MAX-PLANCK-GESELLSCHAFT

Max Planck Institute Magdeburg
Preprints

Peter Benner[1] Matthias Heinkenschloss[2] Jens Saak[1]
Heiko K. Weichelt[1]

Inexact Low-Rank Newton-ADI Method for Large-Scale Algebraic Riccati Equations

MAX–PLANCK–INSTITUT
FÜR DYNAMIK KOMPLEXER
TECHNISCHER SYSTEME
MAGDEBURG

MPIMD/15-06 **September 9, 2015**

Affiliations:

[1] Research Group Computational Methods
in Systems and Control Theory (CSC),
Max Planck Institute for Dynamics of
Complex Technical Systems Magdeburg,
Sandtorstr. 1,
39106 Magdeburg, Germany
({benner,saak,weichelt}@mpi-magdeburg.mpg.de)

[2] Department of Computational and
Applied Mathematics (CAAM),
Rice University,
MS-134, 6100 Main Street,
Houston, TX 77005-1892, USA
(heinken@rice.edu)

The research of this author was supported in part by grants AFOSR FA9550-12-1-0155 and NSF DMS-1115345

Corresponding author:
Heiko K. Weichelt[1]
weichelt@mpi-magdeburg.mpg.de

Imprint:

Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Publisher:
Max Planck Institute for
Dynamics of Complex Technical Systems

Address:
Max Planck Institute for
Dynamics of Complex Technical Systems
Sandtorstr. 1
39106 Magdeburg

http://www.mpi-magdeburg.mpg.de/preprints/

Abstract

This paper improves the inexact Kleinman-Newton method for solving algebraic Riccati equations by incorporating a line search and by systematically integrating the low-rank structure resulting from ADI methods for the approximate solution of the Lyapunov equation that needs to be solved to compute the Kleinman-Newton step. A convergence result is presented that tailors the convergence proof for general inexact Newton methods to the structure of Riccati equations and avoids positive semi-definiteness assumptions on the Lyapunov equation residual, which in general do not hold for low-rank approaches. In the convergence proof of this paper, the line search is needed to ensure that the Riccati residuals decrease monotonically in norm. In the numerical experiments, the line search can lead to substantial reduction in the overall number of ADI iterations and, therefore, overall computational cost.

1 Introduction

We present improvements of the inexact Kleinman–Newton method for the solution of large-scale continuous-time algebraic Riccati equations (CARE)

$$\mathcal{R}(X) = C^T C + A^T X + XA - XBB^T X = 0 \tag{1.1}$$

with $C \in \mathbb{R}^{p\times n}$, $A \in \mathbb{R}^{n\times n}$, $X = X^T \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, and $p + r \ll n$. The algorithmic improvements consist of incorporating a line search and of systematically integrating the low-rank structure resulting from ADI methods for the solution of the Lyapunov equation

$$(A^{(k)})^T X^{(k+1)} + X^{(k+1)} A^{(k)} = -C^T C - X^{(k)} BB^T X^{(k)}, \tag{1.2}$$

where

$$A^{(k)} = A - BB^T X^{(k)},$$

which has to be approximately solved in the k-th iteration. The paper is motivated by the recent work of Feitzinger et al. [9] who propose and analyze inexact Kleinman–Newton methods without line search, by Benner and Byers [2] who incorporate line search into the exact Kleinman–Newton method, and by the recent work of Benner et al. [3, 5] on algorithmic improvements of low-rank ADI methods. The convergence result in [9] makes positive semi-definiteness assumptions on the difference between certain matrices and the residual of the Lyapunov equation that are in general not valid when the Lyapunov equation is solved with low-rank methods like, e.g., the low-rank ADI iteration [7]. Our convergence result follows the theory of general inexact Newton methods, but uses the structure of Riccati equations. We add the inexact solution of the Lyapunov equation to [2] and incorporate the low-rank structure.

Our convergence proof makes use of the fact that the Riccati residuals decrease monotonically in norm, which is ensured by the line search. There is no proof that the inexact Kleinman–Newton, low-rank ADI iteration converges globally without line search. On test examples resulting from the finite element approximation of LQR problems governed by an advection diffusion equation, the incorporation of a line search into the inexact Kleinman–Newton, low-rank ADI iteration can lead to substantial reduction in the overall number of ADI iterations and, therefore, overall computational cost.

The paper is organized as follows. In the next section, we recall a basic existence and uniqueness result for the unique symmetric positive semi-definite stabilizing solution of the CARE (1.1). Section 3 introduces the inexact Kleinman–Newton method with line search and presents the basic convergence result. The basic ingredients of ADI methods that are needed for this paper are reviewed in Section 4. Section 5 discusses the efficient computation of various quantities like the Newton residual using the low-rank structure. As a result, the computational cost of our overall algorithm is proportional to the total number of ADI iterations used; in comparison the cost of other components, such as execution of the line search, are negligible. Finally, we demonstrate the contributions of the various improvements on the overall performance gains in Section 6. As mentioned before, in our numerical tests, our improved inexact Kleinman–Newton method is seven to twelve times faster than the exact Kleinman–Newton method without line search.

Notation. Throughout the paper we consider the Hilbert space of matrices in $\mathbb{R}^{n\times n}$ endowed with the inner product $\langle M, N\rangle = \mathrm{tr}\left(M^T N\right) = \sum_{i,j=1}^{n} M_{ij}N_{ij}$ and the corresponding (Frobenius) norm $\|M\|_F = (\langle M, M\rangle)^{1/2} = (\sum_{i,j=1}^{n} M_{ij}^2)^{1/2}$. Furthermore, given real symmetric matrices M, N, we write $M \succeq N$ if and only if $M - N$ is positive semi-definite, and $M \succ N$ if and only if $M - N$ is positive definite. The spectrum of a symmetric matrix M is denoted by $\sigma(M)$.

2 The Riccati Equation

We recall an existence and uniqueness result for the continuous-time Riccati equation (1.1).

Definition 1 *Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, and $C \in \mathbb{R}^{p\times n}$. The pair (A, B) is called stabilizable if there exists a feedback matrix $K \in \mathbb{R}^{n\times r}$ such that $A - BK^T$ is stable, which means that $A - BK^T$ has only eigenvalues in the open left half complex plane $\mathbb{C}^-$. The pair (C, A) is called detectable if (A^T, C^T) is stabilizable.*

Notice that (A, B) is stabilizable if and only if (A, BB^T) is stabilizable and (C, A) is detectable if and only if $(C^T C, A)$ is detectable. Furthermore, we always use the word stable as defined in [15], whereas, in other literature, this is usually called *asymptotically stable*. Since, as in [15], asymptotically stable is the required property in all our applications we do not need to distinguish between stable and asymptotically stable and, therefore, simply use stable everywhere.

Assumption 2 *The matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, and $C \in \mathbb{R}^{p\times n}$ are given such that (A, B) is stabilizable and (C, A) is detectable.*

If Assumption 2 holds, there exists a unique symmetric positive semi-definite solution $X^{(*)}$ of the CARE (1.1) which is also the unique stabilizing solution. This follows from Theorems 8.5.1 and 9.1.2 (see also p. 244) in [15].

Furthermore, it can be shown that all symmetric positive semi-definite solutions of the CARE (1.1) are stabilizing.

Theorem 3 *If Assumption 2 holds, every symmetric solution $X^{(*)} \succeq 0$ of the CARE (1.1) is stabilizing.*

Proof. Let $X = X^T \succeq 0$ solve the CARE (1.1). We show that $A - BB^TX$ is stable by contradiction.

Assume that μ is an eigenvalue of $A - BB^TX$ with $\mathrm{Re}\,(\mu) \geq 0$ and let $v \in \mathbb{C}^n \backslash \{0\}$ be a corresponding eigenvector. The CARE (1.1) can be written as

$$(A - BB^TX)^TX + X(A - BB^TX) = -C^TC - XBB^TX. \tag{2.1}$$

Multiply (2.1) with v^H from the left and v from the right. The left-hand side of (2.1) yields

$$2\,\mathrm{Re}\,(\mu)\, v^HXv \geq 0, \text{ since } X = X^T \succeq 0,$$

whereas the right-hand side of (2.1) yields

$$-v^HC^TCv - v^HXBB^TXv \leq 0, \text{ since } C^TC \succeq 0 \text{ and } XBB^TX \succeq 0.$$

Hence, left- and right-hand sides of (2.1) multiplied by v^H from the left and v from the right are equal to zero, that is $v^HXv = 0$ and $v^HC^TCv + v^HXBB^TXv = 0$, which yields $Cv = 0$ and $B^TXv = 0$. Since $(A - BB^TX)v = \mu v$, it follows that v is an eigenvector of A with eigenvalue μ and $\mathrm{Re}\,(\mu) \geq 0$.

The Hautus-Popov Test for Detectability, e.g., [12, Sec. 80.3], states that (C, A) is detectable if and only if $Ax = \lambda x$, $x \neq 0$ and $\mathrm{Re}\,(\lambda) \geq 0$ implies $Cx \neq 0$. Thus, the existence of $v \neq 0$ and $\mathrm{Re}\,(\mu) \geq 0$ with $Av = \mu v$ contradicts the detectability of (C, A) by the Hautus-Popov test.

□

3 The Inexact Kleinman–Newton Method with Line Search

This section introduces the inexact Kleinman–Newton method with line search and gives a convergence result. The fundamental ideas are identical to what is well known for inexact Newton methods, see, e.g., Kelley [13, Sec. 8.2], but are tailored to the structure of the Riccati equations. The presentation of the basic algorithm in Section 3.1 combines ideas from general inexact Newton methods, from Kleinman–Newton with inexactness, see, e.g., Feitzinger et al. [9], and Kleinman–Newton with line search, see, e.g., Benner and Byers [2]. In Section 3.3, we will show that the assumptions made in Feitzinger et al. [9], to ensure convergence of the inexact Kleinman Newton method, are in general not valid if low-rank Lyapunov solvers are used to compute the inexact Kleinman–Newton step, and we will present an alternative convergence result that follows more closely that of general inexact Newton methods.

3.1 Derivation of the Method

We want to compute the symmetric, positive semi-definite, stabilizing solution $X^{(*)}$ of the CARE (1.1). The operator $\mathcal{R} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ defined in (1.1) is twice Fréchet differentiable with derivatives given by

$$\mathcal{R}'(X)N = (A - BB^TX)^TN + N(A - BB^TX), \tag{3.1a}$$

$$\mathcal{R}''(X)(N_1, N_2) = -N_1BB^TN_2 - N_2BB^TN_1. \tag{3.1b}$$

Since $\mathcal{R}$ is quadratic in X, the 2nd order Fréchet derivative is independent of X and $\mathcal{R}(Y)$ can be expressed via a Taylor series as

$$\mathcal{R}(Y) = \mathcal{R}(X) + \mathcal{R}'(X)(Y - X) + \frac{1}{2}\mathcal{R}''(X)(Y - X, Y - X). \tag{3.2}$$

The CARE (1.1) can be solved using Newton's method, which in this context is referred to as the Kleinman–Newton method [14]. Given an approximate symmetric solution $X^{(k)}$ of (1.1), the new Kleinman–Newton iterate is given by

$$\mathcal{R}'(X^{(k)})X^{(k+1)} = \mathcal{R}'(X^{(k)})X^{(k)} - \mathcal{R}(X^{(k)}). \tag{3.3}$$

Equation (3.3) is the Lyapunov equation (1.2). Instead of solving (3.3) for the new iterate, one could solve $\mathcal{R}'(X^{(k)})S^{(k)} = -\mathcal{R}(X^{(k)})$ for the step $S^{(k)} = X^{(k+1)} - X^{(k)}$. While the latter equation may be favorable in cases where the Lyapunov equation is solved using direct methods (see, e.g., [2, p. 101]), (3.3) is favorable when the Lyapunov equation is solved inexactly using iterative methods. The right hand side in (3.3) is $-X^{(k)}BB^TX^{(k)} - C^TC = -GG^T$, where $G = [C^T \mid X^{(k)}B] \in \mathbb{R}^{n\times(p+r)}$. As we will see later, this low-rank factorization ($p + r \ll n$) is important when the Kleinman–Newton method is combined with low-rank approximation methods. Expressions of $\mathcal{R}(X^{(k)})$ which lead themselves to the application of low-rank approximation methods, and which are equal to $\mathcal{R}(X^{(k)})$ in the exact Kleinman–Newton method, fail when used in an inexact Kleinman–Newton method as shown in Feitzinger et al. [9].

If Assumption 2 holds, then the special structure of $\mathcal{R}$ allows one to prove global convergence of the Kleinman–Newton method: If the initial iterate $X^{(0)}$ is symmetric and stabilizing, then the Kleinman–Newton method is well defined (i.e., (1.2) has a unique solution), the iterates generated by the Kleinman–Newton method converge with a q-quadratic convergence rate, and satisfy $X^{(1)} \succeq X^{(2)} \succeq \ldots \succeq X^{(*)} \succeq 0$; see, e.g., Kleinman [14] or Lancaster and Rodman [15, Sec. 9.2].

Even though the Kleinman–Newton method exhibits global convergence, it was observed by Benner and Byers [2] that a line search improves its efficiency. Especially in the first iteration of the Kleinman–Newton method, the residual may increase dramatically if no line search is used. For large scale problems, the Newton equation (the Lyapunov equation) (3.3) is solved iteratively, and the residual error in the Lyapunov equation has to be controlled appropriately to ensure convergence. We integrate the inexact solution of (3.3) and a line search into the Kleinman–Newton method. As we have mentioned before, the fundamental ideas are identical to what is well known for inexact Newton methods, see, e.g., Kelley [13, Sec. 8.2].

Given a symmetric $X^{(k)} \in \mathbb{R}^{n\times n}$ and $\alpha > 0$, $\eta_k \in (0,1)$, we compute a symmetric step $S^{(k)} \in \mathbb{R}^{n\times n}$ with

$$\left\|\mathcal{R}'(X^{(k)})S^{(k)} + \mathcal{R}(X^{(k)})\right\|_F \leq \eta_k \left\|\mathcal{R}(X^{(k)})\right\|_F \tag{3.4}$$

and then compute the next iterate

$$X^{(k+1)} = X^{(k)} + \lambda_k S^{(k)},$$

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

$$\left\|\mathcal{R}\left(X^{(k)} + \lambda_k S^{(k)}\right)\right\|_F \leq (1-\lambda_k\alpha)\left\|\mathcal{R}\left(X^{(k)}\right)\right\|_F \tag{3.5}$$

is satisfied and the step size λ_k is not unnecessarily small.

If we define the Newton step residual

$$\mathcal{R}'(X^{(k)})S^{(k)} + \mathcal{R}(X^{(k)}) = L^{(k+1)}, \tag{3.6}$$

then (3.4) reads

$$\left\|L^{(k+1)}\right\|_F \leq \eta_k \left\|\mathcal{R}(X^{(k)})\right\|_F. \tag{3.7}$$

Using the definition (1.1), (3.1a), and

$$\widetilde{X}^{(k+1)} = X^{(k)} + S^{(k)},$$

the equation (3.6) is equivalent to

$$(A^{(k)})^T\widetilde{X}^{(k+1)} + \widetilde{X}^{(k+1)}A^{(k)} = -X^{(k)}BB^TX^{(k)} - C^TC + L^{(k+1)} \tag{3.8a}$$

and the new iterate is

$$X^{(k+1)} = (1-\lambda_k)X^{(k)} + \lambda_k\widetilde{X}^{(k+1)}. \tag{3.8b}$$

The Riccati residual at $X^{(k+1)} = X^{(k)} + \lambda_k S^{(k)}$ can be expressed using (3.2) and (3.6) as

$$\begin{aligned}\mathcal{R}(X^{(k)} + \lambda_k S^{(k)}) &= \mathcal{R}(X^{(k)}) + \lambda_k\mathcal{R}'(X^{(k)})S^{(k)} + \frac{\lambda_k^2}{2}\mathcal{R}''(X^{(k)})(S^{(k)},S^{(k)})\\ &= (1-\lambda_k)\mathcal{R}(X^{(k)}) + \lambda_k L^{(k+1)} - \lambda_k^2 S^{(k)}BB^TS^{(k)}.\end{aligned} \tag{3.9}$$

Algorithm 1 Inexact Kleinman-Newton Method with Line Search

```
Input: $A$, $B$, $C$, stabilizing initial iterate $X^{(0)}$, $tol_{\text{Newton}} > 0$, $\bar{\eta} \in (0,1)$ and $\alpha \in (0, 1-\bar{\eta})$.
Output: Approximate solution of the CARE (1.1).
for $k = 0, 1, \ldots$ do
    if $\|\mathcal{R}(X^{(k)})\| \leq tol_{\text{Newton}}$ then
        Return $X^{(k)}$ as an approximate solution of the CARE (1.1).
    end if
    Set $A^{(k)} = \left(A - BB^T X^{(k)}\right)$, $G = \left[C^T \,|\, X^{(k)}B\right]$.
    Select $\eta_k \in (0, \bar{\eta}]$.
    Compute an approximate solution $\widetilde{X}^{(k+1)}$ of the Lyapunov equation such that
        $$(A^{(k)})^T \widetilde{X}^{(k+1)} + \widetilde{X}^{(k+1)} A^{(k)} = -GG^T + L^{(k+1)}$$
    and $\|L^{(k+1)}\|_F \leq \eta_k \|\mathcal{R}(X^{(k)})\|_F$.
    Set $S^{(k)} = \widetilde{X}^{(k+1)} - X^{(k)}$.
    Compute $\lambda_k > 0$ such that $\|\mathcal{R}(X^{(k)} + \lambda_k S^{(k)})\|_F \leq (1 - \lambda_k \alpha)\|\mathcal{R}(X^{(k)})\|_F$.
    Set $X^{(k+1)} = X^{(k)} + \lambda_k S^{(k)}$.
end for
```

Therefore, if $\eta_k \leq \bar{\eta} < 1$ and $\alpha \in (0, 1-\bar{\eta})$, then (3.7) and (3.9) imply

$$\begin{aligned}
&\|\mathcal{R}(X^{(k)} + \lambda S^{(k)})\|_F \\
&\leq (1-\lambda)\|\mathcal{R}(X^{(k)})\|_F + \lambda \|L^{(k+1)}\|_F + \lambda^2 \|S^{(k)}BB^T S^{(k)}\|_F \\
&\leq (1 - \lambda + \lambda\bar{\eta})\|\mathcal{R}(X^{(k)})\|_F + \lambda^2 \frac{\|S^{(k)}BB^T S^{(k)}\|_F}{\|\mathcal{R}(X^{(k)})\|_F} \|\mathcal{R}(X^{(k)})\|_F \\
&\leq (1 - \alpha\lambda)\|\mathcal{R}(X^{(k)})\|_F
\end{aligned}$$

for all λ with

$$0 < \lambda \leq (1 - \alpha - \bar{\eta}) \frac{\|\mathcal{R}(X^{(k)})\|_F}{\|S^{(k)}BB^T S^{(k)}\|_F}. \tag{3.10}$$

In particular, the sufficient decrease condition (3.5) is satisfied for all λ_k with (3.10).

In the actual computation of the step size λ_k we use (3.9) which implies that

$$\begin{aligned}
f(\lambda) &= \|\mathcal{R}(X^{(k)} + \lambda S^{(k)})\|_F^2 \qquad (3.11) \\
&= (1-\lambda)^2 \alpha^{(k)} + \lambda^2 \beta^{(k)} + \lambda^4 \delta^{(k)} + 2\lambda(1-\lambda)\gamma^{(k)} - 2\lambda^2(1-\lambda)\varepsilon^{(k)} - 2\lambda^3 \zeta^{(k)}
\end{aligned}$$

is a quartic polynomial with

$$\begin{aligned}
\alpha^{(k)} &= \|\mathcal{R}(X^{(k)})\|_F^2, & \delta^{(k)} &= \|S^{(k)}BB^T S^{(k)}\|_F^2, \\
\beta^{(k)} &= \|L^{(k+1)}\|_F^2, & \varepsilon^{(k)} &= \langle \mathcal{R}(X^{(k)}), S^{(k)}BB^T S^{(k)} \rangle, \\
\gamma^{(k)} &= \langle \mathcal{R}(X^{(k)}), L^{(k+1)} \rangle, & \zeta^{(k)} &= \langle L^{(k+1)}, S^{(k)}BB^T S^{(k)} \rangle.
\end{aligned} \tag{3.12}$$

The derivative is $f'(\lambda) = \langle \mathcal{R}(X^{(k)} + \lambda S^{(k)}), -\mathcal{R}(X^{(k)}) + L^{(k+1)} - 2\lambda S^{(k)} BB^T S^{(k)} \rangle$. In particular, using the Cauchy–Schwarz inequality and (3.7), we find $f'(0) < 0$, which again confirms that $S^{(k)}$ is a descent direction.

Remark 4 *If the current iterate $X^{(k)}$ is symmetric positive semi-definite, if the solution $\widetilde{X}^{(k+1)}$ of (3.8a) is symmetric positive semi-definite, and if $\lambda_k \in (0, 1]$, then $X^{(k+1)} = X^{(k)} + \lambda_k(\widetilde{X}^{(k+1)} - X^{(k)})$ is also symmetric positive semi-definite.*

The basic inexact Kleinman–Newton method with line search is summarized in Algorithm 1.

3.2 Line Search

There are many possibilities to compute a step size λ_k that satisfies the sufficient decrease condition (3.5). We review two. In both cases, the representation (3.11) of the Riccati residual as a quartic polynomial can be used for the efficient implementation of the respective procedure.

3.2.1 Armijo Rule

Given $\beta \in (0, 1)$, the Armijo rule in its simplest form selects $\lambda_k = \beta^\ell$, where ℓ is the smallest integer such that the sufficient decrease condition (3.5) is satisfied. See Kelley [13, Sec. 8.2] for more details. Since the sufficient decrease condition (3.5) is satisfied for all step sizes with (3.10) and ℓ is the smallest integer such that $\lambda_k = \beta^\ell$ satisfies (3.5), the step size λ_k generated by the Armijo rule satisfies

$$\lambda_k > \beta(1 - \alpha - \bar{\eta}) \frac{\|\mathcal{R}(X^{(k)})\|_F}{\|S^{(k)} BB^T S^{(k)}\|_F}. \tag{3.13}$$

Using the structure of the CARE (1.1) we can bound the right hand side in (3.13).

Theorem 5 *Assume that $A^{(k)}$ is stable and let r_k denote the rank of $S^{(k)}$. If the forcing parameter η_k that controls the size of the Lyapunov residual, see (3.7), satisfies $\eta_k \le \bar{\eta} < 1$, then the step size generated by the Armijo rule obeys*

$$\lambda_k > \frac{\beta(1 - \alpha - \bar{\eta})}{r_k(1 + \bar{\eta})^2} \frac{1}{\|BB^T\|_F \|\mathcal{R}(X^{(k)})\|_F \int_0^\infty \|\exp(A^{(k)}t)\|_2^2 \, dt}. \tag{3.14}$$

Proof. First we bound the step $S^{(k)}$, the solution of (3.6). Since $A^{(k)}$ is stable, the step is given by $S^{(k)} = \int_0^\infty \exp((A^{(k)})^T t) \Big(\mathcal{L}^{(k+1)} - \mathcal{R}(X^{(k)}) \Big) \exp((A^{(k)})t) \, dt$. Therefore,

$$\|S^{(k)}\|_2 \le \|\mathcal{L}^{(k+1)} - \mathcal{R}(X^{(k)})\|_2 \int_0^\infty \|\exp(A^{(k)}t)\|_2^2 \, dt$$

and

$$\|S^{(k)}\|_F \le \sqrt{r_k} \|\mathcal{L}^{(k+1)} - \mathcal{R}(X^{(k)})\|_F \int_0^\infty \|\exp(A^{(k)}t)\|_2^2 \, dt,$$

since for a square matrix M with rank r, $\|M\|_2 \le \|M\|_F \le \sqrt{r}\|M\|_2$.

The bound on $S^{(k)}$, (3.6), (3.7), and $\eta_k \le \bar{\eta} < 1$ imply

$$\begin{aligned}\|S^{(k)}BB^TS^{(k)}\|_F &\le \|BB^T\|_F\|S^{(k)}\|_F^2 \\ &\le r_k(1+\bar{\eta})^2\|BB^T\|_F\|\mathcal{R}(X^{(k)})\|_F^2 \int_0^\infty \|\exp(A^{(k)}t)\|_2^2\,dt.\end{aligned}$$

Inserting this into (3.13) gives the desired lower bound (3.14).

□

Remark 6 *If $A^{(k)}$ is a normal matrix with spectral abscissa $\alpha(A^{(k)}) := \max\{\mathrm{Re}\,(\mu) : \mu \in \sigma(A^{(k)})\}$, then $\|\exp(A^{(k)}t)\|_2 \le \exp(t\alpha(A^{(k)}))$. If $\alpha(A^{(k)}) < 0$, then*

$$\int_0^\infty \|\exp(A^{(k)}t)\|_2^2\,dt \le 1/(2|\alpha(A^{(k)})|).$$

Otherwise we can use the ϵ-pseudospectrum $\sigma_\epsilon(A^{(k)})$ of $A^{(k)}$ and the ϵ-pseudospectral abscissa $\alpha_\epsilon(A^{(k)}) := \sup\{\mathrm{Re}\,(\mu) : \mu \in \sigma_\epsilon(A^{(k)})\}$. If $\sigma_\epsilon(A^{(k)})$ has a boundary with finite arc length $L_{\epsilon,k}$, then

$$\|\exp(A^{(k)}t)\|_2 \le \frac{L_{\epsilon,k}\exp(t\,\alpha_\epsilon(A^{(k)}))}{2\pi\epsilon} \qquad \epsilon > 0, t \ge 0$$

[22, p. 139], and, if $\alpha_\epsilon(A^{(k)}) < 0$, then

$$\int_0^\infty \|\exp(A^{(k)}t)\|_2^2\,dt \le \frac{L_{\epsilon,k}^2}{8\pi^2\epsilon^2|\alpha_\epsilon(A^{(k)})|}.$$

3.2.2 Exact Line Search

Equation (3.11) shows that $\mathcal{R}(X^{(k)}+\lambda S^{(k)})$ is quadratic in λ. Hence, $\min_{\lambda>0}\|\mathcal{R}(X^{(k)}+\lambda S^{(k)})\|_F^2$ corresponds to the minimization of the quartic polynomial (3.11). For the Kleinman–Newton method with exact Lyapunov equation solves, $L^{(k+1)} = 0$, the exact line search is analyzed by Benner and Byers [2]. In particular, they show that there is a local minimum $\lambda_k \in (0, 2]$, and that if $A^{(k)}$ is stable and $X^{(k+1)}$ is computed with a step length $\lambda_k \in (0, 2]$, then $A^{(k+1)}$ is also stable. However, both results are no longer true, in general, for the inexact case.

3.3 Convergence

Feitzinger et al. [9] extend the convergence results for the Kleinman–Newton method with step size $\lambda_k = 1$ to the inexact case, provided the Lyapunov residual $L^{(k+1)}$ satisfies certain positive semi-definiteness assumptions. The first result establishes the well-posedness of the inexact Kleinman–Newton method.

Theorem 7 ([9, Thm. 4.3]) *Let $X^{(k)}$ be symmetric and positive semi-definite such that $A - BB^T X^{(k)}$ is stable and*

$$L^{(k+1)} \preceq C^T C \tag{3.15}$$

holds. Then

(i) the iterate $X^{(k+1)} = \widetilde{X}^{(k+1)}$ of the inexact Kleinman–Newton method with step-size $\lambda_k = 1$ is well defined, symmetric and positive semi-definite,

(ii) and the matrix $A - BB^T X^{(k+1)}$ is stable.

We will use the low-rank ADI method, see, e.g., [16], to approximately solve the Lyapunov equation, see Section 4. This means that in our algorithm $L^{(k+1)}$, $X^{(k)}$, and other matrices are low-rank. In particular, we will see in Section 4 (see equation (4.4)), $L^{(k+1)} = W^{(k+1)}(W^{(k+1)})^T = \mathfrak{F} G G^T \mathfrak{F}^T$, where $\mathfrak{F}$ is a matrix with spectrum inside the unit ball and $G = [C^T | X^{(k)} B]$.

Lemma 8 *If M, N are symmetric positive semi-definite matrices with $M \succeq N$, then* $\ker(M) \subset \ker(N)$ *and* $\operatorname{range}(N) \subset \operatorname{range}(M)$.

Proof. Assume there exists $v \in \ker(M)$ with $v \notin \ker(N)$, then $v^T M v - v^T N v = -v^T N v < 0$, which contradicts $M \succeq N$. Hence, $\operatorname{range}(M)^\perp = \ker(M) \subset \ker(N) = \operatorname{range}(N)^\perp$ and, consequently, $\operatorname{range}(N) \subset \operatorname{range}(M)$.

□

The definition of G and application of the previous lemma give that $C^T C \succeq L^{(k+1)} = \mathfrak{F} G G^T \mathfrak{F}^T$ implies $\operatorname{range}\left(\mathfrak{F} C^T\right) \subset \operatorname{range}(\mathfrak{F} G) \subset \operatorname{range}\left(C^T\right) \subset \operatorname{range}(G)$. However, the invariance property $\operatorname{range}(\mathfrak{F} G) \subset \operatorname{range}\left(C^T\right)$, or even $\operatorname{range}\left(\mathfrak{F} C^T\right) \subset \operatorname{range}\left(C^T\right)$, is typically not satisfied. Recall that $C^T \in \mathbb{R}^{n \times p}$ while $\mathfrak{F} G \in \mathbb{R}^{n \times (p+r)}$ for $k > 1$. Therefore, in general $C^T C \not\succeq L^{(k+1)}$.

Under an additional semidefiniteness condition on the Lyapunov residual, Feitzinger et al. [9] prove quadratic convergence of the inexact Kleinman–Newton method.

Theorem 9 ([9, Thm. 4.4]) *Let Assumption 2 be satisfied and let $X^{(0)}$, symmetric and positive semi-definite, be such that $A - BB^T X^{(0)}$ is stable. Let (3.15) hold for all $k \in \mathbb{N}$, and let $X^{(k)}$ be the iterates of the inexact Kleinman–Newton method with step size $\lambda_k = 1$. If*

$$0 \preceq L^{(k+1)} \preceq (X^{(k+1)} - X^{(k)}) BB^T (X^{(k+1)} - X^{(k)}) \tag{3.16}$$

hold for all $k \in \mathbb{N}$, then the iterates of inexact Kleinman–Newton (3.8) with step size $\lambda_k = 1$ satisfy

(i) $\lim_{k\to\infty} X^{(k)} = X^\infty$ and $0 \preceq X^\infty \preceq \cdots \preceq X^{(k+1)} \preceq X^{(k)} \preceq \cdots \preceq X^{(1)}$,

(ii) $(A - BB^T X^\infty)$ is stable and X^∞ is the maximal solution of $\mathcal{R}(X) = 0$,

(iii) $\|X^{(k+1)} - X^{\infty}\|_F \leq c\|X^{(k)} - X^{\infty}\|_F^2, k \in \mathbb{N}$.

The condition (3.16) implies the monotonicity $0 \preceq \cdots \preceq X^{(k+1)} \preceq X^{(k)} \preceq \cdots \preceq X^{(1)}$, which implies convergence of the sequence of the iterates. See the proof of [9, Thm. 4.4]. It is also interesting to note that under the condition (3.16), the inexact Kleinman–Newton method convergences q–quadratically, independent of how the forcing parameter η_k in (3.4) is chosen. Unfortunately, the semidefiniteness condition (3.16) implies $\operatorname{range}(\mathfrak{F}\,G) \subset \operatorname{range}\big((X^{(k+1)} - X^{(k)})B\big)$, which is generally not satisfied. Therefore, the convergence analysis in [9] is not applicable if the low-rank ADI method, or any other low-rank solver, is used to approximately solve the Lyapunov equation.

Our convergence proof follows that of inexact Newton methods, see, e.g., Kelley [13, Sec. 8.2]. First, we prove $\|\mathcal{R}(X^{(k)})\|_F \to 0$ and then we use the structure of the Riccati equations to argue convergence of $\{X^{(k)}\}$. In particular, Benner and Byers [2, Lem. 6] prove that if (A, B) is controllable and $\{\mathcal{R}(X^{(k)})\}$ is bounded, then $\{X^{(k)}\}$ is also bounded. Since controllability of (A, B) implies stabilizability of (A, B), the assumption of controllability is stronger than Assumption 2. Guo and Laub [10] removed the controllability assumption and showed that if (A, B) is stabilizable, $\{\mathcal{R}(X^{(k)})\}$ is bounded, and the matrices $A^{(k)}$ are stable, then $\{X^{(k)}\}$ is also bounded.

The papers [14] on exact Kleinman–Newton, [2] on Kleinman–Newton with line search and [9] on inexact Kleinman–Newton contain proofs that the matrices $A^{(k)}$ corresponding to the iterates $X^{(k)}$ are stable, provided that $A^{(0)}$ is stable. This implies the unique solution of the Lyapunov equation (1.2) and, therefore, the well-posedness of the respective method. Since the definiteness assumption in [9, Thm. 4.3] typically does not hold in the low-rank case, there is no result yet on the well-posedness of the inexact Kleinman–Newton method and we have to assume existence of $\widetilde{X}^{(k+1)}$ such that (3.8a) and (3.7) are satisfied.

Theorem 10 *Let Assumption 2 be satisfied and assume that for all k there exists a symmetric positive semi-definite $\widetilde{X}^{(k+1)}$ such that (3.8a) and (3.7) hold.*

(i) If the step sizes are bounded away from zero, $\lambda_k \geq \lambda_{\min} > 0$ for all k, then $\|\mathcal{R}(X^{(k)})\|_F \to 0$.

(ii) If, in addition to (i), the matrices $A^{(k)}$ are stable for $k \geq K_0$, and $X^{(k)} \succeq 0$ for all $k \geq K_0$, then $X^{(k)} \to X^{(*)}$, where $X^{(*)} \succeq 0$ is the unique stabilizing solution of the CARE.

Proof. (i) The first part is a standard line search argument. The sufficient decrease condition (3.5) implies that for any integer K,

$$\begin{aligned}\|\mathcal{R}(X^{(0)})\|_F &\geq \|\mathcal{R}(X^{(0)})\|_F - \|\mathcal{R}(X^{(K+1)})\|_F \\ &= \sum_{k=0}^{K} \|\mathcal{R}(X^{(k)})\|_F - \|\mathcal{R}(X^{(k+1)})\|_F \geq \sum_{k=0}^{K} \lambda_k \alpha \|\mathcal{R}(X^{(k)})\|_F \geq 0.\end{aligned}$$

Taking the limit $K \to \infty$ and using $\lambda_k \geq \lambda_{\min} > 0$ implies $\|\mathcal{R}(X^{(k)})\|_F \to 0$.

(ii) If the matrices $A^{(k)}$ are stable for $k \geq K_0$ and $\{\mathcal{R}(X^{(k)})\}$ is bounded, [10, Lem. 2.3] guarantees that $\{X^{(k)}\}$ is bounded. Hence, $\{X^{(k)}\}$ has a converging subsequence. For any converging subsequence $\lim_j X^{(k_j)} \succeq 0$ and $0 = \lim_j \|\mathcal{R}(X^{(k_j)})\|_F = \|\mathcal{R}(\lim_j X^{(k_j)})\|_F$. Since the symmetric positive semi-definite solution of the CARE (1.1) is unique and stabilizing, every converging subsequence of $\{X^{(k)}\}$ has the same limit $X^{(*)}$. Therefore, the entire sequence converges.

□

Remark 11 *1. If the step size $\lambda_k \in (0, 1]$, then $X^{(k)} \succeq 0$ for all k, see Remark 4.*

2. Lower bounds for the step size computed by the Armijo rule are established in Theorem 5. In particular if

$$\left\{ \|\mathcal{R}(X^{(k)})\|_F \int_0^\infty \|\exp(A^{(k)}t)\|_2^2 \, dt \right\}_{k \in \mathbb{N}}, \tag{3.17}$$

is bounded, the step size is bounded away from zero. Since $\|\mathcal{R}(X^{(k)})\|_F < \|\mathcal{R}(X^{(0)})\|_F$, the sequence (3.17) is bounded, if $\int_0^\infty \|\exp(A^{(k)}t)\|_2^2 \, dt$ is bounded, which is a condition on the uniform stability of the matrices $A^{(k)}$, $k \in \mathbb{N}$.

As it is well known for inexact Newton methods (see, e.g., Kelley [13, Sec. 8.2]), the specific choice of the forcing parameter η_k in (3.4) determines the rate if convergence. In particular, if $\eta_k \to 0$ the inexact Kleinman Newton method converges superlinearly (under the assumptions of Theorem 10) and if $\eta_k = O(\|\mathcal{R}(X^{(k_j)})\|_F)$, the convergence is quadratic.

4 ADI Method

To compute the new iterate $X^{(k+1)}$ within the Kleinman–Newton method one has to solve the Lyapunov equation (1.2). A powerful approach to solve such large-scale Lyapunov equations with low-rank right-hand sides is the *alternating directions implicit* (ADI) method, see, e.g., [7, 16]. In this section, we review the basic ingredients of the ADI method combined with recent algorithmic improvements from [3, 5, 4]. To simplify the notation, we drop the index k and write (1.2) in a more general form as

$$FX + XF^T = -GG^T \tag{4.1}$$

with $G := \left[C^T \,|\, X^{(k)}B\right] \in \mathbb{R}^{n \times (p+r)}$ and $F = (A - BB^TX^{(k)})^T \in \mathbb{R}^{n \times n}$. We assume that F is stable. The original low-rank ADI method computes a low-rank solution factor $\widehat{Z} \in \mathbb{C}^{n \times (\ell(p+r))}$ such that $\widehat{Z}\widehat{Z}^H \approx X \in \mathbb{R}^{n \times n}$ is the approximated solution of the Lyapunov equation (4.1); see, e.g., [7]. For given ADI shifts $\{q_1, \ldots, q_\ell\} \in \mathbb{C}^-$, the low-rank ADI method successively computes

$$\widehat{V}_1 = (F + q_1 I)^{-1} G \in \mathbb{C}^{n \times (p+r)}, \tag{4.2a}$$

$$\widehat{V}_\ell = \widehat{V}_{\ell-1} - (q_\ell + \overline{q}_{\ell-1})(F + q_\ell I)^{-1} \widehat{V}_{\ell-1} \in \mathbb{C}^{n \times (p+r)}, \quad \ell \geq 2. \tag{4.2b}$$

Algorithm 2 real low-rank ADI method [5]

```
Input: $F, G, tol_{\mathrm{ADI}}$, shifts $q_\ell \in \mathbb{C}^-$.
Output: $Z$ such that $ZZ^T \approx X$ solves Eq. (4.1).
Set $\ell = 1$, $Z = [\,]$, $W_0 = G$.
while $\|W_{\ell-1}^T W_{\ell-1}\|_F > tol_{\mathrm{ADI}}$ do
    Solve $V = (F + q_\ell I)^{-1} W_{\ell-1}$.
    if $\mathrm{Im}\,(q_\ell) = 0$ then
        $W_\ell = W_{\ell-1} - 2q_\ell V$
        $\widetilde{V} = \sqrt{-2q_\ell}\, V$
    else
        $\gamma_\ell = 2\sqrt{-\mathrm{Re}\,(q_\ell)}, \quad \delta_\ell = \mathrm{Re}\,(q_\ell)\,/\,\mathrm{Im}\,(q_\ell)$
        $W_{\ell+1} = W_{\ell-1} + \gamma_\ell^2 (\mathrm{Re}\,(V) + \delta_\ell\, \mathrm{Im}\,(V))$
        $\widetilde{V} = \left[\gamma_\ell\,(\mathrm{Re}\,(V) + \delta_\ell\, \mathrm{Im}\,(V)) \mid \gamma_\ell \sqrt{(\delta_\ell^2 + 1)}\, \mathrm{Im}\,(V)\right]$
        $\ell = \ell + 1$
    end if
    $Z = \left[Z \mid \widetilde{V}\right]$
    $\ell = \ell + 1$
end while
```

In the ℓ-th iteration, the approximate low-rank solution factor is

$$\widehat{Z}_\ell = \left[\sqrt{-2\,\mathrm{Re}\,(q_1)}\widehat{V}_1, \ldots, \sqrt{-2\,\mathrm{Re}\,(q_\ell)}\widehat{V}_\ell\right] \in \mathbb{C}^{n \times (\ell \cdot (p+r))}. \tag{4.3}$$

We use two important modifications of the original ADI method, which are due to Benner et al. [3, 5, 4]. The first reorganizes the computation of the $\widehat{V}_\ell$'s to obtain a low-rank representation of the Lyapunov residual in the ADI iterations. The second exploits the fact that the ADI shifts need to occur either as a real number $q_\ell \in \mathbb{R}^-$ or as a pair of complex conjugate numbers $q_\ell \in \mathbb{C}^-, q_{\ell+1} = \overline{q_\ell}$, to write all matrices in the ADI iterations as real matrices. We summarize the main ideas.

In [4, Sec. 4.2], Benner et al. introduced a novel low-rank residual formulation for ADI. For $\ell \geq 2$ the identity (4.2b) can be written as

$$\begin{aligned}\widehat{V}_\ell &= (I - (q_\ell + \overline{q_{\ell-1}})(F + q_\ell I)^{-1})\widehat{V}_{\ell-1} = (F - \overline{q_{\ell-1}} I)(F + q_\ell I)^{-1}\widehat{V}_{\ell-1} \\ &= \Big(\prod_{j=2}^{\ell}(F - \overline{q_{j-1}} I)(F + q_j I)^{-1}\Big)(F + q_1 I)^{-1} G.\end{aligned}$$

Because $(F \pm qI)$ and $(F + \hat{q}I)^{-1}$ commute for all $q, \hat{q} \in \mathbb{C}\backslash\sigma(F)$, these products can be regrouped to yield

$$\widehat{V}_\ell = (F + q_\ell I)^{-1}\Big(\prod_{j=1}^{\ell-1}(F - \overline{q_j} I)(F + q_j I)^{-1}\Big)G =: (F + q_\ell I)^{-1}\widehat{W}_{\ell-1}.$$

By definition of $\widehat{W}_\ell$ (and setting $\widehat{W}_0 = G$),

$$\begin{aligned}\widehat{W}_\ell &= (F - \overline{q_\ell} I)\widehat{V}_\ell = (F - \overline{q_\ell})(F + q_\ell I)^{-1}\widehat{W}_{\ell-1} \\ &= (I - 2\,\mathrm{Re}\,(q_\ell)\,(F + q_\ell I)^{-1})\widehat{W}_{\ell-1} = \widehat{W}_{\ell-1} - 2\,\mathrm{Re}\,(q_\ell)\,\widehat{V}_\ell \in \mathbb{C}^{n\times(p+r)}.\end{aligned}$$

Moreover,

$$\widehat{W}_\ell = (F - \overline{q_\ell} I)\widehat{V}_\ell = \prod_{j=1}^{\ell}(F - \overline{q_j} I)(F + q_j I)^{-1}\, G = \widehat{\mathfrak{F}}\, G$$

with $\widehat{\mathfrak{F}} = \widehat{\mathfrak{F}}(F, q_1, \ldots, q_\ell) := \prod_{j=1}^{\ell}(F - \overline{q_j} I)(F + q_j I)^{-1}$ an analytic matrix function depending on F and the ADI shifts $q_1, \ldots, q_\ell$. Using this formulation, which is mathematically equivalent to the original algorithm in [7, 16], Benner et al. [4, Sec. 4.2] show that the Lyapunov residual after ADI step ℓ, can be written as

$$L_\ell = F\widehat{Z}_\ell\widehat{Z}_\ell^H + \widehat{Z}_\ell\widehat{Z}_\ell^H F^T + GG^T = \widehat{W}_\ell\widehat{W}_\ell^H = \widehat{\mathfrak{F}}GG^T\widehat{\mathfrak{F}}^H \in \mathbb{R}^{n\times n}.$$

Using the low-rank structure $L_\ell = \widehat{W}_\ell\widehat{W}_\ell^H$, $\widehat{W}_\ell \in \mathbb{C}^{n\times(p+r)}$, of the Lyapunov residual together with the commonly known result that the eigenvalues $\sigma(\widehat{W}_\ell\widehat{W}_\ell^H) \setminus \{0\} = \sigma(\widehat{W}_\ell^H\widehat{W}_\ell) \setminus \{0\}$, see, e.g., [11, Theorem 1.32], leads to an efficient way to compute and accumulate the Lyapunov residual and its spectral or Frobenius norm to control the accuracy of the ADI iteration [4].

The previous versions of the low-rank ADI method compute complex low-rank factors $\widehat{V}_\ell, \widehat{W}_\ell \in \mathbb{C}^{n\times(p+r)}$, $\widehat{Z}_\ell \in \mathbb{C}^{n\times(\ell(p+r))}$. To avoid complex arithmetic and storage of complex matrices as much as possible, Benner et al. [3, 5] introduced a reformulated low-rank ADI iteration, where they exploit the fact that the ADI shifts need to occur either as a real number $q_\ell \in \mathbb{R}^-$ or as a pair of complex conjugate numbers $q_\ell \in \mathbb{C}^-, q_{\ell+1} = \overline{q_\ell}$. The resulting low-rank ADI iteration works with real low-rank factors $V_\ell, W_\ell \in \mathbb{R}^{n\times(p+r)}$, $Z_\ell \in \mathbb{R}^{n\times(\ell(p+r))}$. The real low-rank ADI method is shown in Algorithm 2. The approximate solution of the Lyapunov equation (4.1) is

$$X \approx Z_\ell Z_\ell^T \in \mathbb{R}^{n\times n}.$$

The corresponding Lyapunov residual has a real low-rank representation

$$L_\ell = W_\ell W_\ell^T = \mathfrak{F}GG^T\mathfrak{F}^T \in \mathbb{R}^{n\times n}, \tag{4.4}$$

where $\mathfrak{F}(F, q_1, \ldots, q_\ell) \in \mathbb{R}^{n\times n}$ is an analytic matrix function depending on F and the ADI shifts $q_1, \ldots, q_\ell$. Notice that $\mathfrak{F} \equiv \widehat{\mathfrak{F}}$ iff $\{q_i\}_{i=1}^{\ell} = \overline{\{q_i\}_{i=1}^{\ell}}$, i.e., the ADI shifts are closed under complex conjugation. See, e.g., [7, 5] for details.

5 Low-Rank Residual Newton-ADI Method

Using Algorithm 2 as the inner loop to solve the Lyapunov equations in Line 5 of Algorithm 1, we arrive at an algorithm for the Kleinman-Newton method, where the low-rank structure can be used to efficiently compute residuals and the quartic function (3.11) that arises in the line search computation.

As we have seen in (4.4) in the previous section (we now keep track of the Kleinman-Newton iteration counter k), the Lyapunov residual is

$$L_\ell^{(k+1)} = W_\ell^{(k+1)}(W_\ell^{(k+1)})^T,$$

where ℓ is the iteration counter in the inner ADI iteration and $W_\ell^{(k+1)} \in \mathbb{R}^{n\times(p+r)}$.

Since $\|L_\ell^{(k+1)}\|_F^2$ is the sum of the squares of the eigenvalues of $L_\ell^{(k+1)}$ and

$$\sigma(W_\ell^{(k+1)}(W_\ell^{(k+1)})^T) \setminus \{0\} = \sigma((W_\ell^{(k+1)})^T W_\ell^{(k+1)}) \setminus \{0\},$$

the norm $\|L_\ell^{(k+1)}\|_F^2$ can be efficiently computed by solving a small $(p+r)\times(p+r)$ eigenvalue problem.

5.1 Norm of the Difference of Outer Products

Let $W \in \mathbb{R}^{n\times m}$ and $K \in \mathbb{R}^{n\times p}$ with $m+p \ll n$ be generic matrices. We frequently need to compute Frobenius or 2-norms of the difference $WW^T - KK^T$. This can be done efficiently using the indefinite low-rank factorization $WW^T - KK^T = UDU^T \in \mathbb{R}^{n\times n}$, where

$$U = \begin{bmatrix} W & K \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} I_m & 0 \\ 0 & -I_p \end{bmatrix}.$$

For the spectrum we have $\sigma(UDU^T) \setminus \{0\} = \sigma(U^T UD) \setminus \{0\}$ (see, e.g., [11, Theorem 1.32]). Since $U^T UD$ is a small $(m+p)\times(m+p)$ matrix, its spectrum can be computed efficiently and we can use

$$\|WW^T - KK^T\|_2 = \max\{|\lambda| : \lambda \in \sigma(WW^T - KK^T)\} = \max\{|\lambda| : \lambda \in \sigma(U^T UD)\},$$
$$\|WW^T - KK^T\|_F^2 = \sum_{\lambda_i \in \sigma(WW^T - KK^T)} \lambda_i^2 = \sum_{\lambda_i \in \sigma(U^T UD)} \lambda_i^2.$$

Notice that since $U^T UD$ is not symmetric, $\max\{|\lambda| : \lambda \in \sigma(U^T UD)\} \neq \|U^T UD\|_2$ and $\sum_{\lambda_i \in \sigma(U^T UD)} \lambda_i^2 \neq \|U^T UD\|_F^2$.

5.2 Low-Rank Riccati Residual and Feedback Accumulation

Recall that $\widetilde{X}^{(k+1)} = X^{(k)} + S^{(k)}$. Consider

$$S^{(k)}B = \widetilde{X}^{(k+1)}B - X^{(k)}B =: \widetilde{K}^{(k+1)} - K^{(k)} =: \Delta\widetilde{K}^{(k+1)} \in \mathbb{R}^{n\times r}, \tag{5.1}$$

which defines the change of the feedback K corresponding to the trial solution $\widetilde{X}^{(k+1)}$ of (3.8a).

The key ingredient to use the line search idea efficiently for large-scale problems are the low-rank formulations of the Lyapunov and Riccati residuals. Recall from (4.4) that

$$L^{(k+1)} = W^{(k+1)}(W^{(k+1)})^T \tag{5.2a}$$

and assume that

$$\mathcal{R}(X^{(k)}) = W^{(k)}(W^{(k)})^T - \Delta K^{(k)}(\Delta K^{(k)})^T = U^{(k)} D (U^{(k)})^T \tag{5.2b}$$

with

$$D = \begin{bmatrix} I_{r+p} & 0 \\ 0 & -I_r \end{bmatrix} \text{ and } U^{(k)} = \left[W^{(k)} \,|\, \Delta K^{(k)} \right] \in \mathbb{R}^{n \times (2r+p)}. \tag{5.2c}$$

For $k = 0$ and $X^{(0)} = 0$, (5.2) holds with $W^{(0)} = C^T$ and $\Delta K^{(0)} = 0$. We call a factorization of the form (5.2b) an indefinite low-rank factorization (compare Section 5.1).

If one uses (5.2) and the feedback change (5.1), than (3.9) implies

$$\begin{aligned}
&\mathcal{R}(X^{(k+1)}) = \mathcal{R}(X^{(k)} + \lambda_k S^{(k)}) \\
&= (1-\lambda_k) U^{(k)} D (U^{(k)})^T + \lambda_k \, W^{(k+1)}(W^{(k+1)})^T - \lambda_k^2 \Delta \widetilde{K}^{(k+1)} \left(\Delta \widetilde{K}^{(k+1)} \right)^T \\
&= (1-\lambda_k) \left(W^{(k)}(W^{(k)})^T - \Delta K^{(k)}(\Delta K^{(k)})^T \right) + \lambda_k \, W^{(k+1)}(W^{(k+1)})^T \\
&\quad - \lambda_k^2 \Delta \widetilde{K}^{(k+1)} \left(\Delta \widetilde{K}^{(k+1)} \right)^T \\
&= \left[\left[\sqrt{(1-\lambda_k)} W^{(k)} \,|\, \sqrt{\lambda} W^{(k+1)} \right] \left[\sqrt{(1-\lambda_k)} \Delta K^{(k)} \,|\, \lambda_k \Delta \widetilde{K}^{(k+1)} \right] \right] \\
&\quad \times \begin{bmatrix} I_{(s+1)(p+r)} & 0 \\ 0 & -I_{(s+1)r} \end{bmatrix} \\
&\quad \times \left[\left[\sqrt{(1-\lambda_k)} W^{(k)} \,|\, \sqrt{\lambda_k} W^{(k+1)} \right] \left[\sqrt{(1-\lambda_k)} \Delta K^{(k)} \,|\, \lambda_k \Delta \widetilde{K}^{(k+1)} \right] \right]^T,
\end{aligned} \tag{5.3}$$

where $s \in \{0, 1, \ldots, \}$ is the number of iterations immediately before the k-th iteration in which the step size was less than one. See below for more details.

If $\lambda_k = 1$, then $X^{(k+1)} = \widetilde{X}^{(k+1)}$, $\Delta K^{(k+1)} = \Delta \widetilde{K}^{(k+1)}$ and (5.3) simplifies to

$$\begin{aligned}
\mathcal{R}(X^{(k+1)}) &= \mathcal{R}(\widetilde{X}^{(k+1)}) \\
&= W^{(k+1)}(W^{(k+1)})^T - \Delta K^{(k+1)}(\Delta K^{(k+1)})^T =: U^{(k+1)} D (U^{(k+1)})^T
\end{aligned} \tag{5.4}$$

with $U^{(k+1)} = \left[W^{(k+1)} \,|\, \Delta K^{(k+1)} \right]$, which is of the form (5.2b). If $\lambda_k \in (0, 1)$, we can redefine

$$\begin{aligned}
W^{(k+1)} &\leftarrow \left[\sqrt{(1-\lambda_k)} W^{(k)} \,|\, \sqrt{\lambda_k} W^{(k+1)} \right] \in \mathbb{R}^{n \times (s+1)(p+r)}, \\
\Delta K^{(k+1)} &\leftarrow \left[\sqrt{(1-\lambda_k)} \Delta K^{(k)} \,|\, \lambda_k \Delta \widetilde{K}^{(k+1)} \right] \in \mathbb{R}^{n \times (s+1)r}, \\
D &\leftarrow \begin{bmatrix} I_{(s+1)(p+r)} & 0 \\ 0 & -I_{(s+1)r} \end{bmatrix}.
\end{aligned}$$

After this redefinition, (5.4) holds. Notice that if $\lambda_k < 1$, the sizes of $W^{(k+1)}$ and $\Delta K^{(k+1)}$ grow. As mentioned before, their sizes depend on the number $s \in \{0, 1, \ldots, \}$

of iterations immediately before the k-th iteration in which the step size was less than one, i.e., on $s \in \{0, 1, \dots, \}$ with $\lambda_{k-s-1} = 1, \lambda_{k-s} < 1, \lambda_k < 1$.

The representation (5.4) can be used to compute the Riccati residual $\|\mathcal{R}(X^{(k)} + \lambda_k S^{(k)})\|_F$ in dependence of λ_k efficiently (see Section 5.1). It is important to mention that we need to keep $U^{(k)} \in \mathbb{R}^{n\times((s+1)(2r+p))}$ to perform the line search; it is not sufficient to just keep $\|\mathcal{R}(X^{(k)})\|_F$.

The trial iterate $\widetilde{X}^{(k+1)}$ is computed by Algorithm 2 iteratively and, consequently, the trial feedback $\widetilde{K}^{(k+1)} = \widetilde{X}^{(k+1)}B \in \mathbb{R}^{n\times r}$ can already be computed during the execution of Algorithm 2. Let ℓ be the iteration counter in Algorithm 2. We have

$$\begin{aligned}\widetilde{K}_\ell^{(k+1)} &= \widetilde{X}_\ell^{(k+1)}B = \begin{bmatrix}\widetilde{V}_1 & \dots & \widetilde{V}_\ell\end{bmatrix}\left(\begin{bmatrix}\widetilde{V}_1^T \\ \vdots \\ \widetilde{V}_\ell^T\end{bmatrix} B\right) = \sum_{j=1}^{\ell} \widetilde{V}_j(\widetilde{V}_j^T B) \\ &= \widetilde{K}_{\ell-1}^{(k+1)} + \widetilde{V}_\ell(\widetilde{V}_\ell^T B), \qquad \widetilde{K}_0^{(k+1)} = 0.\end{aligned}$$

If we define $\Delta\widetilde{K}_0^{(k+1)} = -K^{(k)}$, then

$$\Delta\widetilde{K}_\ell^{(k+1)} = \widetilde{K}_\ell^{(k+1)} - K^{(k)} = \widetilde{K}_{\ell-1}^{(k+1)} + \widetilde{V}_\ell(\widetilde{V}_\ell^T B) - K^{(k)} = \Delta\widetilde{K}_{\ell-1}^{(k+1)} + \widetilde{V}_\ell(\widetilde{V}_\ell^T B).$$

Thus, the feedback change can be assembled efficiently during the ADI iteration. The low-rank Riccati residual factor for the k+1-st Riccati step after ℓ ADI steps can be written as $U_\ell^{(k+1)} = [W_\ell \mid \Delta\widetilde{K}_\ell^{(k+1)}] \in \mathbb{R}^{n\times(2r+p)}$. The Riccati residual norm $\|\mathcal{R}(X_\ell^{(k+1)})\|_F$ can be computed easily during the ADI iteration by computing the eigenvalues of the small matrix $(U_\ell^{(k+1)})^T U_\ell^{(k+1)} D$, see Section 5.1.

5.3 Low-Rank Line Search Implementation

To compute the step size as discussed in Section 3.2 for large-scale problems, we need to compute the quartic polynomial (3.11). We can compute the coefficients defined in (3.12) efficiently.

The coefficient $\alpha^{(k)} = \|\mathcal{R}(X^{(k)}\|_F^2$ can be computed using (5.2b) (see Section 5.1). Similarly, $\beta^{(k)} = \|L^{(k+1)}\|_F^2 = \|W^{(k+1)}(W^{(k+1)})^T\|_F^2$ can be computed efficiently as shown at the beginning of this section. Instead of using eigenvalues of $M, N \in \mathbb{R}^{n\times n}$, we can use the property $\operatorname{tr}(MN) = \operatorname{tr}(NM)$ and, for symmetric matrices M, $\operatorname{tr}\left(M^2\right) = \sum_{i,j}(M_{ij})^2$, and compute

$$\begin{aligned}\beta^{(k)} &= \|W^{(k+1)}(W^{(k+1)})^T\|_F^2 = \operatorname{tr}\left(W^{(k+1)}(W^{(k+1)})^T W^{(k+1)}(W^{(k+1)})^T\right) \\ &= \operatorname{tr}\left((W^{(k+1)})^T W^{(k+1)}(W^{(k+1)})^T W^{(k+1)}\right) = \|(W^{(k+1)})^T W^{(k+1)}\|_F^2.\end{aligned}$$

Similarly, with $\Delta\widetilde{K}^{(k+1)} = S^{(k)}B \in \mathbb{R}^{n\times r}$,

$$\delta^{(k)} = \|\Delta\widetilde{K}^{(k+1)}(\Delta\widetilde{K}^{(k+1)})^T\|_F^2 = \|(\Delta\widetilde{K}^{(k+1)})^T\Delta\widetilde{K}^{(k+1)}\|_F^2.$$

Application of trace identities gives

$$
\begin{aligned}
\gamma^{(k)} &= \langle \mathcal{R}(X^{(k)}), L^{(k+1)} \rangle = \operatorname{tr}\left(U^{(k)} D (U^{(k)})^T W^{(k+1)} (W^{(k+1)})^T\right) \\
&= \operatorname{tr}\left((U^{(k)})^T W^{(k+1)} (W^{(k+1)})^T U^{(k)} D\right) \\
&= \operatorname{tr}\left(\begin{bmatrix} (W^{(k)})^T W^{(k+1)} \\ (\Delta K^{(k)})^T W^{(k+1)} \end{bmatrix} \left[(W^{(k+1)})^T W^{(k)} \mid -(W^{(k+1)})^T \Delta K^{(k)}\right]\right) \\
&= \sum_{i,j} \begin{bmatrix} (W^{(k)})^T W^{(k+1)} \\ (\Delta K^{(k)})^T W^{(k+1)} \end{bmatrix}_{ij} \begin{bmatrix} (W^{(k)})^T (W^{(k+1)}) \\ -(\Delta K^{(k)})^T W^{(k+1)} \end{bmatrix}_{ij} \\
&= \sum_{i,j} \left(((W^{(k)})^T W^{(k+1)})_{ij}\right)^2 - \sum_{i,j} \left((\Delta K^{(k)})^T W^{(k+1)})_{ij}\right)^2
\end{aligned}
$$

and, analogously,

$$
\begin{aligned}
\varepsilon^{(k)} &= \langle \mathcal{R}(X^{(k)}), S^{(k)} B B^T S^{(k)} \rangle = \operatorname{tr}\left(U^{(k)} D (U^{(k)})^T \Delta \widetilde{K}^{(k+1)} (\Delta \widetilde{K}^{(k+1)})^T\right) \\
&= \sum_{i,j} \left(((W^{(k)})^T \Delta \widetilde{K}^{(k+1)})_{ij}\right)^2 - \sum_{i,j} \left(((\Delta K^{(k)})^T \Delta \widetilde{K}^{(k+1)})_{ij}\right)^2.
\end{aligned}
$$

Finally,

$$
\begin{aligned}
\zeta^{(k)} &= \langle L^{(k+1)}, S^{(k)} B B^T S^{(k)} \rangle = \operatorname{tr}\left(W^{(k+1)} (W^{(k+1)})^T \Delta \widetilde{K}^{(k+1)} (\Delta \widetilde{K}^{(k+1)})^T\right) \\
&= \sum_{i,j} \left(((W^{(k+1)})^T \Delta \widetilde{K}^{(k+1)})_{ij}\right)^2.
\end{aligned}
$$

After choosing λ_k appropriately, the next iterate $X^{(k+1)}$ (3.8b) and the feedback $K^{(k+1)}$ can be computed. Using a low-rank ADI method (see Section 4), $\widetilde{X}^{(k+1)} = \widetilde{Z}^{(k+1)} \left(\widetilde{Z}^{(k+1)}\right)^T$ as low-rank approximation of the solution of (3.8a) and the low-rank approximations of the previous iterate $X^{(k)} = Z^{(k)} (Z^{(k)})^T$ are used.

$$
\begin{aligned}
X^{(k+1)} &= X^{(k)} + \lambda_k S^{(k)} = (1-\lambda_k) X^{(k)} + \lambda_k \widetilde{X}^{(k+1)} \\
&= (1-\lambda_k) Z^{(k)} \left(Z^{(k)}\right)^T + \lambda_k \widetilde{Z}^{(k+1)} \left(\widetilde{Z}^{(k+1)}\right)^T \\
&= \left[\sqrt{1-\lambda_k}\, Z^{(k)} \mid \sqrt{\lambda_k}\, \widetilde{Z}^{(k+1)}\right] \left[\sqrt{1-\lambda_k}\, Z^{(k)} \mid \sqrt{\lambda_k}\, \widetilde{Z}^{(k+1)}\right]^T. \qquad (5.5) \\
K^{(k+1)} &= X^{(k+1)} B = (1-\lambda_k) X^{(k)} B + \lambda_k \widetilde{X}^{(k+1)} B \\
&= (1-\lambda_k) K^{(k)} + \lambda_k \widetilde{K}^{(k+1)}. \qquad (5.6)
\end{aligned}
$$

Notice that the size of $Z^{(k)}$ and $\widetilde{Z}^{(k+1)}$ depends on the number of ADI steps that are needed to solve (3.8a). Although (5.5) might be very large, it is important to mention that it only needs to be computed at the end of the Newton iteration, because the previous iterate $X^{(k)}$ enters the right-hand side of (3.8a) only as product with the input matrix B from the right. This means one only needs the inexpensively accumulated feedback $K^{(k+1)} = X^{(k+1)}B$ in Eq. (5.6) to proceed with the Newton iteration. Furthermore, typically, a line search will only be necessary in the first few Newton steps, so that (5.5) might never be used after the first few iterations and instead simply $X^{(k+1)} = \widetilde{X}^{(k+1)} = \widetilde{Z}^{(k+1)}(\widetilde{Z}^{(k+1)})^T$ is used.

5.4 Complete Implementation

We conclude this section with a summary of the resulting algorithm and some comments on the line search and the convergence of the inexact Kleinman-Newton method with line search.

We perform a line search if after reaching the condition (3.7) at ADI step ℓ it holds that $\|\mathcal{R}(\widetilde{X}_\ell^{(k+1)})\| > (1-\alpha)\|\mathcal{R}(X^{(k)})\|$. We also perform a line search in the following cases:

a) Before reaching the condition (3.7), the actual step $\ell \geq 2$ yields

$$\|L_\ell\|_F > \|L_1\|_F,$$

i.e., the norm of the Lyapunov residual exceeds the norm of the initial Lyapunov residual.

b) The number of ADI steps ℓ exceeds the maximal number of allowed ADI steps without reaching the condition (3.7).

If the conditions in a) or b) are observed, it indicates that the ADI method does not converge, e.g., because the matrix $A^{(k)}$ is not stable. Although condition (3.7) is violated, we perform a line search, since the cost of its execution is small, and accept $X^{(k+1)} = X^{(k)} + \lambda_k S^{(k)}$ if the sufficient decrease condition (3.5) is fulfilled.

If the line search method determines a λ_k that is too small, we switch to an 'exact' Kleinman-Newton method, i.e., we use Algorithm 1 with ADI Algorithm 2 with tolerance $\mathrm{tol}_{\mathrm{ADI}} = 10^{-1}\mathrm{tol}_{\mathrm{Newt}}$ as the inner solver. Since we cannot guarantee stability of $A^{(k)}$, it is not guaranteed that Algorithm 2 converges. If we observe that Algorithm 2 does not converge, we restart the entire process using the 'exact' Kleinman-Newton method as described above. During the 'exact' Kleinman-Newton scheme, the algorithm switches back to the inexact scheme as soon as the Riccati residual shows the expected convergence behavior.

We note that in the numerical example studies in the next section, the ADI Algorithm always reached the required tolerance, i.e., condition (3.7) was always achieved, and the line search was always successful.

The inexact Kleinman-Newton method with line search and a real low-rank ADI method as inner solver is summarized in Algorithm 3. The residual $\mathcal{R}(\widetilde{X}^{(k+1)}) = U_\ell^{(k+1)} D (U_\ell^{(k+1)})^T$ is accumulated during the ADI iteration. In practice, the factor U of the indefinite low-rank decomposition of the Riccati residual in lines 20, 28, and 32 is never assembled explicitly since norm computation and line search directly use W and ΔK.

Algorithm 3 inexact Kleinman-Newton-ADI method with line search

```
Input: $A, B, C$, initial feedback $K^{(0)}$, $\mathrm{tol}_{\mathrm{Newt}}$, $\bar{\eta} \in (0,1)$, and $\alpha \in (0, 1-\bar{\eta})$.
Output: $K^{(k+1)}$ (optional: $Z^{(k+1)}$ such that $Z^{(k+1)}(Z^{(k+1)})^T$ is a stabilizing approximate solution of the CARE (1.1)).
Set $k = 0$, $\mathrm{res}_{\mathrm{Newt}}^{(0)} = \|C^TC + K^{(0)}\left(K^{(0)}\right)^T\|$.
while $\left(\mathrm{res}_{\mathrm{Newt}}^{(k)} > \mathrm{tol}_{\mathrm{Newt}} \cdot \mathrm{res}_{\mathrm{Newt}}^{(0)}\right)$ do
    Set $A^{(k)} = \left(A^T - K^{(k)}B^T\right)$, $G = \left[C^T \,|\, K^{(k)}\right]$.
    Compute ADI shifts $\{q_\ell\}_{\ell=1}^{n_{max,ADI}} \subset \mathbb{C}^-$ and choose $\eta_k \in (0, \bar{\eta}]$.
    Set $\ell = 1$, $W_0 = G$, $\Delta K_0 = -K^{(k)}$ (optional $Z = [\,]$).
    while $\left(\|W_\ell W_\ell^T\|_F > \eta_k \mathrm{res}_{\mathrm{Newt}}^{(k)}\right)$ do
        $V = \left(A^{(k)} + q_\ell I\right)^{-1} W_{\ell-1}$
        if $\mathrm{Im}\,(q_\ell) = 0$ then
            $W_\ell = W_{\ell-1} - 2q_\ell V$
            $\widetilde{V} = \sqrt{-2q_\ell}\, V$
            $\Delta K_\ell = \Delta K_{\ell-1} + \widetilde{V}\left(\widetilde{V}^T B\right)$
        else
            $\gamma = 2\sqrt{-\mathrm{Re}\,(q_\ell)}$, $\delta = \mathrm{Re}\,(q_\ell) \,/\, \mathrm{Im}\,(q_\ell)$
            $W_{\ell+1} = W_{\ell-1} + \gamma^2(\mathrm{Re}\,(V) + \delta\,\mathrm{Im}\,(V))$
            $\widetilde{V} = \left[\gamma\,(\mathrm{Re}\,(V) + \delta\,\mathrm{Im}\,(V)) \,|\, \gamma\sqrt{(\delta^2+1)}\,\mathrm{Im}\,(V)\right]$
            $\ell = \ell + 1$
            $\Delta K_\ell = \Delta K_{\ell-2} + \widetilde{V}\left(\widetilde{V}^T B\right)$
        end if
        (optional $Z = \left[Z \,|\, \widetilde{V}\right]$)
        $U_\ell = [W_\ell \,|\, \Delta K_\ell]$
        $\ell = \ell + 1$
    end while
    if $\|U_\ell D U_\ell^T\|_F > (1-\alpha)\mathrm{res}_{\mathrm{Newt}}^{(k)}$ then
        Choose $\lambda_k \in (0,1)$ using Armijo or exact line search.
        $\Delta K_{\ell-1} = \lambda_k \Delta K_{\ell-1}$
        $W^{(k+1)} = \left[\sqrt{1-\lambda_k}W^{(k)} \,|\, \sqrt{\lambda_k}W^{(k+1)}\right]$
        $\Delta K^{(k+1)} = \left[\sqrt{1-\lambda_k}\Delta K^{(k)} \,|\, \Delta K_{\ell-1}\right]$
        $U^{(k+1)} = \left[W^{(k+1)} \,|\, \Delta K^{(k+1)}\right]$
        (optional $Z^{(k+1)} = \left[\sqrt{1-\lambda_k}Z^{(k)} \,|\, \sqrt{\lambda_k}Z\right]$)
    else
        $W^{(k+1)} = W_\ell$, $\Delta K^{(k+1)} = \Delta K_{\ell-1}$
        $U^{(k+1)} = U_\ell$
        (optional $Z^{(k+1)} = Z$)
    end if
    $K^{(k+1)} = K^{(k)} + \Delta K_{\ell-1}$
    $\mathrm{res}_{\mathrm{Newt}}^{(k+1)} = \|U^{(k+1)}D\left(U^{(k+1)}\right)^T\|_F$
    $k = k + 1$
end while
```

6 Numerical experiments

Consider the infinite dimensional optimal control problem

$$
\begin{aligned}
&\text{minimize } \frac{1}{2}\int_0^\infty \left(\gamma \int_{\Omega_O} \tilde{x}(\xi,t)\mathrm{d}\xi\right)^2 + u^2(t)\,\mathrm{d}t,\\
&\text{subject to } \frac{\partial \tilde{x}}{\partial t}(\xi,t) = \Delta\tilde{x}(\xi,t) + 20\frac{\partial \tilde{x}}{\partial \xi_2}(\xi,t) + 100\tilde{x}(\xi,t) + f(\xi)u(t), \qquad \xi\in\Omega,\ t>0,\\
&\qquad\qquad \tilde{x}(\xi,t) = 0, \qquad\qquad \xi\in\partial\Omega,\ t>0,
\end{aligned}
$$

with $\Omega = (0,1)^d$, $d \in 2,3$, $\Omega_O \subset \Omega$, $\gamma > 0$, and

$$
f(\xi) := \begin{cases} 100 & \xi \in \Omega_C, \\ 0 & \text{else}, \end{cases}
$$

where $\Omega_C = (0.1,0.3)\times(0.4,0.6)$ if $d=2$ and $\Omega_C = (0.1,0.3)\times(0.4,0.6)\times(0.1,0.3)$ if $d=3$. For $d=2$, this example was also used by Feitzinger et al. [9] and by Morris and Navasca [20].[1] We use piecewise linear finite elements to discretize the optimal control problem. More specifically, we use P1 finite elements on a uniform triangulation. If $d=2$, $\Omega=(0,1)^2$ is divided into squares of size $h\times h$ and each square is divided into two triangles. If $d=3$, $\Omega=(0,1)^3$ is divided into cubes of size $h\times h\times h$ and each cube is divided into six tetrahedra. We use mesh sizes h such that the mesh is aligned with the boundaries of Ω_O and of Ω_C. This leads to the linear quadratic control problem

$$
\begin{aligned}
&\text{Minimize } \frac{1}{2}\int_0^\infty y(t)^T y(t) + u^2(t)\,\mathrm{d}t, && && (6.1a)\\
&\text{subject to } E\dot{x}(t) = Ax(t) + Bu(t), && t>0, && (6.1b)\\
&\qquad y(t) = \gamma C x(t), && t>0, && (6.1c)
\end{aligned}
$$

where $E\in\mathbb{R}^{n\times n}$ is the symmetric positive definite mass matrix, $A\in\mathbb{R}^{n\times n}$, $B\in\mathbb{R}^{n\times 1}$, $C\in\mathbb{R}^{1\times n}$, $x(t)\in\mathbb{R}^n$, $u(t)\in\mathbb{R}^r$, and $y(t)\in\mathbb{R}$, where n is proportional to $1/h^2$.

For the output, we consider the cases $\Omega_O=\Omega$ and $\Omega_O=\Omega_C$. If $\Omega_O=\Omega$, then the finite element discretization results in the output matrix $C=e^TE$, where e is the vector of all ones, and if $\Omega_O=\Omega_C$, then the finite element discretization results in the output matrix $C=B^T/100$.

The matrix $E^{-1}A$ is stable and, therefore, for this LQR problem (6.1) the Assumption 2 is satisfied. It is a basic result, see, e.g., [18], that $u_*(t) = -Kx_*(t)$ with

[1] Both papers [9] and [20] use a central finite difference method on a uniform grid with mesh size $h = 1/(n+1)$. Feitzinger et al. [9] use the output matrix $C=[0.1,\dots,0.1]$ and $\gamma=1$, and Morris and Navasca [20] use the output matrix $C=B^T$ and $\gamma=1$. These output matrices correspond to scaled versions of the outputs resulting from $\Omega_O=\Omega$ and $\Omega_O=\Omega_C$, respectively, in the PDE model. In fact, if $\Omega_O=\Omega=(0,1)^2$, the finite difference spatial discretization of the objective function is $\gamma\int_\Omega \tilde{x}(\xi,t)\mathrm{d}\xi \approx \gamma h^2\sum_{i,j=1}^n x_{ij}(t) = \gamma Cx(t)$ where $x_{ij}(t)\approx\tilde{x}((ih,jh),t)$, $i,j=1,\dots,n$, and $C=[h^2,\dots,h^2]\in\mathbb{R}^{1\times n^2}$. This output matrix C scaled by $1/(10h^2)$ corresponds to the output matrix in [9]. If $\Omega_O=\Omega_C\subset(0,1)^2$, then the finite difference spatial discretization of the objective function leads to the output matrix $(h^2/100)B^T$, which is the output matrix in [20] scaled by $100/h^2$.

$K = B^T XE$ minimizes the cost functional (6.1a) with X as stabilizing solution of the generalized CARE

$$\gamma^2 C^T C + A^T XE + E^T XA - E^T XBB^T XE = 0. \tag{6.2}$$

The extension of our results for the solution of CARE (1.1) to the generalized CARE (6.2) with nonsingular E is straightforward.

A $\gamma \gg 1$ increases the effect that $\|\mathcal{R}(X^{(1)})\|_F \gg \|\mathcal{R}(X^{(0)})\|_F$. The ADI shifts are computed following the V-shifts idea in [6]. In all computations the mesh size is $h = 1/30$. This leads to matrix sizes $n = 841$ in the 2D case and $n = 24,389$ in the 3D case.

We apply the Kleinman-Newton-ADI method either 'exactly' or inexactly. In the latter case we either use the forcing parameter η_k in (3.4) given by $\eta_k = 1/(k^3+1)$ or by $\eta_k = \min\{0.1, 0.9\|\mathcal{R}(X^{(k)})\|_F\}$. The first choice leads to superlinear convergence, while the second results in quadratic convergence (under the assumptions of Theorem 10). In all cases the Kleinman-Newton-ADI method is stopped when the normalized residual $\|\mathcal{R}(X^{(k)})\|/\|C^T C\|$ drops below $\text{tol}_{\text{Newt}} = 10^{-12}$. In the 'exact' Kleinman-Newton-ADI method, the ADI tolerance is set to $\text{tol}_{\text{ADI}} = \text{tol}_{\text{Newt}}/10$. We apply all methods without line search ('no LS'), i.e., set $\lambda_k = 1$ in all iterations, and with line search. If the sufficient decrease condition (3.5) is not satisfied for $\lambda_k = 1$, then we compute a step size using a simple implementation of the Armijo rule with $\beta = 0.5$, cf. Section 3.2.1.

The performances of the various Kleinman-Newton-ADI methods are summarized in Tables 1 to 7. In all tables, # Newt. is the total number of (inexact) Newton steps executed before the stopping criterion $\|\mathcal{R}(X^{(k)})\|/\|C^T C\| < \text{tol}_{\text{Newt}} = 10^{-12}$ is satisfied, # ADI is the total number of ADI iterations executed, and # LS is the total number of times the step size λ_k was chosen to be less than one. The entry 'no LS' indicates that the algorithm was run without line search, i.e., that $\lambda_k = 1$, $\forall k$. In all variations of the Kleinman-Newton-ADI method, the execution times are essentially proportional to the total number of ADI steps performed. Due to the low-rank structure, the execution times for other algorithm components, such as line search, are negligible compared to the execution of one ADI iteration.

In all examples shown, the exact and inexact versions of the Kleinman-Newton methods converged, and the inexact versions of the Kleinman-Newton method significantly outperform the exact version. We note that although in all examples the inexact Kleinman-Newton method without line search converged, there is no convergence proof to guarantee this (unless the conditions on the Lyapunov residual in Feitzinger et al. [9] can be satisfied, which is not the case when low-rank ADI methods are used).

The line search performed differently for the outputs $\Omega_O = \Omega_C$ ($C = B^T/100$) (see Tables 1, 3) and $\Omega_O = \Omega$ ($C = e^T E$) (see Tables 5, 7). In the example $\Omega_O = \Omega_C$ ($C = B^T/100$), the line search is active, i.e., $\lambda_k \neq 1$, in at most the first two iterations and it is only active if $\gamma \gg 1$. In this example, using the line search always resulted in fewer Newton iterations and led to fewer ADI iterations overall.

In the example $\Omega_O = \Omega$ ($C = e^T E$), the line search is active, i.e., $\lambda_k \neq 1$, in more iterations. The line search is active in the first iterations and if $\lambda_k = 1$ in one iteration

k, it is equal to one on all subsequent iterations. In the 2D case with $\gamma = 1$ (Table 6a), the line search leads to significantly more Newton and ADI iterations. In this case, $\left\|\mathcal{R}(X^{(k)} + S^{(k)})\right\|_F \gg \left\|\mathcal{R}(X^{(k)})\right\|_F$ for the first iterations, and a small step size λ_k is needed to satisfy the sufficient decrease condition (3.5). This leads to small steps initially and a substantial increase in Newton iterations. It may be possible to improve the performance of the inexact Kleinman-Newton method with line search by refining the forcing parameter η_k, i.e., the choice of Lyapunov residual tolerances. This is part of future research.

In all other cases, using the line search leads to fewer Newton iterations and mostly fewer ADI iterations. Notice that the line search enforces the monotonicity

$$\left\|\mathcal{R}(X^{(k+1)})\right\|_F < \left\|\mathcal{R}(X^{(k)})\right\|_F$$

which can result in a significantly smaller right hand side in (3.4), i.e., a smaller Lyapunov equation solver tolerance, compared to when no line search is used. Therefore, using a line search can require more ADI iterations per Newton iteration; compare Table 6b ('superlinear'), Table 8b ('inexact'), and Table 8c ('quadratic').

7 Conclusions

We have presented an efficient implementation of the inexact Kleinman-Newton method with a low-rank ADI subproblem solver. On the theoretical side, we presented a convergence proof which is based on convergence proofs for general inexact Newton methods. Because of the low-rank case and lack of positive semi-definiteness conditions, like the one in Theorem 7 [9, Thm. 4.3], it is not possible to ensure that all iterates are stabilizing if the initial iterate is stabilizing. This was not an issue in our numerical example. In our convergence proof, the line search is needed to ensure that the Riccati residuals decrease monotonically in norm. Although in our numerical examples, the inexact Kleinman-Newton method with a low-rank ADI subproblem solver always converged when used without line search, there is no guarantee for this and we have observed other examples where the inexact Kleinman-Newton method without line search failed. The numerical example showed that the line search can lead to substantial reduction in the overall number of ADI iterations and, therefore, overall computational cost, but there is one case where the line search results in substantially more Kleinman-Newton iterations and in a substantially higher number of total ADI iterations. Possible improvements by changing the forcing parameter, i.e., the choice of Lyapunov residual tolerances, is part of future research. We have begun numerical experiments with the computation of feedback controls for incompressible Navier-Stokes flows, similar to [1], where stability of iterates can be an issue. A detailed report of these tests, and comparisons with other large-scale Riccati solvers, like [8, 21, 19, 17], is part of future research.

References

[1] E. Bänsch, P. Benner, J. Saak, and H. K. Weichelt, *Riccati-based boundary feedback stabilization of incompressible Navier-Stokes flows*, SIAM J. Sci. Comput., 37 (2015), pp. A832–A858.

[2] 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.

[3] P. Benner, P. Kürschner, and J. Saak, *Efficient handling of complex shift parameters in the low-rank Cholesky factor ADI method*, Numer. Algorithms, 62 (2013), pp. 225–251.

[4] ——, *An improved numerical method for balanced truncation for symmetric second order systems*, Math. Comp. Model. Dyn. Syst., 19 (2013), pp. 593–615.

[5] ——, *A reformulated low-rank ADI iteration with explicit residual factors*, Proc. Appl. Math. Mech., 13 (2013), pp. 585–586.

[6] ——, *Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations*, Electron. Trans. Numer. Anal., 43 (2014), pp. 142–162.

[7] P. Benner, J.-R. Li, and T. Penzl, *Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems*, Numer. Lin. Alg. Appl., 15 (2008), pp. 755–777.

[8] P. Benner and J. Saak, *A Galerkin-Newton-ADI Method for Solving Large-Scale Algebraic Riccati Equations*, Preprint SPP1253-090, DFG-SPP1253, 2010. Available from http://www.am.uni-erlangen.de/home/spp1253/wiki/images/2/28/Preprint-SPP1253-090.pdf.

[9] 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.

[10] C.-H. Guo and A. J. Laub, *On a Newton-like method for solving algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 694–698.

[11] N. J. Higham, *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[12] L. Hogben, ed., *Handbook of Linear Algebra*, Chapman & Hall/CRC, Boca Raton, London, New York, 2nd edition ed., 2014.

[13] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Society for Industrial and Applied Mathematics, Philadelphia, 1995.

[14] D. L. Kleinman, *On an iterative technique for Riccati equation computations*, IEEE Trans. Automat. Control, 13 (1968), pp. 114–115.

[15] P. Lancaster and L. Rodman, *The algebraic Riccati equation*, Oxford University Press, Oxford, 1995.

[16] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.

[17] Y. Lin and V. Simoncini, *A new subspace iteration method for the algebraic Riccati equation*, Numer. Linear Algebra Appl., 22 (2014), pp. 26–47.

[18] A. Locatelli, *Optimal control: An introduction*, Birkhäuser Verlag, Basel, Boston, Berlin, 2001.

[19] A. Massoudi, M. R. Opmeer, and T. Reis, *The ADI method for algebraic Riccati equations*, Hamburger Beiträge zur Angewandten Mathematik 2014-16, Universität Hamburg, 2014.

[20] K. Morris and C. Navasca, *Solution of algebraic Riccati equations arising in control of partial differential equations*, in Control and boundary analysis, J. Cagnol and J.-P. Zolésio, eds., vol. 240 of Lect. Notes Pure Appl. Math., Chapman & Hall/CRC, Boca Raton, FL, 2005, pp. 257–280.

[21] 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.

[22] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra. The Behavior of Nonnormal Matrices and Operators*, Princeton University Press, Princeton, NJ, 2005.

Table 1: Performance of the various Kleinman-Newton-ADI methods for the 2D problem with output $\Omega_O = \Omega_C$ ($C = B^T/100$). Overall the inexact Kleinman-Newton-ADI with forcing parameter $\eta_k = \min\{0.1, 0.9\|\mathcal{R}(X^{(k-1)})\|_F\}$ (quadratic) performs the best, although in some cases other choices of forcing terms lead to slightly fewer ADI iterations. For $\gamma \gg 1$ the line search can lead to significant savings.

(a) Comparison for $\gamma = 1$

Method		# Newt.	# ADI	# LS
exact	$\mathrm{tol}_{\mathrm{ADI}} = 10^{-13}$	3	105	no LS
		3	105	0
inexact	superlinear	5	83	no LS
		5	83	0
	quadratic	4	62	no LS
		4	62	0

(b) Comparison for $\gamma = 10^2$

Method		# Newt.	# ADI	# LS
exact	$\mathrm{tol}_{\mathrm{ADI}} = 10^{-13}$	7	227	no LS
		6	202	1
inexact	superlinear	7	75	no LS
		6	69	1
	quadratic	7	77	no LS
		6	73	1

(c) Comparison for $\gamma = 10^4$

Method		# Newt.	# ADI	# LS
exact	$\mathrm{tol}_{\mathrm{ADI}} = 10^{-13}$	13	376	no LS
		7	186	2
inexact	superlinear	13	88	no LS
		8	80	2
	quadratic	13	56	no LS
		7	52	2

Table 3: Performance of the various Kleinman-Newton-ADI methods for the 3D problem with output $\Omega_O = \Omega_C$ ($C = B^T/100$). The inexact Kleinman-Newton-ADI with forcing parameter $\eta_k = \min\{0.1, 0.9\|\mathcal{R}(X^{(k-1)})\|_F\}$ (quadratic) performs the best. For $\gamma \gg 1$ the line search can lead to significant savings.

(a) Comparison for $\gamma = 1$

Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	2	67	no LS
		2	67	0
inexact	superlinear	5	79	no LS
		5	79	0
	quadratic	4	67	no LS
		4	67	0

(b) Comparison for $\gamma = 10^2$

Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	4	143	no LS
		4	143	0
inexact	superlinear	5	86	no LS
		5	86	0
	quadratic	4	66	no LS
		4	66	0

(c) Comparison for $\gamma = 10^4$

Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	10	279	no LS
		6	177	2
inexact	superlinear	10	85	no LS
		7	81	2
	quadratic	10	72	no LS
		6	58	2

(d) Comparison for $\gamma = 10^6$

Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	16	499	no LS
		6	164	1
inexact	superlinear	16	90	no LS
		7	55	1
	quadratic	16	49	no LS
		6	46	1

Table 5: Performance of the various Kleinman-Newton-ADI methods for the 2D problem with output $\Omega_O = \Omega$ ($C = e^T E$). The inexact Kleinman-Newton-ADI with forcing parameter $\eta_k = \min\{0.1, 0.9\|\mathcal{R}(X^{(k-1)})\|_F\}$ (quadratic) performs the best.

(a) Comparison for $\gamma = 1$

Method		# Newt.	# ADI	# LS
exact	$\mathrm{tol}_{\mathrm{ADI}} = 10^{-13}$	5	190	no LS
		15	577	11
inexact	superlinear	6	82	no LS
		15	181	11
	quadratic	6	80	no LS
		15	130	11

(b) Comparison for $\gamma = 10^2$

Method		# Newt.	# ADI	# LS
exact	$\mathrm{tol}_{\mathrm{ADI}} = 10^{-13}$	13	425	no LS
		9	326	5
inexact	superlinear	13	110	no LS
		10	121	5
	quadratic	14	96	no LS
		10	86	5

(c) Comparison for $\gamma = 10^4$

Method		# Newt.	# ADI	# LS
exact	$\mathrm{tol}_{\mathrm{ADI}} = 10^{-13}$	20	636	no LS
		8	224	4
inexact	superlinear	20	128	no LS
		9	105	4
	quadratic	21	83	no LS
		8	82	4

Table 7: Performance of the various Kleinman-Newton-ADI methods for the 3D problem with output $\Omega_O = \Omega$ ($C = e^T E$). The inexact Kleinman-Newton-ADI with forcing parameter $\eta_k = \min\{0.1, 0.9\|\mathcal{R}(X^{(k-1)})\|_F\}$ (quadratic) performs the best. Line search always reduced the number of Newton iterations.

(a) Comparison for $\gamma = 1$

Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	3	117	no LS
		3	117	0
inexact	superlinear	6	116	no LS
		6	116	0
	quadratic	4	73	no LS
		4	73	0

(b) Comparison for $\gamma = 10^2$

Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	8	289	no LS
		7	254	3
inexact	superlinear	9	93	no LS
		8	107	3
	quadratic	9	72	no LS
		7	78	3

(c) Comparison for $\gamma = 10^4$

Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	16	476	no LS
		9	265	5
inexact	superlinear	16	123	no LS
		10	113	5
	quadratic	16	63	no LS
		9	75	5

(d) Comparison for $\gamma = 10^6$

Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	22	707	no LS
		9	258	4
inexact	superlinear	22	119	no LS
		9	82	4
	quadratic	23	77	no LS
		9	72	4

Max Planck Institute Magdeburg Preprints