On the Implementation of an Inexact Low Rank Kleinman-Newton Iteration for large and Sparse Riccati Equations

Jens Saak and Timo Hylla

1 Max Planck Institute for Dynamics of Complex Technical Systems
Computational Methods in Systems and Control Theory

2 University of Trier
Faculty IV - Mathematics
Numerical Mathematics Group
Let $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$.
Consider the linear quadratic regulator problem

$$\min_{u \in U_{ad}} J(u) = \int_{0}^{\infty} x^T C^T C x + u^T u \, dt$$

constrained to the linear time invariant system

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t).$$

Then the optimal state feedback control is given by

$$u(t) = -B^T X_\infty x(t),$$

where $X_\infty$ is the stabilizing solution of the algebraic Riccati equation (ARE)

$$0 = C^T C + A^T X + XA - XBB^T X.$$
Preliminaries and Notation

Let $E, A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $p, m \ll n$ and $\text{rank}(E) = n$. Consider the linear quadratic regulator problem

$$
\min_{u \in U_{ad}} \mathcal{J}(u) = \int_{0}^{\infty} x^T C^T C x + u^T u \, dt
$$

constrained to the linear time invariant system

$$
E \dot{x}(t) = Ax(t) + Bu(t) \quad y(t) = Cx(t).
$$

Then the optimal state feedback control is given by

$$
u(t) = -B^T X_\infty E x(t), \quad X_\infty
$$

where $X_\infty$ is the stabilizing solution of the algebraic Riccati equation (ARE)

$$
0 = C^T C + A^T X E + E^T X A - E^T X B B^T X E.
$$
Preliminaries and Notation

Definition

We call $A$ stable, iff $\Lambda(A) \subset \mathbb{C}_{<0}$.
Preliminaries and Notation

**Definition**

We call $A$ stable, iff $\Lambda(A) \subset \mathbb{C}_{<0}$.

**Definition**

$(A, BB^T)$ is stabilizable, iff $\exists X \in \mathbb{R}^{n \times n}$ such that $A - BB^T X$ is stable.
### Preliminaries and Notation

#### Definition

We call $A$ stable, iff $\Lambda(A) \subseteq \mathbb{C}_{<0}$.

#### Definition

$(A, BB^T)$ is stabilizable, iff $\exists X \in \mathbb{R}^{n \times n}$ such that $A - BB^TX$ is stable.

#### Definition

$(C^TC, A)$ is detectable, iff $(A^T, C^TC)$ is stabilizable.
Preliminaries and Notation

Definition
We call $A$ stable, iff $\Lambda(A) \subset \mathbb{C}_{<0}$.

Definition
$(A, BB^T)$ is stabilizable, iff $\exists X \in \mathbb{R}^{n \times n}$ such that $A - BB^TX$ is stable.

Definition
$(C^TC, A)$ is detectable, iff $(A^T, C^TC)$ is stabilizable.

Assumption 1
- $(A, BB^T)$ is stabilizable,
- $(C^TC, A)$ is detectable and
- $X_0$ symmetric and positive semi-definite, such that $A - BB^TX_0$ is stable.
Preliminaries and Notation

Definition
We call \( A \) stable, iff \( \Lambda(A) \subset \mathbb{C}_{<0} \).

Definition
\((A, BB^T)\) is stabilizable, iff \( \exists X \in \mathbb{R}^{n \times n} \) such that \( A - BB^T X \) is stable.

Definition
\((C^T C, A)\) is detectable, iff \((A^T, C^T C)\) is stabilizable.

Assumption 1
- \((A, BB^T)\) is stabilizable,
- \((C^T C, A)\) is detectable and
- \(X_0\) symmetric and positive semi-definite, such that \( A - BB^T X_0 \) is stable.

In case \( E \neq I \):
\[ A \sim E^{-1} A \text{ and } B \sim E^{-1} B \]
Outline

1. Iterative Solution of the ARE
2. Solving Large Lyapunov Equations
3. LRCF-NM for the ARE
4. Inexact LRCF-NM For the ARE
5. Tests and Conclusions
Iterative Solution of the ARE

1. Iterative Solution of the ARE
   - Newton’s Method for AREs
   - Inexact Newton for AREs

2. Solving Large Lyapunov Equations

3. LRCF-NM for the ARE

4. Inexact LRCF-NM For the ARE

5. Tests and Conclusions
Iterative Solution of the ARE

Newton’s Method for AREs

Consider

\[ \mathcal{R}(X) := C^T C + A^T X + XA - XBB^T X = 0 \]

Newton’s Iteration for the ARE

\[ \mathcal{R}'|_{X_\ell}(N_\ell) = -\mathcal{R}(X_\ell), \quad X_{\ell+1} = X_\ell + N_\ell, \quad \ell = 0, 1, \ldots \]

where the Frechét derivative of \( \mathcal{R} \) at \( X \) is the Lyapunov operator

\[ \mathcal{R}'|_X : \quad Q \mapsto (A - BB^T X)^T Q + Q(A - BB^T X), \]

i.e., in every Newton step solve a

Lyapunov Equation

\[ (A - BB^T X_\ell)^T N_\ell + N_\ell (A - BB^T X_\ell) = -\mathcal{R}(X_\ell). \]
Iterative Solution of the ARE

Newton’s Method for AREs

Consider
\[ \mathcal{R}(X) := C^T C + A^T X + XA - XBB^T X = 0 \]

Kleinman’s Iteration for the ARE

\[ \mathcal{R}'|_{X_{\ell+1}}(X_{\ell+1}) = \mathcal{R}'|_{X_{\ell}}(X_{\ell}) - \mathcal{R}(X_{\ell}), \quad \ell = 0, 1, \ldots \]

where the Frechét derivative of \( \mathcal{R} \) at \( X \) is the Lyapunov operator

\[ \mathcal{R}'|_{X} : \quad Q \mapsto (A - BB^T X)^T Q + Q(A - BB^T X), \]

i.e., in every Newton step solve a

Lyapunov Equation

\[ (A - BB^T X_{\ell})^T X_{\ell+1} + X_{\ell+1}(A - BB^T X_{\ell}) = -C^T C - X_{\ell}BB^T X_{\ell}. \]
Iterative Solution of the ARE

Newton’s Method for AREs

Consider

\[ \mathcal{R}(X) := C^T C + A^T X + XA - XBB^T X = 0 \]

**Kleinman’s Iteration for the ARE**

\[ \mathcal{R}'|_{X_\ell}(X_{\ell+1}) = \mathcal{R}'|_{X_\ell}(X_\ell) - \mathcal{R}(X_\ell), \quad \ell = 0, 1, \ldots \]

where the Frechét derivative of \( \mathcal{R} \) at \( X \) is the Lyapunov operator

\[ \mathcal{R}'|_X : \quad Q \mapsto (A - BB^T X)^T Q + Q(A - BB^T X), \]

i.e., in every Newton step solve a

**Lyapunov Equation**

\[ F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T. \]
Iterative Solution of the ARE

Newton's Method for AREs

Consider

\[ \mathcal{R}(X) := C^T C + A^T X E + E^T X A - E^T X B B^T X E = 0 \]

Kleinman’s Iteration for the ARE \[\text{[Kleinman '68]}\]

\[ \mathcal{R}'|_{X_\ell}(X_{\ell+1}) = \mathcal{R}'|_{X_\ell}(X_\ell) - \mathcal{R}(X_\ell), \quad \ell = 0, 1, \ldots \]

where the Frechét derivative of \( \mathcal{R} \) at \( X \) is the Lyapunov operator

\[ \mathcal{R}'|_X : \quad Q \mapsto (A - B B^T X E)^T Q E + E^T Q (A - B B^T X E), \]

i.e., in every Newton step solve a

Lyapunov Equation \[\text{[Kleinman '68]}\]

\[ F_{\ell}^T X_{\ell+1} E + E^T X_{\ell+1} F_{\ell} = - \tilde{G}_{\ell} \tilde{G}_{\ell}^T. \]
Iterative Solution of the ARE

Newton’s Method for AREs (Convergence Result)

*e.g.* [Kleinman ’68, Mehrmann ’91, Lancaster/Rodman ’95]

**Theorem**

*Let Assumption 1 hold, then the iterates defined by

\[ F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T, \]

converge to the unique symmetric matrix \( X_\infty \), such that

- \( \mathcal{R}(X_\infty) = 0 \)
- and \( A - BB^T X_\infty \) is stable.

Furthermore the convergence is quadratic and monotone with

\[ 0 \leq X_\infty \leq \cdots \leq X_{k+1} \leq X_k \leq \cdots \leq X_1. \]
Consider \( \mathcal{A}(X) := C^T C + A^T X + XA - XBB^T X = 0 \)

**Inexact Newton’s Iteration for the ARE**

\[
\mathcal{A}'(X_{\ell}) + \mathcal{A}(X_{\ell}) = R_{\ell}, \quad X_{\ell+1} = X_{\ell} + N_{\ell}, \quad \ell = 0, 1, \ldots
\]

i.e., in every Newton step (approximately) solve a

**Lyapunov Equation**

\[
(A - BB^T X_{\ell})^T N_{\ell} + N_{\ell}(A - BB^T X_{\ell}) = -\mathcal{A}(X_{\ell}) + R_{\ell}.
\]
Iterative Solution of the ARE

Inexact Newton for AREs (Basic Concept)

Consider

\[ \mathcal{R}(X) := C^T C + A^T X + X A - X B B^T X = 0 \]

Inexact Kleinman’s Iteration for the ARE

\[ \mathcal{R}'|_{X_{\ell}(X_{\ell+1})} - \mathcal{R}'|_{X_{\ell}(X_{\ell})} + \mathcal{R}(X_{\ell}) = R_\ell, \quad \ell = 0, 1, \ldots \]

i.e., in every Newton step (approximately) solve a

Lyapunov Equation

\[ (A - B B^T X_{\ell})^T X_{\ell+1} + X_{\ell+1}(A - B B^T X_{\ell}) = -C^T C - X_{\ell} B B^T X_{\ell} + R_\ell. \]
Iterative Solution of the ARE

Inexact Newton for AREs (Basic Concept)

Consider

\[ \mathcal{R}(X) := C^T C + A^T X + XA - XBB^T X = 0 \]

Inexact Kleinman’s Iteration for the ARE

\[ \mathcal{R}'|_{x_\ell}(X_{\ell+1}) - \mathcal{R}'|_{x_\ell}(X_\ell) + \mathcal{R}(X_\ell) = R_\ell, \quad \ell = 0, 1, \ldots \]

i.e., in every Newton step (approximately) solve a

Lyapunov Equation

\[ F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T + R_\ell. \]
Iterative Solution of the ARE

Inexact Newton for AREs (Basic Concept)

Consider \( \mathcal{R}(X) := C^T C + A^T X E + E^T X A - E^T X B B^T X E = 0 \)

Inexact Kleinman’s Iteration for the ARE

\[
\mathcal{R}'|_{X_\ell(X_{\ell+1})} - \mathcal{R}'|_{X_\ell(X_\ell)} + \mathcal{R}(X_\ell) = R_\ell, \quad \ell = 0, 1, \ldots
\]

i.e., in every Newton step (approximately) solve a

Lyapunov Equation

\[
F_\ell^T X_{\ell+1} E + E^T X_{\ell+1} F_\ell = - \tilde{G}_\ell \tilde{G}_\ell^T + R_\ell.
\]
Iterative Solution of the ARE
Inexact Newton for AREs (Convergence Result)

[Feitzinger/Hylla/Sachs ’09, Hylla ’10]

Theorem

Let Assumption 1 hold,

\[ 0 \leq R_\ell \leq C^T C \quad \text{and} \quad 0 \leq R_\ell \leq N_\ell BB^T N_\ell. \]

Then the iterates defined by

\[ F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T + R_\ell, \]

converge to the unique symmetric matrix \( X_\infty \), such that

- \( \Re(X_\infty) = 0 \)
- and \( A - BB^T X_\infty \) is stable.

Furthermore the convergence is quadratic and monotone with

\[ 0 \leq X_\infty \leq \cdots \leq X_{k+1} \leq X_k \leq \cdots \leq X_1. \]
Iterative Solution of the ARE

Inexact Newton for AREs (Convergence Result)

Theorem

Let Assumption 1 hold,

\[ 0 \leq R_\ell \leq C^T C \quad \text{and} \quad 0 \leq R_\ell \leq N_\ell BB^T N_\ell. \]

Then the iterates defined by

\[ F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T + R_\ell, \]

converge to the unique symmetric matrix \( X_\infty \), such that

- \( R(X_\infty) = 0 \)
- and \( A - BB^T X_\infty \) is stable.

Furthermore the convergence is \textit{quadratic and monotone} with

\[ 0 \leq X_\infty \leq \cdots \leq X_{k+1} \leq X_k \leq \cdots \leq X_1. \]
Iterative Solution of the ARE

Inexact Newton for AREs (Remarks)

Weaker Condition

Replacing

\[ R_\ell \leq C^T C \]

by

\[ R_\ell \leq C^T C + X_\ell B B^T X_\ell \]

keeps the iteration well defined.
Weaker Condition

Replacing

\[ R_\ell \leq C^T C \]

by

\[ R_\ell \leq C^T C + X_\ell BB^T X_\ell \]

keeps the iteration well defined.

Large Scale Difficulty

None of the conditions

- \( R_\ell \leq C^T C \),
- \( R_\ell \leq C^T C + X_\ell BB^T X_\ell \), and
- \( 0 \leq R_\ell \leq N_\ell BB^T N_\ell \),

can be tested in large scale applications.
**Weaker Condition**

Replacing

\[ R_\ell \leq C^T C \]

by

\[ R_\ell \leq C^T C + X_\ell BB^T X_\ell \]

keeps the iteration well defined.

**Large Scale Difficulty**

None of the conditions

- \( R_\ell \leq C^T C \)
- \( R_\ell \leq C^T C + X_\ell BB^T X_\ell \), and
- \( 0 \leq R_\ell \leq N_\ell BB^T N_\ell \)

can be tested in large scale applications.

Crucial for the proof. ⇒ Lyapunov solver needs to ensure this.
Solving Large Lyapunov Equations

1. Iterative Solution of the ARE

2. Solving Large Lyapunov Equations
   - LRCF-ADI
   - LRCF-ADI with Galerkin-Projection-Acceleration

3. LRCF-NM for the ARE

4. Inexact LRCF-NM For the ARE

5. Tests and Conclusions
Solving Large Lyapunov Equations

**LRCF-ADI**

Consider

$$FX + XF^T = -GG^T \quad F \in \mathbb{R}^{n \times n}, \ G \in \mathbb{R}^{n \times p}$$

**Task**

Find $Z \in \mathbb{K}^{n \times nz}$, such that $nZ \ll n$ and $X \approx ZZ^H$

**Algorithm**

$$V_1 = \sqrt{-2p_1(F + p_1 I)^{-1}}G, \quad Z_1 = V_1$$

$$V_i = \frac{\sqrt{p_i}}{\sqrt{p_{i-1}}} \left[ I - (p_i + p_{i-1})(F + p_i I)^{-1} \right] V_{i-1} \quad Z_i = [Z_{i-1} V_i]$$

For certain shift parameters $\{p_1, ..., p_J\} \subset \mathbb{C}_{<0}$.

Stop if

- $\|V_i V_i^H\|$ is small, or
- $\|FZ_i Z_i^H + Z_i Z_i^H F^T + GG^T\|$ is small.
Solving Large Lyapunov Equations

LRCF-ADI

Consider

\[ FX + XF^T = -GG^T \quad F \in \mathbb{R}^{n \times n}, G \in \mathbb{R}^{n \times p} \]

Task

Find \( Z \in \mathbb{K}^{n, n_Z} \), such that \( n_Z \ll n \) and \( X \approx ZZ^H \)

Algorithm

\[
V_1 = \sqrt{-2p_1(F + p_1 I)^{-1}}G, \quad Z_1 = V_1
\]

\[
V_i = \frac{\sqrt{p_i}}{\sqrt{p_{i-1}}} [I - (p_i + \frac{1}{p_{i-1}})(F + p_i I)^{-1}] V_{i-1} \quad Z_i = [Z_{i-1} V_i]
\]

For certain shift parameters \( \{p_1, ..., p_J\} \subset \mathbb{C}_{<0} \).

Stop if

\[
\|V_i V_i^H\| \text{ is small, or } \|FZ_i Z_i^H + Z_i Z_i^T F^T + GG^T\| \text{ is small.}
\]

\[\text{[HYLLA '10]: Then } \forall i : R_i \geq 0, \quad \text{and } \exists i_0 \quad \forall i \geq i_0 \quad R_i \leq C^T C.\]
Solving Large Lyapunov Equations

\[ G-LRCF-ADI \]

Consider
\[
FX E^T + EXF^T = -GG^T, \quad E, F \in \mathbb{R}^{n \times n}, G \in \mathbb{R}^{n \times p}
\]

Task
Find \( Z \in \mathbb{K}^{n, n_z} \), such that \( n_z \ll n \) and \( X \approx ZZ^H \)

Algorithm

\[
V_1 = \sqrt{-2p_1}(F + p_1E)^{-1}G, \quad Z_1 = V_1
\]
\[
V_i = \frac{\sqrt{p_i}}{\sqrt{p_{i-1}}} \left[ I - (p_i + p_{i-1})(F + p_iE)^{-1} \right] EV_{i-1} \quad Z_i = [Z_{i-1} V_i]
\]

For certain shift parameters \( \{p_1, ..., p_J\} \subset \mathbb{C}_{<0} \).

Stop if
- \( \|V_i V_i^H\| \) is small, or
- \( \|FZ_i Z_i^H E^T + EZ_i Z_i^H F^T + GG^T\| \) is small.
Solving Large Lyapunov Equations

G-LRCF-ADI

Consider

\[ FXE^T + EXF^T = -GG^T \quad E, F \in \mathbb{R}^{n \times n}, G \in \mathbb{R}^{n \times p} \]

Task

Find \( Z \in \mathbb{K}^{n, nz} \), such that \( nz \ll n \) and \( X \approx ZZ^H \)

Algorithm

\[
V_1 = \sqrt{-2p_1(F + p_1E)^{-1}}G, \quad Z_1 = V_1 \\
V_i = \frac{\sqrt{p_i}}{\sqrt{p_{i-1}}} \left[ I - (p_i + p_{i-1})(F + p_iE)^{-1} \right] EV_{i-1} \quad Z_i = [Z_{i-1}V_i]
\]

For certain shift parameters \( \{p_1, ..., p_J\} \subset \mathbb{C}_{<0} \).

Stop if

- \( \|V_iV_i^H\| \) is small, or
- \( \|FZ_iZ_i^HET + EZ_iZ_i^HF^T + GG^T\| \) is small.

Patrick showed: Can ensure \( Z \in \mathbb{R}^{n, nz} \) even if \( \{p_1, ..., p_J\} \notin \mathbb{R} \)
Solving Large Lyapunov Equations
LRCF-ADI with Galerkin-Projection-Acceleration

Legend:
- new factor
- original matrix
- projected matrix
- projected Cholesky factor
- old factor
- original rhs
- projected rhs

\[ F Z \mathbf{Z}^T + \mathbf{Z} \mathbf{Z}^T F^T = -G \]
Solving Large Lyapunov Equations

LRCF-ADI with Galerkin-Projection-Acceleration

\[
\begin{align*}
F & \quad Z^T \\
F_m & \quad Z \\
F^T & \quad -G
\end{align*}
\]

Legend:
- new factor
- original matrix
- projected matrix
- projected Cholesky factor
- old factor
- original rhs
- projected rhs
Solving Large Lyapunov Equations

LRCF-ADI with Galerkin-Projection-Acceleration

\[ F_m C_m C_m^T + C_m C_m^T = -G_m G_m^T \]

Legend:
- new factor
- old factor
- original matrix
- projected matrix
- original rhs
- projected rhs
Solving Large Lyapunov Equations
LRCF-ADI with Galerkin-Projection-Acceleration

\[ F_m \quad + \quad C_m C_m^T \quad = \quad -G_m G_m^T \]

Legend:
- new factor
- old factor
- original matrix
- projected matrix
- original rhs
- projected rhs
- projected Cholesky factor

Max Planck Institute Magdeburg

Jens Saak, *Inexact Newton for Large AREs*
Solving Large Lyapunov Equations
LRCF-ADI with Galerkin-Projection-Acceleration

**Projected ADI Step → LRCF-ADI-GP**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Compute the LRCF-ADI iterate $Z_i$</td>
</tr>
<tr>
<td>2</td>
<td>Compute orthogonal basis via QR factorization: $Q_i R_i \Pi_i = Z_i^a$</td>
</tr>
<tr>
<td>3</td>
<td>Solve (for $\tilde{Z}$) the projected Lyapunov equation</td>
</tr>
<tr>
<td>$$(Q_i^T F Q_i) \tilde{Z} \tilde{Z}^T + \tilde{Z} \tilde{Z}^T (Q_i^T F^T Q_i) = -Q_i^T G G^T Q_i$$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Update $Z_i$ according to $Z_i := Q_i \tilde{Z}$</td>
</tr>
</tbody>
</table>

---

*S.'09, Benner/S.'10

---

*a* economy size QR with column pivoting; crucial to compute correct subspace if $Z_i$ (almost) rank deficient.
## Solving Large Lyapunov Equations
### LRCF-ADI with Galerkin-Projection-Acceleration

### Projected ADI Step → LRCF-ADI-GP

1. Compute the LRCF-ADI iterate $Z_i$
2. Compute orthogonal basis via QR factorization: $Q_i R_i \Pi_i = Z_i$
3. Solve (for $\tilde{Z}$) the projected Lyapunov equation
   
   $$(Q_i^T F Q_i) \tilde{Z} \tilde{Z}^T + \tilde{Z} \tilde{Z}^T (Q_i^T F^T Q_i) = -Q_i^T G G^T Q_i$$
4. Update $Z_i$ according to $Z_i := Q_i \tilde{Z}$

### Tests and Conclusions
- Ensure projected systems remain stable, e.g., $F + F^T < 0$
- Orthogonalization can be avoided
- Perform projected ADI step only every $k$-th step (e.g. $k = 5$)
- Evaluate residuals only in projected ADI steps

[S.’09, Benner/S.’10]
Solving Large Lyapunov Equations

LRCF-ADI with Galerkin-Projection-Acceleration

1. Compute the LRCF-ADI iterate $Z_i$

2. Compute orthogonal basis via QR factorization:
$$Q_i R_i \Pi_i = Z_i$$

3. Solve (for $\tilde{Z}$) the projected Lyapunov equation
$$\begin{align*}
(Q_i^T F Q_i) \tilde{Z} \tilde{Z}^T + \tilde{Z} \tilde{Z}^T (Q_i^T F^T Q_i) &= -Q_i^T G G^T Q_i
\end{align*}$$

4. Update $Z_i$ according to $Z_i := Q_i \tilde{Z}$

- Ensure projected systems remain stable, e.g., $F + F^T < 0$
- Orthogonalization can be avoided
- Perform projected ADI step only every $k$-th step (e.g. $k = 5$)
- Evaluate residuals only in projected ADI steps

Exploit $P^\perp = Z_i (Z_i^T Z_i)^{-1} Z_i^T$

solving generalized projected LE:

$$\begin{align*}
(Z_i^T F Z_i) \tilde{Z} \tilde{Z}^T Z_i + Z_i^T Z_i \tilde{Z} \tilde{Z}^T (Z_i^T F^T Z_i) &= -Z_i^T G G^T Z_i
\end{align*}$$

using $Q_i := Z_i L_i$ where $(Z_i^T Z_i)^{-1} = L_i L_i^T$
Solving Large Lyapunov Equations
LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step → G-LRCF-ADI-GP

1. Compute the G-LRCF-ADI iterate \( Z_i \)
2. Compute orthogonal basis via QR factorization: \( Q_i R_i \Pi_i = Z_i \)
3. Solve (for \( \tilde{Z} \)) the projected Lyapunov equation

\[
(Q_i^T F Q_i) \tilde{Z} \tilde{Z}^T (Q_i^T E^T Q_i) + (Q_i^T E Q_i) \tilde{Z} \tilde{Z}^T (Q_i^T F^T Q_i) = -Q_i^T G G^T Q_i
\]

4. Update \( Z_i \) according to \( Z_i := Q_i \tilde{Z} \)

Using \( E \) orthogonalization breaks accuracy, when \( \| . \|_2 \) is used in stopping criteria.

Reason: \( \| A \|_2 \geq \sqrt{\| E \|_2 \| A \|_E} \)
LRCF-NM for the ARE

1. Iterative Solution of the ARE

2. Solving Large Lyapunov Equations

3. LRCPF-NM for the ARE
   - Low-Rank Newton-ADI (LRCPF-NM) for AREs
   - Low-Rank Newton-ADI Variants

4. Inexact LRCPF-NM For the ARE

5. Tests and Conclusions
LRCF-NM for the ARE

Low-Rank Newton-ADI (LRCF-NM) for AREs

Consider

\[ \mathcal{R}(X) := C^T C + A^T X + XA - XBB^T X = 0 \]

Kleinman’s Iteration for the ARE

\[ \mathcal{R}'|_{X_{\ell}}(X_{\ell+1}) = \mathcal{R}'|_{X_{\ell}}(X_{\ell}) - \mathcal{R}(X_{\ell}), \quad \ell = 0, 1, \ldots \]

where the Frechét derivative of \( \mathcal{R} \) at \( X \) is the Lyapunov operator

\[ \mathcal{R}'|_{X} : Q \mapsto (A - BB^T X)^T Q + Q(A - BB^T X), \]

i.e., in every Newton step solve a

Lyapunov Equation

\[ (A - BB^T X) X_{\ell+1} + X_{\ell+1}(A - BB^T X) = -C^T C - X_{\ell} BB^T X_{\ell}. \]
LRCF-NM for the ARE
Low-Rank Newton-ADI (LRCF-NM) for AREs

Factored Newton-Kleinman Iteration

\[
F_\ell = A - BB^T X_\ell =: A - BK_\ell
\]

\[
G_\ell = [C^T \; K_\ell^T]
\]

Find low rank factor \(Z_\ell \in \mathbb{R}^{n,n_Z}\), where \(n_Z \ll n\) and \(X_\ell = Z_\ell Z_\ell^T\).
**Factored Newton-Kleinman Iteration**

\[
F_\ell = A - BB^T X_\ell =: A - BK_\ell \\
G_\ell = [C^T K_\ell^T]
\]

- apply LRCF-ADI in every Newton step
- exploit structure of \(F_\ell\) using **Sherman-Morrison-Woodbury formula**

\[
(A - BK_\ell + p_k^{(\ell)} I_n)^{-1} = \\
(I_n + (A + p_k^{(\ell)} I_n)^{-1} B(I_m - K_\ell(A + p_k^{(\ell)} I_n)^{-1} B)^{-1} K_\ell)(A + p_k^{(\ell)} I_n)^{-1}
\]
**Factored Newton-Kleinman Iteration**  

\[
F_\ell = A - BB^T X_\ell =: A - BK_\ell \\
G_\ell = [C^T \ K_\ell^T]
\]

- apply LRCF-ADI in every Newton step
- exploit structure of \( F_\ell \) using Sherman-Morrison-Woodbury formula

\[
(A - BK_\ell + p_k^{(\ell)} I_n)^{-1} = (I_n + (A + p_k^{(\ell)} I_n)^{-1} B(I_m - K_\ell(A + p_k^{(\ell)} I_n)^{-1} B)^{-1} K_\ell)(A + p_k^{(\ell)} I_n)^{-1}
\]
## LRCF-NM for the ARE

### Low-Rank Newton-ADI (LRCF-NM) for AREs

**Factored Newton-Kleinman Iteration**

\[
F_\ell = A - BB^TX_\ell =: A - BK_\ell \\
G_\ell = [C^T K_\ell^T]
\]

- apply LRCF-ADI in every Newton step
- exploit structure of \(F_\ell\) using Sherman-Morrison-Woodbury formula

\[
(A - BK_\ell + p_k^{(\ell)} I_n)^{-1} = \\
(I_n + (A + p_k^{(\ell)} I_n)^{-1} B(I_m - K_\ell(A + p_k^{(\ell)} I_n)^{-1} B)^{-1} K_\ell)(A + p_k^{(\ell)} I_n)^{-1}
\]
Factored Newton-Kleinman Iteration

\[ F_\ell = A - BB^T X_\ell =: A - BK_\ell \]
\[ G_\ell = [C^T K_\ell^T] \]

- apply LRCF-ADI in every Newton step
- exploit structure of \( F_\ell \) using Sherman-Morrison-Woodbury formula

\[
(A - BK_\ell + p_k^{(\ell)} I_n)^{-1} = \]
\[
(I_n + (A + p_k^{(\ell)} I_n)^{-1} B (I_m - K_\ell (A + p_k^{(\ell)} I_n)^{-1} B)^{-1} K_\ell) (A + p_k^{(\ell)} I_n)^{-1}
\]
Factored Newton-Kleinman Iteration [Benner/Li/Penzl ’99/’08]

\[
F_\ell = A - BB^T X_\ell E =: A - BK_\ell
\]
\[
G_\ell = [C^T \ K_\ell^T]
\]

- apply LRCF-ADI in every Newton step
- exploit structure of \( F_\ell \) using Sherman-Morrison-Woodbury formula

\[
(A - BK_\ell + p_k^{(\ell)} E)^{-1} =
\]
\[
(I_n + (A + p_k^{(\ell)} E)^{-1} B(I_m - K_\ell(A + p_k^{(\ell)} E)^{-1} B)^{-1} K_\ell)(A + p_k^{(\ell)} E)^{-1}
\]
LRCF-NM for the ARE

Low-Rank Newton-ADI (LRCF-NM) for AREs

Algorithm 1 Low-Rank Cholesky Factor Newton Method (LRCF-NM)

Input: $A, B, C, K^{(0)}$ for which $A - BK^{(0)}^T$ is stable
Output: $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of

$$CTC + A^TX + XA - XBB^TX = 0.$$

1: for $k = 1, 2, \ldots, k_{\text{max}}$ do
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \ldots$ with respect to the matrix $F^{(k)} = A^T - K^{(k-1)}B^T$.
3: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: Compute $Z^{(k)}$ using (LRCF-ADI) such that

$$F^{(k)}Z^{(k)}Z^{(k)H} + Z^{(k)}Z^{(k)H}F^{(k)}^T \approx -G^{(k)}G^{(k)}^T.$$

5: $K^{(k)} = Z^{(k)}(Z^{(k)H}B)$
6: end for
Algorithm 1 Low-Rank Cholesky Factor Newton Method (G-LRCF-NM)

Input: $E, A, B, C, K^{(0)}$ for which $A - BK^{(0)}^T$ is stable
Output: $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of
\[ C^T C + A^T X E + E^T X A - E^T X B B^T X E = 0. \]

1: for $k = 1, 2, \ldots, k_{\text{max}}$ do
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \ldots$ with respect to the matrix $F^{(k)} = A^T E^{-T} - K^{(k-1)} B^T E^{-T}$.
3: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: Compute $Z^{(k)}$ using (G-LRCF-ADI) such that
\[ F^{(k)} Z^{(k)} Z^{(k)^H} E + E^T Z^{(k)} Z^{(k)^H} F^{(k)^T} \approx -G^{(k)} G^{(k)^T}. \]
5: $K^{(k)} = E^T (Z^{(k)} (Z^{(k)^H} B))$
6: end for
Simplified Factored Newton Iteration

- Gradient updates are *cheap*
LRCF-NM for the ARE
Low-Rank Newton-ADI Variants

Simplified Factored Newton Iteration

- Gradient updates are *cheap*
- Reuse ADI shifts instead
  1. Reuse shifts from first step.
  2. Compute shift according to desired closed loop matrix.
LRCF-NM for the ARE
Low-Rank Newton-ADI Variants

Simplified Factored Newton Iteration
- Gradient updates are *cheap*
- Reuse ADI shifts instead
  1. Reuse shifts from first step.
  2. Compute shift according to desired closed loop matrix.

Factored Newton-Galerkin Iteration
[S. ’09, Benner/S. ’10]
1. Apply (G-)LRCF-ADI-GP in every Newton step. (ADI loop)
LRCF-NM for the ARE
Low-Rank Newton-ADI Variants

Simplified Factored Newton Iteration
- Gradient updates are *cheap*
- Reuse ADI shifts instead
  1. Reuse shifts from first step.
  2. Compute shift according to desired closed loop matrix.

Factored Newton-Galerkin Iteration
[S. ’09, Benner/S. ’10]
1. Apply (G-)LRCF-ADI-GP in every Newton step. (ADI loop)
2. Add Galerkin projection for ARE. (Newton loop)
LRCF-NM for the ARE
Low-Rank Newton-ADI Variants

Simplified Factored Newton Iteration

- Gradient updates are *cheap*
- Reuse ADI shifts instead
  1. Reuse shifts from first step.
  2. Compute shift according to desired closed loop matrix.

Factored Newton-Galerkin Iteration

1. Apply (G-)LRCF-ADI-GP in every Newton step. (ADI loop)
2. Add Galerkin projection for ARE. (Newton loop)

Inexact Factored Newton-Kleinman Iteration

- Control (G-)LRCF-ADI accuracy according to Newton progress.
LRCF-NM for the ARE
Low-Rank Newton-ADI Variants

**Algorithm 1** Low-Rank Cholesky Factor Newton Method (LRCF-NM)

**Input:** $A, B, C, K^{(0)}$ for which $A - BK^{(0)}^T$ is stable  
**Output:** $Z = Z^{(k_{max})}$, such that $ZZ^H$ approximates the solution $X$ of  
$$C^T C + A^T X + XA - XBB^TX = 0.$$  

1: **for** $k = 1, 2, \ldots, k_{max}$ **do**  
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \ldots$  
   with respect to the matrix $F^{(k)} = A^T - K^{(k-1)} B^T$.  
3: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$  
4: Compute $Z^{(k)}$ using (LRCF-ADI) such that  
   $$F^{(k)} Z^{(k)} Z^{(k)^H} + Z^{(k)} Z^{(k)^H} F^{(k)^T} = -G^{(k)} G^{(k)^T}.$$  
5: $K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$  
6: **end for**
Algorithm 2 Simpl. Low-Rank Cholesky Factor Newton Method

**Input:** $A, B, C, K^{(0)}$ for which $A - BK^{(0)\top}$ is stable
**Output:** $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of

$$C^T C + A^T X + XA - XBB^T X = 0.$$

1: Determine (sub)optimal ADI shift parameters $p_1, p_2, \ldots$ with respect to the matrix $F^{(0)} = A^T - K^{(0)\top} B^T$.
2: for $k = 1, 2, \ldots, k_{\text{max}}$ do
3: \quad $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: \quad Compute $Z^{(k)}$ using (LRCF-ADI) such that

$$F^{(k)} Z^{(k)} Z^{(k)\top} + Z^{(k)} Z^{(k)\top} F^{(k)\top} = -G^{(k)} G^{(k)\top}.$$
5: \quad $K^{(k)} = Z^{(k)} (Z^{(k)\top} B)$
6: end for
LRCF-NM for the ARE
Low-Rank Newton-ADI Variants

**Algorithm 3** Low-Rank Cholesky Factor Galerkin-Newton Method (LRCF-NM-GP)

**Input:** $A, B, C, K^{(0)}$ for which $A - BK^{(0)}^T$ is stable

**Output:** $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of

$$C^TC + A^TX +XA - XBB^TX = 0.$$

1: **for** $k = 1, 2, \ldots, k_{\text{max}}$ **do**
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \ldots$ with respect to the matrix $F^{(k)} = A^T - K^{(k-1)}B^T$.
3: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: Compute $Z^{(k)}$ using (LRCF-ADI-GP) such that

$$F^{(k)}Z^{(k)}Z^{(k)H} + Z^{(k)}Z^{(k)H}F^{(k)T} = -G^{(k)}G^{(k)T}.$$  

5: Project ARE, solve and prolongate solution
6: $K^{(k)} = Z^{(k)}(Z^{(k)H}B)$
7: **end for**
Algorithm 4 Simpl. Low-Rank Cholesky Factor Galerkin-Newton Method

(LRCF-NM-S-GP)

Input: $A, B, C, K^{(0)}$ for which $A - BK^{(0)}^T$ is stable
Output: $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of

$$C^T C + A^T X + XA - XBB^T X = 0.$$ 

1: Determine (sub)optimal ADI shift parameters $p_1, p_2, \ldots$
   with respect to the matrix $F^{(0)} = A^T - K^{(0)} B^T$.
2: for $k = 1, 2, \ldots, k_{\text{max}}$ do
3:   $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4:   Compute $Z^{(k)}$ using (LRCF-ADI-GP) such that
   $$F^{(k)} Z^{(k)} Z^{(k)^H} + Z^{(k)} Z^{(k)^H} F^{(k)^T} = -G^{(k)} G^{(k)^T}.$$ 
5:   Project ARE, solve and prolongate solution
6:   $K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$
7: end for
Algorithm 5 Inexact Low-Rank Cholesky Factor Newton Method (I-LRCF-NM)

Input: $A, B, C, K^{(0)}$ for which $A - BK^{(0)}^T$ is stable
Output: $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of

$$C^T C + A^T X +XA - XBB^T X = 0.$$ 

1: for $k = 1, 2, \ldots, k_{\text{max}}$ do
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \ldots$ with respect to the matrix $F^{(k)} = A^T - K^{(k-1)}B^T$.
3: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: Compute $Z^{(k)}$ using (LRCF-ADI) such that

$$F^{(k)} Z^{(k)} Z^{(k)H} + Z^{(k)} Z^{(k)H} F^{(k)T} \approx -G^{(k)} G^{(k)T}.$$ 

5: $K^{(k)} = Z^{(k)} (Z^{(k)H} B)$
6: end for
Algorithm 6 \textbf{Inexact} Low-Rank Cholesky Factor Newton Method

\textbf{(I-LRCF-NM-S)}

\textbf{Input:} $A$, $B$, $C$, $K^{(0)}$ for which $A - BK^{(0)\top}$ is stable

\textbf{Output:} $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of

$$C^T C + A^T X + XA - XBB^T X = 0.$$ 

1: Determine (sub)optimal ADI shift parameters $p_1, p_2, \ldots$ with respect to the matrix $F^{(0)} = A^T - K^{(0)}B^T$. 
2: \textbf{for} $k = 1, 2, \ldots, k_{\text{max}}$ \textbf{do}
3: \hspace{1em} $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: \hspace{1em} Compute $Z^{(k)}$ using (LRCF-ADI) such that

$$F^{(k)}Z^{(k)}Z^{(k)\top} + Z^{(k)}Z^{(k)\top}F^{(k)\top} \approx - G^{(k)}G^{(k)\top}.$$ 
5: \hspace{1em} $K^{(k)} = Z^{(k)}(Z^{(k)\top}B)$
6: \textbf{end for}
LRCF-NM for the ARE
Low-Rank Newton-ADI Variants

Algorithm 7 Inexact Low-Rank Cholesky Factor Galerkin-Newton Method (I-LRCF-NM-GP)

Input: $A, B, C, K^{(0)}$ for which $A - BK^{(0)}^T$ is stable
Output: $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of

$$C^T C + A^T X + XA - XBB^TX = 0.$$ 

1: for $k = 1, 2, \ldots, k_{\text{max}}$ do
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \ldots$ with respect to the matrix $F^{(k)} = A^T - K^{(k-1)}B^T$.
3: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: Compute $Z^{(k)}$ using (LRCF-ADI-GP) such that

$$F^{(k)}Z^{(k)}Z^{(k)H} + Z^{(k)}Z^{(k)H}F^{(k)} \approx - G^{(k)}G^{(k)T}.$$ 

5: Project ARE, solve and prolongate solution
6: $K^{(k)} = Z^{(k)}(Z^{(k)H}B)$
7: end for
Algorithm 8 Inexact Low-Rank Cholesky Factor Galerkin-Newton Method (I-LRCF-NM-S-GP)

Input: $A, B, C, K^{(0)}$ for which $A - BK^{(0)}^T$ is stable
Output: $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of

$$C^T C + A^T X + XA - XBB^T X = 0.$$ 

1: Determine (sub)optimal ADI shift parameters $p_1, p_2, \ldots$ with respect to the matrix $F^{(0)} = A^T - K^{(0)}B^T$.
2: for $k = 1, 2, \ldots, k_{\text{max}}$ do
3: \hspace{1em} $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: \hspace{1em} Compute $Z^{(k)}$ using (LRCF-ADI-GP) such that
5: \hspace{2em} $F^{(k)} Z^{(k)} Z^{(k)^H} + Z^{(k)} Z^{(k)^H} F^{(k)^T} \approx - G^{(k)} G^{(k)^T}$.
6: Project ARE, solve and prolongate solution
7: \hspace{1em} $K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$
8: end for
Inexact LRCF-NM For the ARE

1. Iterative Solution of the ARE
2. Solving Large Lyapunov Equations
3. LRCF-NM for the ARE
4. Inexact LRCF-NM For the ARE
   - Accuracy control for the (G-)LRCF-ADI
   - Implementation
5. Tests and Conclusions
Main Problem:

How can we ensure quadratic convergence without checking

\[ 0 \leq R_\ell \leq C^T C \quad \text{and} \quad 0 \leq R_\ell \leq N_\ell BB^T N_\ell? \]
Main Problem:

How can we ensure quadratic convergence without checking

\[ 0 \leq R_\ell \leq C^T C \quad \text{and} \quad 0 \leq R_\ell \leq N_\ell BB^T N_\ell? \]

Due to the quadratic nature of \( R(.) \) we have

\[
R(Y) = R(X) + R'|_X (Y - X) + \frac{1}{2} R''|_X (Y - X, Y - X).
\]
Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

Main Problem:

How can we ensure quadratic convergence without checking

\[ 0 \leq R_\ell \leq C^T C \quad \text{and} \quad 0 \leq R_\ell \leq N_\ell B B^T N_\ell \? \]

Due to the quadratic nature of \( R(.) \) we have

\[
R(Y) = R(X) + R'|_X (Y - X) + \frac{1}{2} R''|_X (Y - X, Y - X). 
\]

Recall the Inexact Kleinman step:

\[
R_\ell = R'|_{X_\ell} (X_{\ell+1}) - R'|_{X_\ell} (X_\ell) + R(X_\ell) 
\]
Main Problem:

How can we ensure quadratic convergence without checking

\[ 0 \leq R_\ell \leq C^T C \quad \text{and} \quad 0 \leq R_\ell \leq N_\ell BB^T N_\ell? \]

Due to the quadratic nature of \( \mathcal{K}(.) \) we have

\[ \mathcal{K}(Y) = \mathcal{K}(X) + \mathcal{K}'|X(Y - X) + \frac{1}{2} \mathcal{K}''|X(Y - X, Y - X). \]

Recall the Inexact Kleinman step:

\[ R_\ell = \mathcal{K}'|X_\ell(X_{\ell+1}) - \mathcal{K}'|X_\ell(X_\ell) + \mathcal{K}(X_\ell) = \mathcal{K}'|X_\ell(X_{\ell+1} - X_\ell) + \mathcal{K}(X_\ell), \]
Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

Main Problem:

How can we ensure quadratic convergence without checking

\[ 0 \leq R_\ell \leq C^T C \quad \text{and} \quad 0 \leq R_\ell \leq N_\ell BB^T N_\ell \, ? \]

Due to the quadratic nature of \( \mathcal{R}(\cdot) \) we have

\[ \mathcal{R}(Y) = \mathcal{R}(X) + \mathcal{R}'|x(Y - X) + \frac{1}{2} \mathcal{R}''|x(Y - X, Y - X). \]

Recall the Inexact Kleinman step:

\[ R_\ell = \mathcal{R}'|x_\ell (X_{\ell+1}) - \mathcal{R}'|x_\ell (X_\ell) + \mathcal{R}(X_\ell) = \mathcal{R}'|x_\ell (X_{\ell+1} - X_\ell) + \mathcal{R}(X_\ell), \]

and thus

\[ \mathcal{R}(X_{\ell+1}) = R_\ell + \frac{1}{2} \mathcal{R}''|x_\ell (X_{\ell+1} - X_\ell, X_{\ell+1} - X_\ell). \]
Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

Main Problem:

How can we ensure quadratic convergence without checking

\[ 0 \leq R_\ell \leq C^T C \quad \text{and} \quad 0 \leq R_\ell \leq N_\ell BB^T N_\ell? \]

Due to the quadratic nature of \( R(\cdot) \) we have

\[ R(Y) = R(X) + R'(X)(Y - X) + \frac{1}{2} R''(X)(Y - X, Y - X). \]

Recall the Inexact Kleinman step:

\[ R_\ell = R'(X_\ell)(X_{\ell+1}) - R'(X_\ell)(X_\ell) + R(X_\ell) = R'(X_\ell)(X_{\ell+1} - X_\ell) + R(X_\ell), \]

and thus

\[ R(X_{\ell+1}) = R_\ell + \frac{1}{2} R''(X_\ell)(X_{\ell+1} - X_\ell, X_{\ell+1} - X_\ell). \]
Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

Main Problem:

How can we ensure quadratic convergence without checking
\[ 0 \leq R_{\ell} \leq C^T C \quad \text{and} \quad 0 \leq R_{\ell} \leq N_{\ell} BB^T N_{\ell} \? \]

Due to the quadratic nature of \( R(\cdot) \) we have
\[
R(Y) = R(X) + R'|_X (Y - X) + \frac{1}{2} R''|_X (Y - X, Y - X).
\]

Recall the Inexact Kleinman step:
\[
R_{\ell} = R'|_X (X_{\ell+1}) - R'|_X (X_{\ell}) + R(X_{\ell}) = R'|_X (X_{\ell+1} - X_{\ell}) + R(X_{\ell}),
\]
and thus
\[
R(X_{\ell+1}) = R_{\ell} + \frac{1}{2} R''|_X (N_{\ell}, N_{\ell}).
\]
Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

Main Problem:

How can we ensure quadratic convergence without checking

\[ 0 \leq R_\ell \leq C^T C \quad \text{and} \quad 0 \leq R_\ell \leq N_\ell BB^T N_\ell? \]

Due to the quadratic nature of \( R(\cdot) \) we have

\[ R(Y) = R(X) + R'|_X (Y - X) + \frac{1}{2} R''|_X (Y - X, Y - X). \]

Recall the Inexact Kleinman step:

\[ R_\ell = R'|_X (X_{\ell+1}) - R'|_X (X_\ell) + R(X_\ell) = R'|_X (X_{\ell+1} - X_\ell) + R(X_\ell), \]

and thus

\[ R(X_{\ell+1}) = R_\ell + \frac{1}{2} N_\ell BB^T N_\ell. \]
New Question

How can we exploit \( \mathcal{R}(X_{\ell+1}) = R_\ell + \frac{1}{2} N_\ell B B^T N_\ell \) to control the ADI accuracy?
Inexact LRCF-NM For the ARE
Accuracy control for the (G-)LRCF-ADI

New Question

How can we exploit $\mathcal{R}(X_{\ell+1}) = R_\ell + \frac{1}{2} N_\ell B B^T N_\ell$ to control the ADI accuracy?

Riccati residual
inner Lyapunov residual

$N_\ell B B^T N_\ell = (X_{\ell+1} - X_\ell) B B^T (X_{\ell+1} - X_\ell)$
Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

New Question

How can we exploit \( R(X_{\ell+1}) = R_\ell + \frac{1}{2} N_\ell BB^T N_\ell \) to control the ADI accuracy?

\[
N_\ell BB^T N_\ell = (X_{\ell+1} - X_\ell)BB^T (X_{\ell+1} - X_\ell)
= X_{\ell+1}BB^T X_{\ell+1} + X_\ell BB^T X_\ell - X_\ell BB^T X_{\ell+1} - X_{\ell+1} BB^T X_\ell
\]
Inexact LR CF-NM For the ARE
Accuracy control for the (G-)LR CF-ADI

New Question

How can we exploit $\mathcal{R}(X_{\ell+1}) = R_\ell + \frac{1}{2} N_\ell BB^T N_\ell$ to control the ADI accuracy?

Riccati residual
inner Lyapunov residual

$N_\ell BB^T N_\ell = (X_{\ell+1} - X_\ell) BB^T (X_{\ell+1} - X_\ell)$
$= X_{\ell+1} BB^T X_{\ell+1} + X_\ell BB^T X_\ell - X_\ell BB^T X_{\ell+1} - X_{\ell+1} BB^T X_\ell$
$= K_{\ell+1}^T K_{\ell+1} + K_\ell^T K_\ell - K_{\ell+1}^T K_\ell - K_\ell^T K_{\ell+1}$

$\|R_\ell\|_2$ is stopping criterion in the LR CF-ADI.
$\|\frac{1}{2} N_\ell BB^T N_\ell\|_2$ can be approximated via eigensolver due to symmetry.
Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

New Question

How can we exploit $\mathcal{R}(X_{\ell+1}) = \mathbf{R}_\ell + \frac{1}{2} N_\ell \mathbf{B}\mathbf{B}^T N_\ell$ to control the ADI accuracy?

Riccati residual  inner Lyapunov residual

$N_\ell \mathbf{B}\mathbf{B}^T N_\ell = (X_{\ell+1} - X_{\ell}) \mathbf{B}\mathbf{B}^T (X_{\ell+1} - X_{\ell})$

$= X_{\ell+1} \mathbf{B}\mathbf{B}^T X_{\ell+1} + X_{\ell} \mathbf{B}\mathbf{B}^T X_{\ell} - X_{\ell} \mathbf{B}\mathbf{B}^T X_{\ell+1} - X_{\ell+1} \mathbf{B}\mathbf{B}^T X_{\ell}$

$= K_{\ell+1}^T K_{\ell+1} + K_{\ell}^T K_{\ell} - K_{\ell+1}^T K_{\ell} - K_{\ell}^T K_{\ell+1}$

- $\| \mathbf{R}_\ell \|_2$ is stopping criterion in the LRCF-ADI.
- $\| \frac{1}{2} N_\ell \mathbf{B}\mathbf{B}^T N_\ell \|_2$ can be approximated via eigensolver due to symmetry.
Inexact LRCF-NM For the ARE

Implementation

\( K_{\ell+1} \) can be accumulated during LRCF-ADI

Recall \( Z_{i+1} = [Z_i V_i] \) in LRCF-ADI.

\[
\Rightarrow K_{\ell+1}^{(i+1)} = B^T Z_{i+1} Z_{i+1}^T = B^T Z_i Z_i^T + B^T V_i V_i^T = K_{\ell+1}^{(i)} + B^T V_i V_i^T
\]
Inexact LRCF-NM For the ARE

Implementation

\[ K_{\ell+1} \text{ can be accumulated during LRCF-ADI} \]

Recall \( Z_{i+1} = [Z_i \ V_i] \) in LRCF-ADI.

\[ \Rightarrow K_{\ell+1}^{(i+1)} = B^T Z_{i+1} Z_{i+1}^T E = B^T Z_i Z_i^T E + B^T V_i V_i^T E = K_{\ell+1}^{(i)} + B^T V_i V_i^T E \]
Inexact LRCF-NM For the ARE

Implementation

\( K_{\ell+1} \) can be accumulated during LRCF-ADI

Recall \( Z_{i+1} = [Z_i V_i] \) in LRCF-ADI.

\[
K^{(i+1)}_{\ell+1} = B^T Z_{i+1} Z_{i+1}^T E = B^T Z_i Z_i^T E + B^T V_i V_i^T E = K^{(i)}_{\ell+1} + B^T V_i V_i^T E
\]

We can force quadratic convergence via

\[
\|R(X_{\ell+1})\|_2 \leq \|R_\ell\|_2 + \frac{1}{2} \|K_{\ell+1}^T K_{\ell+1} + K_{\ell}^T K_{\ell} - K_{\ell+1}^T K_{\ell} - K_{\ell}^T K_{\ell+1}\|_2 \\
\leq \varepsilon_\ell := \alpha R(X_\ell)^2
\]
Tests and Conclusions

Test Examples and Hardware

Rail 1357
- Oberwolfach MOR collection: Rail model
- MIMO (7 inputs, 6 outputs)
- n=1357

Rail 5177
- Oberwolfach MOR collection: Rail model
- MIMO (7 inputs, 6 outputs)
- n=5177

CPU type: Intel® Xeon® X5650 @ 2.67GHz
#CPUs: 2 #Cores: 12 (6 each)
RAM: 48 GB
## Tests and Conclusions

### Rail 1357

### I-LRCF-NM-S

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
<th>term.flag</th>
<th>inex.tol ($\varepsilon_\ell$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.208639 e-01</td>
<td>1</td>
<td>R</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.045024 e-01</td>
<td>4</td>
<td>R</td>
<td>3.854720 e-01</td>
</tr>
<tr>
<td>3</td>
<td>3.782191 e-03</td>
<td>8</td>
<td>R</td>
<td>1.092076 e-02</td>
</tr>
<tr>
<td>4</td>
<td>1.106234 e-05</td>
<td>20</td>
<td>R</td>
<td>1.430497 e-05</td>
</tr>
<tr>
<td>5</td>
<td>9.728725 e-11</td>
<td>43</td>
<td>L</td>
<td>1.223754 e-10</td>
</tr>
</tbody>
</table>

**time:** 2.73s
### Tests and Conclusions

**Rail 1357**

<table>
<thead>
<tr>
<th>I-LRCF-NM-S</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>step</td>
<td>rel.residual</td>
<td># ADI</td>
<td>term.flag</td>
<td>inex.tol ($\varepsilon_\ell$)</td>
<td>time: <strong>2.73s</strong></td>
</tr>
<tr>
<td>1</td>
<td>6.208639 e-01</td>
<td>1</td>
<td>R</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.045024 e-01</td>
<td>4</td>
<td>R</td>
<td>3.854720 e-01</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3.782191 e-03</td>
<td>8</td>
<td>R</td>
<td>1.092076 e-02</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.106234 e-05</td>
<td>20</td>
<td>R</td>
<td>1.430497 e-05</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9.728725 e-11</td>
<td>43</td>
<td>L</td>
<td>1.223754 e-10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LRCF-NM</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>time: <strong>4.60s</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>step</td>
<td>rel.residual</td>
<td># ADI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6.739966 e-05</td>
<td>44</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9.423553 e-09</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7.831133 e-11</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Tests and Conclusions

**Rail 1357**

### I-LRCF-NM-S

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
<th>term.flag</th>
<th>inex.tol (εℓ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.208639e-01</td>
<td>1</td>
<td>R</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.045024e-01</td>
<td>4</td>
<td>R</td>
<td>3.854720e-01</td>
</tr>
<tr>
<td>3</td>
<td>3.782191e-03</td>
<td>8</td>
<td>R</td>
<td>1.092076e-02</td>
</tr>
<tr>
<td>4</td>
<td>1.106234e-05</td>
<td>20</td>
<td>R</td>
<td>1.430497e-05</td>
</tr>
<tr>
<td>5</td>
<td>9.728725e-11</td>
<td>43</td>
<td>L</td>
<td>1.223754e-10</td>
</tr>
</tbody>
</table>

*Time:* 2.73s

### LRCF-NM

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.739966e-05</td>
<td>44</td>
</tr>
<tr>
<td>2</td>
<td>9.423553e-09</td>
<td>45</td>
</tr>
<tr>
<td>3</td>
<td>7.831133e-11</td>
<td>45</td>
</tr>
</tbody>
</table>

*Time:* 4.60s

### LRCF-NM-GP (projection in every outer and inner step)

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.361686e-11</td>
<td>31</td>
</tr>
</tbody>
</table>

*Time:* 3.78s
# Tests and Conclusions

## Rail 1357

### I-LRCF-NM-S

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
<th>term.flag</th>
<th>inex.tol (ε_{ℓ})</th>
<th>time: 2.73s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.208639 e-01</td>
<td>1</td>
<td>R</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.045024 e-01</td>
<td>4</td>
<td>R</td>
<td>3.854720 e-01</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3.782191 e-03</td>
<td>8</td>
<td>R</td>
<td>1.092076 e-02</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.106234 e-05</td>
<td>20</td>
<td>R</td>
<td>1.430497 e-05</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9.728725 e-11</td>
<td>43</td>
<td>L</td>
<td>1.223754 e-10</td>
<td></td>
</tr>
</tbody>
</table>

### LRCF-NM

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
<th>time: 4.60s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.739966 e-05</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9.423553 e-09</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7.831133 e-11</td>
<td>45</td>
<td></td>
</tr>
</tbody>
</table>

### LRCF-NM-GP (projection in every outer step)

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
<th>time: 1.47s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.272812 e-12</td>
<td>44</td>
<td></td>
</tr>
</tbody>
</table>

---

Max Planck Institute Magdeburg

Jens Saak, *Inexact Newton for Large AREs*
## Tests and Conclusions

### Rail 5177

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
<th>term.flag</th>
<th>inex.tol ($\varepsilon_{\ell}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.674262e-01</td>
<td>1</td>
<td>R</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.733687e-01</td>
<td>3</td>
<td>R</td>
<td>4.454578e-01</td>
</tr>
<tr>
<td>3</td>
<td>9.853188e-03</td>
<td>7</td>
<td>R</td>
<td>3.005672e-02</td>
</tr>
<tr>
<td>4</td>
<td>8.072432e-05</td>
<td>19</td>
<td>R</td>
<td>9.708532e-05</td>
</tr>
<tr>
<td>5</td>
<td>2.761797e-09</td>
<td>41</td>
<td>R</td>
<td>6.516416e-09</td>
</tr>
<tr>
<td>6</td>
<td>4.808556e-11</td>
<td>53</td>
<td>L</td>
<td>7.627521e-18</td>
</tr>
</tbody>
</table>

**time:** 18.79s
### Tests and Conclusions

#### Rail 5177

#### I-LRCF-NM-S

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
<th>term.flag</th>
<th>inex.tol ($\varepsilon_{\ell}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.674262 e-01</td>
<td>1</td>
<td>R</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.733687 e-01</td>
<td>3</td>
<td>R</td>
<td>4.454578 e-01</td>
</tr>
<tr>
<td>3</td>
<td>9.853188 e-03</td>
<td>7</td>
<td>R</td>
<td>3.005672 e-02</td>
</tr>
<tr>
<td>4</td>
<td>8.072432 e-05</td>
<td>19</td>
<td>R</td>
<td>9.708532 e-05</td>
</tr>
<tr>
<td>5</td>
<td>2.761797 e-09</td>
<td>41</td>
<td>R</td>
<td>6.516416 e-09</td>
</tr>
<tr>
<td>6</td>
<td>4.808556 e-11</td>
<td>53</td>
<td>L</td>
<td>7.627521 e-18</td>
</tr>
</tbody>
</table>

**time:** 18.79s

#### LRCF-NM

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.691724 e-05</td>
<td>54</td>
</tr>
<tr>
<td>2</td>
<td>2.321039 e-09</td>
<td>54</td>
</tr>
<tr>
<td>3</td>
<td>8.044616 e-11</td>
<td>54</td>
</tr>
</tbody>
</table>

**time:** 23.49s
## Tests and Conclusions

**Rail 5177**

### I-LRCF-NM-S

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
<th>term.flag</th>
<th>inex.tol ($\varepsilon_\ell$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.674262e-01</td>
<td>1</td>
<td>R</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.733687e-01</td>
<td>3</td>
<td>R</td>
<td>4.454578e-01</td>
</tr>
<tr>
<td>3</td>
<td>9.853188e-03</td>
<td>7</td>
<td>R</td>
<td>3.005672e-02</td>
</tr>
<tr>
<td>4</td>
<td>8.072432e-05</td>
<td>19</td>
<td>R</td>
<td>9.708532e-05</td>
</tr>
<tr>
<td>5</td>
<td>2.761797e-09</td>
<td>41</td>
<td>R</td>
<td>6.516416e-09</td>
</tr>
<tr>
<td>6</td>
<td>4.808556e-11</td>
<td>53</td>
<td>L</td>
<td>7.627521e-18</td>
</tr>
</tbody>
</table>

**time:** 18.79s

### LRCF-NM

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.691724e-05</td>
<td>54</td>
</tr>
<tr>
<td>2</td>
<td>2.321039e-09</td>
<td>54</td>
</tr>
<tr>
<td>3</td>
<td>8.044616e-11</td>
<td>54</td>
</tr>
</tbody>
</table>

**time:** 23.49s

### LRCF-NM-GP (projection in every outer and inner step)

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.762903e-11</td>
<td>38</td>
</tr>
</tbody>
</table>

**time:** 11.62s
## Tests and Conclusions

### Rail 5177

**I-LRCF-NM-S**

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
<th>term.flag</th>
<th>inex.tol ($\varepsilon_I$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.674262 e-01</td>
<td>1</td>
<td>R</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.733687 e-01</td>
<td>3</td>
<td>R</td>
<td>4.454578 e-01</td>
</tr>
<tr>
<td>3</td>
<td>9.853188 e-03</td>
<td>7</td>
<td>R</td>
<td>3.005672 e-02</td>
</tr>
<tr>
<td>4</td>
<td>8.072432 e-05</td>
<td>19</td>
<td>R</td>
<td>9.708532 e-05</td>
</tr>
<tr>
<td>5</td>
<td>2.761797 e-09</td>
<td>41</td>
<td>R</td>
<td>6.516416 e-09</td>
</tr>
<tr>
<td>6</td>
<td>4.808556 e-11</td>
<td>53</td>
<td>L</td>
<td>7.627521 e-18</td>
</tr>
</tbody>
</table>

**LRCF-NM**

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.691724 e-05</td>
<td>54</td>
</tr>
<tr>
<td>2</td>
<td>2.321039 e-09</td>
<td>54</td>
</tr>
<tr>
<td>3</td>
<td>8.044616 e-11</td>
<td>54</td>
</tr>
</tbody>
</table>

**LRCF-NM-GP** *(projection in every outer step)*

<table>
<thead>
<tr>
<th>step</th>
<th>rel.residual</th>
<th># ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.204343 e-12</td>
<td>54</td>
</tr>
</tbody>
</table>

**Time:**
- I-LRCF-NM-S: 18.79s
- LRCF-NM: 23.49s
- LRCF-NM-GP: 5.17s
Conclusions and Future Perspectives

Conclusions

- Inexact Newton can be applied in the large scale case.
- Number of ADI steps taken can be reduced drastically.
- Newton Galerkin approach still faster by far.

Future Work

- Combine inexact and projected Newton methods,
- Check other inner loop solvers, e.g., [Simoncini ’07-’11, Vandereycken ’10],
- Integrate these approaches in DRE solvers
Conclusions and Future Perspectives

Conclusions

- Inexact Newton can be applied in the large scale case.
- Number of ADI steps taken can be reduced drastically.
- Newton Galerkin approach still faster by far.

Future Work

- Combine inexact and projected Newton methods,
- Check other inner loop solvers, e.g., [Simoncini ’07–’11, Vandereycken ’10],
- Integrate these approaches in DRE solvers

Many thanks for your attention.
An LRCF-QADI Projection Method

Newton-Kleinman-ADI vs. QADI

**QADI**

\[
\begin{align*}
(A^T - K_{j-1}^T B^T + p_j I)X_{j-\frac{1}{2}} &= -Q - X_{j-1}^T(A - p_j I), \\
(A^T - K_{j-\frac{1}{2}}^T B^T + p_j I)X_j &= -Q - X_{j-\frac{1}{2}}(A - p_j I).
\end{align*}
\]

[Wong/Balakrishnan '04-'07]
An LRCF-QADI Projection Method
Newton-Kleinman-ADI vs. QADI

QADI

\[ (A^T - K_{j-1}^T B^T + p_j l) X_{j-\frac{1}{2}} = -Q - X_{j-1}^T (A - p_j l), \]
\[ (A^T - K_{j-\frac{1}{2}}^T B^T + p_j l) X_j = -Q - X_{j-\frac{1}{2}} (A - p_j l). \]

Newton-Kleinman-ADI

\[ (A^T - K_{j-1}^T B^T + p_k l) X_{k-\frac{1}{2}} = -Q - K_{j-1}^T K_{j-1} - X_{k-1}^T (A - BK_{j-1} - p_k l), \]
\[ (A^T - K_{j-1}^T B^T + p_k l) X_k = -Q - K_{j-1}^T K_{j-1} - X_{k-\frac{1}{2}} (A - BK_{j-1} - p_k l). \]
# An LRCF-QADI Projection Method

**Newton-Kleinman-ADI vs. QADI**

### QADI

[Wong/Balakrishnan ’04-'07]

\[
(A^T - K_{j-1}^T B^T + p_j I)X_{j-\frac{1}{2}} = -Q - K_{j-1}^T K_{j-1} - X_{j-1}^T (A - BK_{j-1} - p_j I),
\]

\[
(A^T - K_{j-\frac{1}{2}}^T B^T + p_j I)X_j = -Q - K_{j-\frac{1}{2}}^T K_{j-\frac{1}{2}} - X_{j-\frac{1}{2}} (A - BK_{j-\frac{1}{2}} - p_j I).
\]

### Newton-Kleinman-ADI

[Kleinman ’68]

\[
(A^T - K_{j-1}^T B^T + p_k I)X_{k-\frac{1}{2}} = -Q - K_{j-1}^T K_{j-1} - X_{k-1}^T (A - BK_{j-1} - p_k I),
\]

\[
(A^T - K_{j-1}^T B^T + p_k I)X_k = -Q - K_{j-1}^T K_{j-1} - X_{k-\frac{1}{2}} (A - BK_{j-1} - p_k I).
\]
An LRCF-QADI Projection Method

Newton-Kleinman-ADI vs. QADI

QADI

\[ (A^T - K_{j-1}^T B^T + p_j l)X_{j-\frac{1}{2}} = -Q - K_{j-1}^T K_{j-1} - X_{j-1}^T (A - BK_{j-1} - p_j l), \]

\[ (A^T - K_{j-\frac{1}{2}}^T B^T + p_j l)X_j = -Q - K_{j-\frac{1}{2}}^T K_{j-\frac{1}{2}} - X_{j-\frac{1}{2}} (A - BK_{j-\frac{1}{2}} - p_j l). \]

Newton-Kleinman-ADI

\[ (A^T - K_{j-1}^T B^T + p_k l)X_{k-\frac{1}{2}} = -Q - K_{j-1}^T K_{j-1} - X_{k-1}^T (A - BK_{j-1} - p_k l), \]

\[ (A^T - K_{j-1}^T B^T + p_k l)X_k = -Q - K_{j-1}^T K_{j-1} - X_{k-\frac{1}{2}} (A - BK_{j-1} - p_k l). \]
An LRCF-QADI Projection Method
Newton-Kleinman-ADI vs. QADI

**QADI**

\[
(A^T - K_{j-1}^T B^T + p_j l)X_{j-\frac{1}{2}} = -Q - K_{j-1}^T K_{j-1} - X_{j-1}^T (A - BK_{j-1} - p_j l),
\]
\[
(A^T - K_{j-\frac{1}{2}}^T B^T + p_j l)X_j = -Q - K_{j-\frac{1}{2}}^T K_{j-\frac{1}{2}} - X_{j-\frac{1}{2}}^T (A - BK_{j-\frac{1}{2}} - p_j l).
\]

**Newton-Kleinman-ADI**

\[
(A^T - K_{j-1}^T B^T + p_k l)X_{k-\frac{1}{2}} = -Q - K_{j-1}^T K_{j-1} - X_{k-1}^T (A - BK_{j-1} - p_k l),
\]
\[
(A^T - K_{j-\frac{1}{2}}^T B^T + p_k l)X_k = -Q - K_{j-\frac{1}{2}}^T K_{j-\frac{1}{2}} - X_{k-\frac{1}{2}}^T (A - BK_{j-\frac{1}{2}} - p_k l).
\]
An LRCF-QADI Projection Method

The LRCF-QADI Iteration

Idea

[Benner/S. ’09]

Apply the Gauß-Seidel-like idea in the (LRCF-NM), i.e.,

- do not distinguish between inner and outer loops
- update \( K \) and thus also \( F \) in every ADI step.

Shift Parameters?

1. Use shifts for initial closed loop matrix \( A - BK^{(0)} \)
2. Compute shifts with respect to stable eigenvalues of

\[
H := \begin{bmatrix}
A & BB^T \\
C^T C & -A^T
\end{bmatrix},
\]

i.e., the eigenvalues of the desired closed loop matrix using

[Effenberger ’09]
An LRCF-QADI Projection Method

The LRCF-QADI Iteration

Algorithm 1 Low-Rank Cholesky Factor Newton Method (LRCF-NM)

Input: \( A, B, C, K^{(0)} \) for which \( A - BK^{(0)}\) is stable
Output: \( Z = Z^{(k_{\text{max}})} \), such that \( ZZ^H \) approximates the solution \( X \) of
\[
C^T C + A^T X + XA - XBB^T X = 0.
\]

1: for \( k = 1, 2, \ldots, k_{\text{max}} \) do
2: Determine (sub)optimal ADI shift parameters \( p_1^{(k)} , p_2^{(k)} , \ldots \)
   with respect to the matrix \( F^{(k)} = A^T - K^{(k-1)} B^T \).
3: \( G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix} \)
4: Compute \( Z^{(k)} \) using (LRCF-ADI) or (LRCF-ADI-GP) such that
   \( F^{(k)} Z^{(k)} Z^{(k)H} + Z^{(k)} Z^{(k)H} F^{(k)} = -G^{(k)} G^{(k)T} \).
5: \( K^{(k)} = Z^{(k)} (Z^{(k)H} B) \)
6: end for
Algorithm 9 Low-Rank Cholesky Factor QADI (LRCF-QADI)

Input: $A, B, C, K^{(0)}$ for which $A - BK^{(0)^T}$ is stable
Output: $Z = Z^{(k_{max})}$, such that $ZZ^H$ approximates the solution $X$ of

$$
C^T C + A^T X + XA - XBB^T X = 0.
$$

1: Determine QADI shift parameters $p_1, p_2, \ldots$
2: $G = \begin{bmatrix} C^T & K^{(0)} \end{bmatrix}$
3: for $k = 1, 2, \ldots, k_{max}$ do
4: $F^{(k)} = A^T - K^{(k-1)^T} B^T$
5: For $\tilde{V}$ solve $(F^{(k)} + p_k I)\tilde{V} = V_{k-1}$
6: $V_k = \sqrt{\text{Re}(p_k) / \text{Re}(p_{k-1})} \left( V_{k-1} - (p_k + \bar{p}_{k-1})\tilde{V} \right)$
7: $Z_k = \begin{bmatrix} Z_{k-1} & V_k \end{bmatrix}$
8: $K^{(k)} = K^{(k-1)} + V_k V_k^T B$
9: end for
An LRCF-QADI Projection Method

The LRCF-QADI Iteration

**Algorithm 9** Low-Rank Cholesky Factor QADI (LRCF-QADI)

**Input:** $A$, $B$, $C$, $K^{(0)}$ for which $A - BK^{(0)}{^T}$ is stable

**Output:** $Z = Z^{(k_{max})}$, such that $ZZ^H$ approximates the solution $X$ of

$$
C^T C + A^T X + XA - XBB^T X = 0.
$$

1: Determine QADI shift parameters $p_1, p_2, \ldots$
2: $G = \begin{bmatrix} C^T & K^{(0)} \end{bmatrix}$
3: for $k = 1, 2, \ldots, k_{max}$ do
4: $F^{(k)} = A^T - K^{(k-1)} B^T$
5: For $\tilde{V}$ solve $(F^{(k)} + p_k I)\tilde{V} = V_{k-1}$
6: $V_k = \sqrt{\frac{\text{Re}(p_k)}{\text{Re}(p_{k-1})}} \left( V_{k-1} - (p_k + p_{k-1})\tilde{V} \right)$
7: $Z_k = [Z_{k-1} \quad V_k]$
8: Project ARE, solve and prolongate solution
9: $K^{(k)} = Z^{(k)}(Z^{(k)}^H B)$
10: end for
The RicADI Projection Method

The RicADI Idea

Observation

LRCF-NM-GP often converges after only one Newton step.

“Is this the holy grail?”
## The RicADI Projection Method

### The RicADI Idea

<table>
<thead>
<tr>
<th>Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRCF-NM-GP often converges after only one Newton step.</td>
</tr>
</tbody>
</table>

“Is this the holy grail?”

<table>
<thead>
<tr>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can we avoid updating $F$ in the Newton based solvers completely?</td>
</tr>
</tbody>
</table>
# The RicADI Projection Method

## Invariance of the Krylov Subspaces

### Factored Newton-Kleinman Iteration

[Benner/Li/Penzl '99/'08]

\[
F_\ell = A - BB^TX_\ell E =: A - BK_\ell
\]

\[
G_\ell = [C^T \ K_\ell^T]
\]

is “sparse + low rank”

is low rank factor

Need to solve:

\[
F_\ell^TX_{\ell+1} + X_{\ell+1}F_\ell = -G_\ell G_\ell^T.
\]

### Feedback Invariance of Subspaces

[Benner/Beckermann '11]

\[
\mathcal{K}_m(F_\ell^T, G_\ell) = \mathcal{K}_m(A^T, G_\ell)
\]
The RicADI Projection Method

Invariance of the Krylov Subspaces

Factored Newton-Kleinman Iteration

\[ F_\ell = A - B B^T X_\ell E =: A - B K_\ell \]
\[ G_\ell = [C^T \quad K_\ell^T] \]

Need to solve:
\[ F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T. \]

Feedback Invariance of Subspaces

\[ \mathcal{K}_m(F_\ell^T, G_\ell) = \mathcal{K}_m(A^T, G_\ell) \]

\[ \mathcal{K}_m(F_\ell^T, G_\ell) = \mathcal{K}_m((A - B K_\ell)^T, [C^T \quad K_\ell^T]) \]
\[ = \text{sp}[C^T, K_\ell^T, A^T C^T - K_\ell^T B^T C^T, A^T K_\ell^T - K_\ell^T B^T K_\ell^T, \ldots] \]
\[ = \text{sp}[C^T, K_\ell^T, A^T C^T, A^T K_\ell^T, \ldots] \]
\[ = \mathcal{K}_m(A^T, [C^T K_\ell]) = \mathcal{K}_m(A^T, G_\ell) \]
The RicADI Projection Method

Invariance of the Krylov Subspaces

**Factored Newton-Kleinman Iteration**

\[
F_{\ell} = A - BB^TX_{\ell}E =: A - BK_{\ell}
\]

\[
G_{\ell} = [C^T K_{\ell}^T]
\]

Need to solve:

\[
F_{\ell}^TX_{\ell+1} + X_{\ell+1}F_{\ell} = -G_{\ell}G_{\ell}^T.
\]

is “sparse + low rank”

is low rank factor

**Feedback Invariance of Subspaces**

**Benner/Beckermann '11**

\[
\mathcal{K}_m(F_{\ell}^T, G_{\ell}) = \mathcal{K}_m((A - BK_{\ell})^T, [C^T K_{\ell}^T])
\]

\[
= \text{sp}[C^T, K_{\ell}^T, A^T C^T - K_{\ell}^T B^T C^T, A^T K_{\ell}^T - K_{\ell}^T B^T K_{\ell}^T, \ldots]
\]

\[
= \text{sp}[C^T, K_{\ell}^T, A^T C^T, A^T K_{\ell}^T, \ldots]
\]

\[
= \mathcal{K}_m(A^T, [C^T K_{\ell}]) = \mathcal{K}_m(A^T, G_{\ell})
\]
The RicADI Projection Method

RicADI Algorithm

Algorithm 10  Low-rank Cholesky factor RicADI iteration  
(LRCF-RicADI)  
[Benner/Köhler/S. ’11]

Input: $A$, $B$, $C$, $K^{(0)}$ for which $F = A - BK^{(0)}^T$ is stable.

Output: $Z = Z_{i_{\text{max}}} \in \mathbb{C}^{n \times t_{i_{\text{max}}}}$, such that $ZZ^H \approx X$ approximates the solution $X$ of

$C^T C + A^T X + X A - X BB^T X = 0$.

1: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \ldots$

with respect to the matrix $F$

2: For $V_1$ solve $(F + p_1 I) V_1 = \sqrt{-2 \Re(p_1)} G$

3: $Z_1 = V_1$, $i = 2$

4: repeat

5: For $\tilde{V}$ solve $(F + p_i I) \tilde{V} = V_{i-1}$

6: $V_i = \sqrt{\Re(p_i) / \Re(p_{i-1})} \left( V_{i-1} - (p_i + \bar{p}_{i-1}) \tilde{V} \right)$

7: $Z_i = [Z_{i-1} \ V_i]$, $i = i + 1$

8: Project ARE, solve and prolongate solution

9: until $\| C^T C + A^T Z_i Z_i^T + Z_i Z_i^T A - Z_i Z_i^T B B^T Z_i Z_i^T \| \leq TOL$
The RicADI Projection Method

RicADI Algorithm

Algorithm 10 Low-rank Cholesky factor RicADI iteration
(LRCF-RicADI)

Input: \( A, B, C, K^{(0)} \) for which \( F = A - BK^{(0)} \) is stable.
Output: \( Z = Z_{i_{\text{max}}} \in \mathbb{C}^{n \times t_{i_{\text{max}}}} \), such that \( ZZ^H \approx X \) approximates the solution \( X \) of
\[
C^T C + A^T X + X A - XBB^T X = 0.
\]

1: Determine (sub)optimal ADI shift parameters \( p_1, p_2, \ldots \)
2: For \( V_1 \) solve \((F + p_1 I)V_1 = \sqrt{-2 \text{Re}(p_1)} G\)
3: \( Z_1 = V_1, \quad i = 2 \)
4: repeat
5: For \( \tilde{V} \) solve \((F + p_i I)\tilde{V} = \sqrt{\text{Re}(p_i)} \tilde{V}\)
6: \( V_i = \sqrt{\text{Re}(p_i)/\text{Re}(p_{i-1})} \left(V_{i-1} - (p_i + p_{i-1}) \tilde{V}\right)\)
7: \( Z_i = [Z_{i-1} \quad V_i], \quad i = i + 1 \)
8: Project ARE, solve and prolongate solution
9: until \( \|C^T C + A^T Z_i Z_i^T + Z_i Z_i^T A - Z_i Z_i^T B B^T Z_i Z_i^T\| \leq \text{TOL} \)

Compute projection basis via \( Q_i = Z_i L_i \), where
\[
L_i = U(:, 1 : k) \sqrt{\Sigma(1 : k, 1 : k)^{-1}}
\]
and
\[
[U, \Sigma, V] = \text{svd}(Z^T Z)
\]
to avoid problems with (almost) rank deficient \( Z_i \) and still avoid frequent orthogonalization.
The RicADI Projection Method

RicADI Algorithm

Algorithm 10 General. Low-rank Cholesky factor RicADI iteration

\[(\text{G-LRCF-RicADI})\]

\[\text{Input: } E, A, B, C, K^{(0)} \text{ for which } (F = A - BK^{(0)}^T, E) \text{ is stable.}\]

\[\text{Output: } Z = Z_{\text{max}} \in \mathbb{C}^{n \times t_{\text{max}}}, \text{ such that } ZZ^H \approx X \text{ approximates the solution } X \text{ of}\]

\[C^T C + A^T X E + E^T X A - E^T X B B^T X E = 0.\]

1: Determine (sub)optimal ADI shift parameters \(p_1^{(k)}, p_2^{(k)}, \ldots\)
   with respect to the matrix \(F\)
2: For \(V_1\) solve \((F + p_1 E) V_1 = \sqrt{-2 \text{Re}(p_1)} G\)
3: \(Z_1 = V_1, \quad i = 2\)
4: repeat
   5: For \(\tilde{V}\) solve \((F + p_i E) \tilde{V} = E V_{i-1}\)
   6: \(V_i = \sqrt{\text{Re}(p_i)/\text{Re}(p_{i-1})} \left( V_{i-1} - (p_i + \bar{p}_{i-1}) \tilde{V} \right)\)
   7: \(Z_i = [Z_{i-1} \quad V_i], i = i + 1\)
8: Project ARE, solve and prolongate solution
9: until \(\|C^T C + A^T Z_i Z_i^T E + E^T Z_i Z_i^T A - E^T Z_i Z_i^T B B^T Z_i Z_i^T E\| \leq \text{TOL}\)
Numerical Results

Test Examples and Hardware

FDM 40k
- 2d heat equation on unit square
- 200 grid points per direction
- 5 point difference stars
- SISO

FDM 250k
- 2d heat equation on unit square
- 500 grid points per direction
- 5 point difference stars
- SISO

Rail 79k
- Oberwolfach MOR collection: Rail model
- here $B = 100B$, i.e., weight $10^4$ on control term in cost functional
- MIMO (7 inputs, 6 outputs)

- CPU type: Intel® Xeon® X5650 @ 2.67GHz
- #CPUs: 2 #Cores: 12 (6 each)
- RAM: 48 GB
## Numerical Results

### MATLAB

<table>
<thead>
<tr>
<th></th>
<th>LRCF-NM</th>
<th>LRCF-NM-S</th>
<th>LRCF-NM-GP</th>
<th>RicADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDM 40k</td>
<td>135.499 s</td>
<td>83.953 s</td>
<td>13.629 s</td>
<td>11.760 s</td>
</tr>
<tr>
<td>FDM 250k</td>
<td>1912.790 s</td>
<td>1168.900 s</td>
<td>139.102 s</td>
<td>121.616 s</td>
</tr>
<tr>
<td>Rail 79k</td>
<td>1775.280 s</td>
<td>1784.001 s</td>
<td>131.280 s</td>
<td>80.854 s</td>
</tr>
</tbody>
</table>

**Table:** Computation time in seconds on otto using MATLAB 2010b
### Numerical Results

**MATLAB**

<table>
<thead>
<tr>
<th></th>
<th>LRCF-NM</th>
<th>LRCF-NM-S</th>
<th>LRCF-NM-GP</th>
<th>RicADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDM 40k</td>
<td>135.499 s</td>
<td>83.953 s</td>
<td>13.629 s</td>
<td>11.760 s</td>
</tr>
<tr>
<td>FDM 250k</td>
<td>1,912.790 s</td>
<td>1,168.900 s</td>
<td>139.102 s</td>
<td>121.616 s</td>
</tr>
<tr>
<td>Rail 79k</td>
<td>1,775.280 s</td>
<td>1,784.001 s</td>
<td>131.280 s</td>
<td>80.854 s</td>
</tr>
</tbody>
</table>

**Table:** Computation time in seconds on `otto` using MATLAB 2010b

<table>
<thead>
<tr>
<th></th>
<th>LRCF-NM</th>
<th>LRCF-NM-S</th>
<th>LRCF-NM-GP</th>
<th>RicADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDM 40k</td>
<td>6.619 e-10</td>
<td>6.360 e-10</td>
<td>2.611 e-12</td>
<td>3.027 e-10</td>
</tr>
<tr>
<td>FDM 250k</td>
<td>1.954 e-10</td>
<td>1.038 e-10</td>
<td>4.324 e-12</td>
<td>6.915 e-11</td>
</tr>
<tr>
<td>Rail 79k</td>
<td>4.709 e-10</td>
<td>4.710 e-10</td>
<td>3.004 e-10</td>
<td>2.136 e-09</td>
</tr>
</tbody>
</table>

**Table:** Final residuals on `otto` using MATLAB 2010b
Numerical Results

<table>
<thead>
<tr>
<th></th>
<th>LRCF-NM</th>
<th>LRCF-NM-S</th>
<th>LRCF-NM-GP</th>
<th>RicADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDM 40k</td>
<td>44.84 s</td>
<td>37.664 s</td>
<td>4.160 s</td>
<td>3.440 s</td>
</tr>
<tr>
<td>FDM 250k</td>
<td>1568.94 s</td>
<td>462.653 s</td>
<td>135.854 s</td>
<td>127.716 s</td>
</tr>
<tr>
<td>Rail 79k</td>
<td>1306.99 s</td>
<td>1468.850 s</td>
<td>56.379 s</td>
<td>22.956 s</td>
</tr>
</tbody>
</table>

Table: Computation time in seconds on Otto using C
## Numerical Results

<table>
<thead>
<tr>
<th>Method</th>
<th>LRCF-NM</th>
<th>LRCF-NM-S</th>
<th>LRCF-NM-GP</th>
<th>RicADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDM 40k</td>
<td>44.84 s</td>
<td>37.664 s</td>
<td>4.160 s</td>
<td>3.440 s</td>
</tr>
<tr>
<td>FDM 250k</td>
<td>1568.94 s</td>
<td>462.653 s</td>
<td>135.854 s</td>
<td>127.716 s</td>
</tr>
<tr>
<td>Rail 79k</td>
<td>1306.99 s</td>
<td>1468.850 s</td>
<td>56.379 s</td>
<td>22.956 s</td>
</tr>
</tbody>
</table>

**Table**: Computation time in seconds on otto using C

<table>
<thead>
<tr>
<th>Method</th>
<th>LRCF-NM</th>
<th>LRCF-NM-S</th>
<th>LRCF-NM-GP</th>
<th>RicADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDM 40k</td>
<td>6.619e-10</td>
<td>6.350e-10</td>
<td>5.901e-11</td>
<td>3.001e-10</td>
</tr>
<tr>
<td>FDM 250k</td>
<td>1.960e-11</td>
<td>5.318e-11</td>
<td>7.912e-12</td>
<td>8.163e-10</td>
</tr>
<tr>
<td>Rail 79k</td>
<td>2.019e-10</td>
<td>2.019e-10</td>
<td>2.240e-11</td>
<td>9.528e-10</td>
</tr>
</tbody>
</table>

**Table**: Final residuals on otto using C