}^T \\
&\quad + \mathcal{V}_{\overline{m}_k+1}E_{\overline{m}_k+1}^T\underline{T}_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T + V_{\overline{m}_k}Y_{\overline{m}_k}\underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}\mathcal{V}_{\overline{m}_k+1}^T, \\
&\quad \mathcal{V}_{\overline{m}_{k+1}+1}E_{\overline{m}_{k+1}+1}^T\underline{T}_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T + V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}\underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}\mathcal{V}_{\overline{m}_{k+1}+1}^T\rangle_F \\
&= \langle V_{\overline{m}_k}\left(T_{\overline{m}_k}Y_{\overline{m}_k} + Y_{\overline{m}_k}T_{\overline{m}_k}^T - Y_{\overline{m}_k}B_{\overline{m}_k}B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T\right)V_{\overline{m}_k}^T, \\
&\quad \mathcal{V}_{\overline{m}_{k+1}+1}E_{\overline{m}_{k+1}+1}^T\underline{T}_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T + +V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}\underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}\mathcal{V}_{\overline{m}_{k+1}+1}^T\rangle_F \\
&\quad + \langle \mathcal{V}_{\overline{m}_k+1}E_{\overline{m}_k+1}^T\underline{T}_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T + V_{\overline{m}_k}Y_{\overline{m}_k}\underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}\mathcal{V}_{\overline{m}_k+1}^T, \\
&\quad \mathcal{V}_{\overline{m}_{k+1}+1}E_{\overline{m}_{k+1}+1}^T\underline{T}_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T + V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}\underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}\mathcal{V}_{\overline{m}_{k+1}+1}^T\rangle_F
\end{aligned}$$

The first inner product in the above expression is zero due to the orthogonality of $V_{\overline{m}_k}$ and $\mathcal{V}_{\overline{m}_{k+1}+1}$. The same happens also to the second term if $\overline{m}_{k+1} > \overline{m}_k$. If $\overline{m}_{k+1} = \overline{m}_k$ instead, we have

$$\begin{aligned}
\gamma_k &= \langle \mathcal{V}_{\overline{m}_k+1}E_{\overline{m}_k+1}^T\underline{T}_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T + V_{\overline{m}_k}Y_{\overline{m}_k}\underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}\mathcal{V}_{\overline{m}_k+1}^T, \\
&\quad \mathcal{V}_{\overline{m}_{k+1}+1}E_{\overline{m}_{k+1}+1}^T\underline{T}_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T + V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}\underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}\mathcal{V}_{\overline{m}_{k+1}+1}^T\rangle_F \\
&= \langle E_{\overline{m}_k+1}^T\underline{T}_{\overline{m}_k}Y_{\overline{m}_k} + Y_{\overline{m}_k}\underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}, E_{\overline{m}_{k+1}+1}^T\underline{T}_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}} + \widetilde{Y}_{\overline{m}_{k+1}}\underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}\rangle_F.
\end{aligned}$$

Recalling that

$$Z_k = \widetilde{X}_{k+1} - X_k = V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T - V_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T = V_{\overline{m}_{k+1}}(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}))V_{\overline{m}_{k+1}}^T,$$

we have

$$\begin{aligned}
\delta_k &= \|Z_kBB^TZ_k\|_F^2 \\
&= \|V_{\overline{m}_{k+1}}\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)V_{\overline{m}_{k+1}}^T BB^T V_{\overline{m}_{k+1}}\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)V_{\overline{m}_{k+1}}^T\|_F^2 \\
&= \|\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)B_{\overline{m}_{k+1}}B_{\overline{m}_{k+1}}^T\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)}\right)\|_F^2.
\end{aligned}$$

Moreover,

$$\begin{aligned}
\epsilon_k &= \langle \mathcal{R}(X_k), Z_kBB^TZ_k\rangle_F \\
&= \langle V_{\overline{m}_k}\left(T_{\overline{m}_k}Y_{\overline{m}_k} + Y_{\overline{m}_k}T_{\overline{m}_k}^T - Y_{\overline{m}_k}B_{\overline{m}_k}B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T\right)V_{\overline{m}_k}^T \\
&\quad + \mathcal{V}_{\overline{m}_k+1}E_{\overline{m}_k+1}^T\underline{T}_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T + V_{\overline{m}_k}Y_{\overline{m}_k}\underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}\mathcal{V}_{\overline{m}_k+1}^T, \\
&\quad V_{\overline{m}_{k+1}}\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)B_{\overline{m}_{k+1}}B_{\overline{m}_{k+1}}^T\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)V_{\overline{m}_{k+1}}^T\rangle_F \\
&= \langle V_{\overline{m}_k}\left(T_{\overline{m}_k}Y_{\overline{m}_k} + Y_{\overline{m}_k}T_{\overline{m}_k}^T - Y_{\overline{m}_k}B_{\overline{m}_k}B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T\right)V_{\overline{m}_k}^T, \\
&\quad V_{\overline{m}_{k+1}}\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)B_{\overline{m}_{k+1}}B_{\overline{m}_{k+1}}^T\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)V_{\overline{m}_{k+1}}^T\rangle_F \\
&\quad + \langle \mathcal{V}_{\overline{m}_k+1}E_{\overline{m}_k+1}^T\underline{T}_{\overline{m}_k}Y_{\overline{m}_k}V_{\overline{m}_k}^T + V_{\overline{m}_k}Y_{\overline{m}_k}\underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}\mathcal{V}_{\overline{m}_k+1}^T, \\
&\quad V_{\overline{m}_{k+1}}\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)B_{\overline{m}_{k+1}}B_{\overline{m}_{k+1}}^T\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)V_{\overline{m}_{k+1}}^T\rangle_F.
\end{aligned}$$

If $\overline{m}_{k+1} = \overline{m}_k$, the second inner product above is zero while the first can be written as $\langle T_{\overline{m}_k}Y_{\overline{m}_k} + Y_{\overline{m}_k}T_{\overline{m}_k}^T - Y_{\overline{m}_k}B_{\overline{m}_k}B_{\overline{m}_k}^T Y_{\overline{m}_k} + E_1\boldsymbol{\gamma}\boldsymbol{\gamma}^T E_1^T, (\widetilde{Y}_{\overline{m}_{k+1}} - Y_{\overline{m}_k})B_{\overline{m}_{k+1}}B_{\overline{m}_{k+1}}^T(\widetilde{Y}_{\overline{m}_{k+1}} - Y_{\overline{m}_k})\rangle_F$. If $\overline{m}_{k+1} > \overline{m}_k$, also the second term in the above expression must be taken into account and this can be written as

$$\begin{aligned}
&\langle V_{\overline{m}_k+1}\left(E_{\overline{m}_k+1}E_{\overline{m}_k+1}^T\underline{T}_{\overline{m}_k}[Y_{\overline{m}_k}, O_{2q\overline{m}_k\times 2q}] + [Y_{\overline{m}_k}; O_{2q\times 2q\overline{m}_k}]\underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}E_{\overline{m}_k+1}^T\right)V_{\overline{m}_k+1}^T, \\
&\quad V_{\overline{m}_{k+1}}\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)B_{\overline{m}_{k+1}}B_{\overline{m}_{k+1}}^T\left(\widetilde{Y}_{\overline{m}_{k+1}} - \mathrm{diag}(Y_{\overline{m}_k}, O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)V_{\overline{m}_{k+1}}^T\rangle_F,
\end{aligned}$$

that is

$$\langle E_{\overline{m}_k+1}E_{\overline{m}_k+1}^T\underline{T}_{\overline{m}_k}[Y_{\overline{m}_k},O_{2q\overline{m}_k\times 2q}]+[Y_{\overline{m}_k};O_{2q\times 2q\overline{m}_k}]\underline{T}_{\overline{m}_k}^T E_{\overline{m}_k+1}E_{\overline{m}_k+1}^T,$$
$$V_{\overline{m}_k+1}^T V_{\overline{m}_{k+1}}\left(\widetilde{Y}_{\overline{m}_{k+1}}-\mathrm{diag}(Y_{\overline{m}_k},O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)B_{\overline{m}_{k+1}}B_{\overline{m}_{k+1}}^T\left(\widetilde{Y}_{\overline{m}_{k+1}}-\mathrm{diag}(Y_{\overline{m}_k},O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)V_{\overline{m}_{k+1}}^T V_{\overline{m}_k+1}\rangle_F.$$

Since

$$V_{\overline{m}_k}^T V_{\overline{m}_{k+1}}=[I_{2q\overline{m}_k},O_{2q\overline{m}_k\times 2q(\overline{m}_{k+1}-\overline{m}_k)}],$$

and

$$V_{\overline{m}_k+1}^T V_{\overline{m}_{k+1}}=[I_{2q(\overline{m}_k+1)},O_{2q(\overline{m}_k+1)\times 2q(\overline{m}_{k+1}-\overline{m}_k-1)}],$$

we get the result.

To conclude,

$$\begin{aligned}\zeta_k &= \langle L_{k+1},Z_kBB^TZ_k\rangle_F\\ &= \langle \mathcal{V}_{\overline{m}_{k+1}+1}E_{\overline{m}_{k+1}+1}^T\underline{T}_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}V_{\overline{m}_{k+1}}^T+V_{\overline{m}_{k+1}}\widetilde{Y}_{\overline{m}_{k+1}}\underline{T}_{\overline{m}_{k+1}}^T E_{\overline{m}_{k+1}+1}\mathcal{V}_{\overline{m}_{k+1}+1}^T,\\ &\quad V_{\overline{m}_{k+1}}\left(\widetilde{Y}_{\overline{m}_{k+1}}-\mathrm{diag}(Y_{\overline{m}_k},O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)B_{\overline{m}_{k+1}}B_{\overline{m}_{k+1}}^T\left(\widetilde{Y}_{\overline{m}_{k+1}}-\mathrm{diag}(Y_{\overline{m}_k},O_{2q(\overline{m}_{k+1}-\overline{m}_k)})\right)V_{\overline{m}_{k+1}}^T\rangle_F\\ &= 0,\end{aligned}$$

since $\mathcal{V}_{\overline{m}_{k+1}+1}^T V_{\overline{m}_{k+1}}=O_{2q\times 2q\overline{m}_{k+1}}$. ∎

REFERENCES

[1] L. Amodei and J.-M. Buchot, *An invariant subspace method for large-scale algebraic Riccati equation*, Appl. Numer. Math., 60 (2010), pp. 1067–1082.

[2] A. C. Antoulas, *Approximation of large-scale dynamical systems*, vol. 6 of Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.

[3] H. T. Banks and K. Ito, *A numerical algorithm for optimal feedback gains in high-dimensional linear quadratic regulator problems*, SIAM J. Control Optim., 29 (1991), pp. 499–515.

[4] R. H. Bartels and G. W. Stewart, *Algorithm 432: Solution of the Matrix Equation $AX + XB = C$*, Comm. ACM, 15 (1972), pp. 820–826.

[5] U. Baur, *Low rank solution of data-sparse Sylvester equations*, Numer. Linear Algebra Appl., 15 (2008), pp. 837–851.

[6] U. Baur and P. Benner, *Factorized solution of Lyapunov equations based on hierarchical matrix arithmetic*, Computing, 78 (2006), pp. 211–234.

[7] P. Benner and Z. Bujanović, *On the solution of large-scale algebraic Riccati equations by using low-dimensional invariant subspaces*, Linear Algebra Appl., 488 (2016), pp. 430–459.

[8] P. Benner, Z. Bujanović, P. Kürschner, and J. Saak, *A numerical comparison of solvers for large-scale, continuous-time algebraic Riccati equations*, (2018). Preprint, ArXiv: 1811.00850.

[9] P. Benner, Z. Bujanović, P. Kürschner, and J. Saak, *RADI: a low-rank ADI-type algorithm for large scale algebraic Riccati equations*, Numer. Math., 138 (2018), pp. 301–330.

[10] P. Benner and R. Byers, *An exact line search method for solving generalized continuous-time algebraic Riccati equations*, IEEE Trans. Automat. Control, 43 (1998), pp. 101–107.

[11] P. Benner, M. Heinkenschloss, J. Saak, and H. K. Weichelt, *An inexact low-rank Newton-ADI method for large-scale algebraic Riccati equations*, Appl. Numer. Math., 108 (2016), pp. 125–142.

[12] P. Benner, P. Kürschner, and J. Saak, *Low-rank Newton-ADI methods for large nonsymmetric algebraic Riccati equations*, J. Franklin Inst., 353 (2016), pp. 1147–1167.

[13] P. Benner, J.-R. Li, and T. Penzl, *Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems*, Numerical Linear Algebra with Applications, 15 (2008), pp. 755–777.

[14] P. Benner and J. Saak, *A Galerkin-Newton-ADI method for solving large-scale algebraic Riccati equations*, Preprint, 2010.

[15] D. A. Bini, B. Iannazzo, and F. Poloni, *A fast Newton's method for a nonsymmetric algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 276–290.

[16] T. A. Davis and Y. Hu, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), pp. Art. 1, 25.

[17] V. Druskin, L. Knizhnerman, and V. Simoncini, *Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation*, SIAM J. Numer. Anal., 49 (2011), pp. 1875–1898.

[18] V. Druskin and V. Simoncini, *Adaptive rational Krylov subspaces for large-scale dynamical systems*, Systems Control Lett., 60 (2011), pp. 546–560.

[19] V. Druskin, V. Simoncini, and M. Zaslavsky, *Adaptive tangential interpolation in rational Krylov subspaces for MIMO dynamical systems*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 476–498.

[20] F. Feitzinger, T. Hylla, and E. W. Sachs, *Inexact Kleinman-Newton method for Riccati equations*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 272–288.

[21] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, fourth ed., 2013.
[22] M. H. Gutknecht, *Krylov subspace algorithms for systems with multiple right hand sides: an introduction*, (2006). Available at `http://www.sam.math.ethz.ch/~mhg/pub/delhipap.pdf`.
[23] S. J. Hammarling, *Numerical solution of the stable, nonnegative definite Lyapunov equation*, IMA J. Numer. Anal., 2 (1982), pp. 303–323.
[24] M. Heyouni and K. Jbilou, *An extended block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation*, Electron. Trans. Numer. Anal., 33 (2008/09), pp. 53–62.
[25] M. E. Hochstenbach, *Fields of values and inclusion regions for matrix pencils*, Electron. Trans. Numer. Anal., 38 (2011), pp. 98–112.
[26] I. M. Jaimoukha and E. M. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., 31 (1994), pp. 227–251.
[27] K. Jbilou, *Block Krylov subspace methods for large algebraic Riccati equations*, Numer. Algorithms, 34 (2003), pp. 339–353. International Conference on Numerical Algorithms, Vol. II (Marrakesh, 2001).
[28] D. Kleinman, *On an iterative technique for Riccati equation computations*, IEEE Transactions on Automatic Control, 13 (1968), pp. 114–115.
[29] L. Knizhnerman and V. Simoncini, *Convergence analysis of the extended Krylov subspace method for the Lyapunov equation*, Numer. Math., 118 (2011), pp. 567–586.
[30] J. G. Korvink and E. B. Rudnyi, *Oberwolfach benchmark collection*, in Dimension Reduction of Large-Scale Systems, P. Benner, D. C. Sorensen, and V. Mehrmann, eds., Berlin, Heidelberg, 2005, Springer Berlin Heidelberg, pp. 311–315.
[31] D. Kressner, S. Massei, and L. Robol, *Low-rank updates and divide-and-conquer method for linear matrix equations*, (2017). Preprint, ArXiv: 1712.04349.
[32] P. Lancaster and L. Rodman, *The algebraic Riccati equation*, Oxford University Press, Oxford, 1995.
[33] J.-R. Li and J. White, *Low-rank solution of Lyapunov equations*, SIAM Rev., 46 (2004), pp. 693–713.
[34] Y. Lin and V. Simoncini, *A new subspace iteration method for the algebraic Riccati equation*, Numer. Linear Algebra Appl., 22 (2015), pp. 26–47.
[35] MATLAB, *version 9.3.0 (R2017b)*, The MathWorks Inc., Natick, Massachusetts, 2017.
[36] V. L. Mehrmann, *The autonomous linear quadratic control problem*, vol. 163 of Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin, 1991. Theory and numerical solution.
[37] K. Morris and C. Navasca, *Solution of algebraic Riccati equations arising in control of partial differential equations*, in Control and boundary analysis, vol. 240 of Lect. Notes Pure Appl. Math., Chapman & Hall/CRC, Boca Raton, FL, 2005, pp. 257–280.
[38] P. J. Psarrakos, *Numerical range of linear pencils*, Linear Algebra Appl., 317 (2000), pp. 127–141.
[39] A. Ruhe, *The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: complex shifts for real matrices*, BIT, Vol. 34 (1994), pp. pp. 165–176.
[40] J. Saak, M. Köhler, and P. Benner, *M-M.E.S.S.-1.0.1 - The Matrix Equations Sparse Solvers library*. DOI:10.5281/zenodo.50575, April 2016. See also www.mpi-magdeburg.mpg.de/projects/mess.
[41] V. Simoncini, *Restarted full orthogonalization method for shifted linear systems*, BIT, 43 (2003), pp. 459–466.
[42] ———, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.
[43] ———, *Analysis of the rational Krylov subspace projection method for large-scale algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 1655–1674.
[44] ———, *Computational methods for linear matrix equations*, SIAM Rev., 58 (2016), pp. 377–441.
[45] V. Simoncini, D. B. Szyld, and M. Monsalve, *On two numerical methods for the solution of large-scale algebraic Riccati equations*, IMA J. Numer. Anal., 34 (2014), pp. 904–920.