Workshop on Matrix Equations
and Tensor Techniques
November 21-22, 2011

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

Jens Saak[1] and Timo Hylla[2]

[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

Preliminaries and Notation

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

$$\min_{u\in\mathcal{U}_{ad}} \mathcal{J}(u) = \int_0^\infty x^T C^T C x + u^T u \, dt$$

constrained to the linear time invariant system

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

Then the optimal state feedback control is given by

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

where X_∞ 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 $\operatorname{rank}(E) = n$.
Consider the linear quadratic regulator problem

$$\min_{u \in \mathcal{U}_{ad}} \mathcal{J}(u) = \int_0^\infty x^T C^T C x + u^T u \, dt$$

constrained to the linear time invariant system

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t). \end{aligned}$$

Then the optimal state feedback control is given by

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

where X_∞ 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 $\Lambda(A) \subset \mathbb{C}_{<0}$.

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.

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.

Preliminaries and Notation

Definition

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

In case $E \neq I$:

$A \rightsquigarrow E^{-1}A$ and $B \rightsquigarrow E^{-1}B$

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.

Outline

Iterative Solution of the ARE

Iterative Solution of the ARE

Newton's Method for AREs

Consider $\mathfrak{R}(X) := C^T C + A^T X + XA - XBB^T X = 0$

Newton's Iteration for the ARE

$$\mathfrak{R}'|_{X_\ell}(N_\ell) = -\mathfrak{R}(X_\ell), \qquad X_{\ell+1} = X_\ell + N_\ell, \qquad \ell = 0, 1, \ldots$$

where the Frechét derivative of $\mathfrak{R}$ at X is the Lyapunov operator

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

i.e., in every Newton step solve a

Lyapunov Equation

$$(A - BB^T X_\ell)^T N_\ell + N_\ell (A - BB^T X_\ell) = -\mathfrak{R}(X_\ell).$$

Iterative Solution of the ARE

Newton's Method for AREs

Consider $\mathfrak{R}(X) := C^T C + A^T X + XA - XBB^T X = 0$

Kleinman's Iteration for the ARE [Kleinman '68]

$$\mathfrak{R}'|_{X_\ell}(X_{\ell+1}) = \mathfrak{R}'|_{X_\ell}(X_\ell) - \mathfrak{R}(X_\ell), \qquad \ell = 0, 1, \ldots$$

where the Frechét derivative of $\mathfrak{R}$ at X is the Lyapunov operator

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

i.e., in every Newton step solve a

Lyapunov Equation [Kleinman '68]

$$(A - BB^T X_\ell)^T X_{\ell+1} + X_{\ell+1}(A - BB^T X_\ell) = -C^T C - X_\ell BB^T X_\ell.$$

Iterative Solution of the ARE

Newton's Method for AREs

Consider $\qquad \mathfrak{R}(X) := C^T C + A^T X + XA - XBB^T X = 0$

Kleinman's Iteration for the ARE [Kleinman '68]

$$\mathfrak{R}'|_{X_\ell}(X_{\ell+1}) = \mathfrak{R}'|_{X_\ell}(X_\ell) - \mathfrak{R}(X_\ell), \qquad \ell = 0, 1, \ldots$$

where the Frechét derivative of $\mathfrak{R}$ at X is the Lyapunov operator

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

i.e., in every Newton step solve a

Lyapunov Equation [Kleinman '68]

$$F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T.$$

Iterative Solution of the ARE

Newton's Method for AREs

Consider $\mathfrak{R}(X) := C^T C + A^T X E + E^T XA - E^T XBB^T X E = 0$

Kleinman's Iteration for the ARE [Kleinman '68]

$$\mathfrak{R}'|_{X_\ell}(X_{\ell+1}) = \mathfrak{R}'|_{X_\ell}(X_\ell) - \mathfrak{R}(X_\ell), \qquad \ell = 0, 1, \ldots$$

where the Frechét derivative of $\mathfrak{R}$ at X is the Lyapunov operator

$$\mathfrak{R}'|_X : \quad Q \mapsto (A - BB^T X E)^T Q E + E^T Q (A - BB^T X E),$$

i.e., in every Newton step solve a

Lyapunov Equation [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_∞, such that

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

Furthermore the convergence is quadratic and monotone with

$$0 \leq X_\infty \leq \cdots \leq X_{k+1} \leq X_k \leq \cdots \leq X_1.$$

Iterative Solution of the ARE

Inexact Newton for AREs (Basic Concept) [Feitzinger/Hylla/Sachs '09, Hylla '10]

Consider $$\mathfrak{R}(X) := C^T C + A^T X + XA - XBB^T X = 0$$

Inexact Newton's Iteration for the ARE

$$\mathfrak{R}'|_{X_\ell}(N_\ell) + \mathfrak{R}(X_\ell) = R_\ell, \quad X_{\ell+1} = X_\ell + N_\ell, \quad \ell = 0, 1, \ldots$$

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

Lyapunov Equation

$$(A - BB^T X_\ell)^T N_\ell + N_\ell (A - BB^T X_\ell) = -\mathfrak{R}(X_\ell) + R_\ell.$$

Iterative Solution of the ARE

Inexact Newton for AREs (Basic Concept)

[FEITZINGER/HYLLA/SACHS '09, HYLLA '10]

Consider $\mathfrak{R}(X) := C^T C + A^T X + XA - XBB^T X = 0$

Inexact Kleinman's Iteration for the ARE

$$\mathfrak{R}'|_{X_\ell}(X_{\ell+1}) - \mathfrak{R}'|_{X_\ell}(X_\ell) + \mathfrak{R}(X_\ell) = R_\ell, \qquad \ell = 0, 1, \ldots$$

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

Lyapunov Equation

$$(A - BB^T X_\ell)^T X_{\ell+1} + X_{\ell+1}(A - BB^T X_\ell) = -C^T C - X_\ell BB^T X_\ell + R_\ell.$$

Iterative Solution of the ARE

Inexact Newton for AREs (Basic Concept)

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

Consider $\mathfrak{R}(X) := C^T C + A^T X + XA - XBB^T X = 0$

Inexact Kleinman's Iteration for the ARE

$$\mathfrak{R}'|_{X_\ell}(X_{\ell+1}) - \mathfrak{R}'|_{X_\ell}(X_\ell) + \mathfrak{R}(X_\ell) = R_\ell, \qquad \ell = 0, 1, \ldots$$

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) [Feitzinger/Hylla/Sachs '09, Hylla '10]

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

Inexact Kleinman's Iteration for the ARE

$$\mathfrak{R}'|_{X_\ell}(X_{\ell+1}) - \mathfrak{R}'|_{X_\ell}(X_\ell) + \mathfrak{R}(X_\ell) = R_\ell, \qquad \ell = 0, 1, \ldots$$

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,

$$0 \leq R_\ell \leq C^T C \quad \text{and} \quad 0 \leq R_\ell \leq N_\ell B B^T N_\ell.$$

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_∞, such that

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

Furthermore the convergence is quadratic and monotone with

$$0 \leq X_\infty \leq \cdots \leq X_{k+1} \leq X_k \leq \cdots \leq X_1.$$

Iterative Solution of the ARE

Inexact Newton for AREs (Convergence Result) [Feitzinger/Hylla/Sachs '09, Hylla '10]

Theorem

Let Assumption 1 hold,

$$0 \le R_\ell \le C^T C \quad \text{and} \quad 0 \le R_\ell \le N_\ell B B^T N_\ell.$$

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_∞, such that

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

Furthermore the convergence is quadratic and monotone with

$$0 \le X_\infty \le \cdots \le X_{k+1} \le X_k \le \cdots \le X_1.$$

Iterative Solution of the ARE

Inexact Newton for AREs (Remarks)

Weaker Condition [Hylla '10]

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.

Iterative Solution of the ARE

Inexact Newton for AREs (Remarks)

Weaker Condition [Hylla '10]

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.

Large Scale Difficulty

None of the conditions

- $R_\ell \leq C^T C$,
- $R_\ell \leq C^T C + X_\ell B B^T X_\ell$, and
- $0 \leq R_\ell \leq N_\ell B B^T N_\ell$,

can be tested in large scale applications.

Iterative Solution of the ARE

Inexact Newton for AREs (Remarks)

Weaker Condition

[Hylla '10]

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.

Large Scale Difficulty

Crucial for the proof. $\Rightarrow$ Lyapunov solver needs to ensure this.

None of the conditions

- $R_\ell \leq C^T C$,
- $R_\ell \leq C^T C + X_\ell B B^T X_\ell$, and
- $0 \leq R_\ell \leq N_\ell B B^T N_\ell$,

can be tested in large scale applications.

Solving Large Lyapunov Equations

Solving Large Lyapunov Equations

LRCF-ADI

e.g.,[Benner/Li/Penzl '08]

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

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

Algorithm

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

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

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

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

e.g.,[Benner/Li/Penzl '08]

Consider	$FX + XF^T = -GG^T \qquad F \in \mathbb{R}^{n\times n}, G \in \mathbb{R}^{n\times p}$
Task	Find $Z \in \mathbb{K}^{n,n_Z}$, such that $n_Z \ll n$ and $X \approx ZZ^H$

Algorithm

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

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

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

Stop if

- $\|V_i V_i^H\|$ is small, or
- $\|FZ_iZ_i^H + Z_iZ_i^HF^T + GG^T\|$ is small.

$$Z_0 = [\,] \Rightarrow X_0 = 0$$

$$\Rightarrow R_0 = GG^T \geq 0$$

[Hylla '10]: Then $\forall i : R_i \geq 0$,

and $\exists i_0 \quad \forall i \geq i_0 \quad R_i \leq C^T C$.

Solving Large Lyapunov Equations

G-LRCF-ADI

e.g.,[S. '09]

Consider $FXE^T + EXF^T = -GG^T \qquad E, F \in \mathbb{R}^{n\times n}, G \in \mathbb{R}^{n\times p}$

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

Algorithm

$$V_1 = \sqrt{-2p_1}(F + p_1E)^{-1}G, \qquad Z_1 = V_1$$

$$V_i = \frac{\sqrt{p_i}}{\sqrt{p_{i-1}}}\left[I - (p_i + \overline{p_{i-1}})(F + p_iE)^{-1}\right]EV_{i-1} \qquad Z_i = [Z_{i-1}\,V_i]$$

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

Stop if

- $\|V_iV_i^H\|$ is small, or
- $\|FZ_iZ_i^HE^T + EZ_iZ_i^HF^T + GG^T\|$ is small.

Solving Large Lyapunov Equations

G-LRCF-ADI

e.g.,[S. '09]

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

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

Algorithm

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

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

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

Stop if

- $\|V_i V_i^H\|$ is small, or
- $\|FZ_iZ_i^HE^T + EZ_iZ_i^HF^T + GG^T\|$ is small.

Patrick showed: Can ensure $Z \in \mathbb{R}^{n,n_Z}$ even if $\{p_1, \ldots, p_J\} \not\subset \mathbb{R}$

Solving Large Lyapunov Equations

LRCF-ADI with Galerkin-Projection-Acceleration

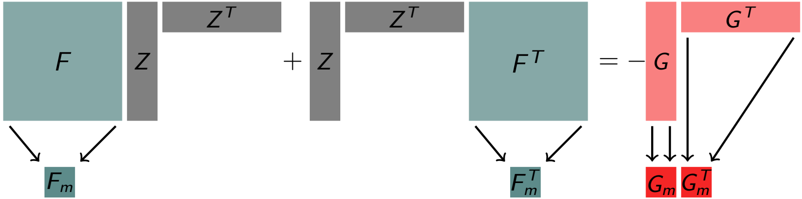
$$F\,Z\,Z^T + Z\,Z^T\,F^T = -G\,G^T$$

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



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

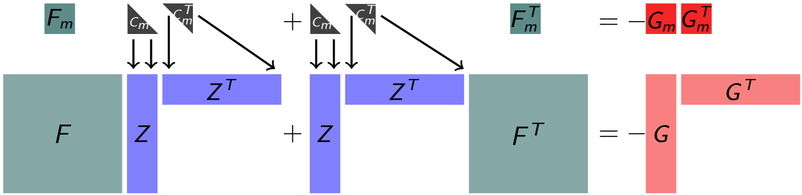
$$F_m \; C_m \, C_m^T \quad + C_m \, C_m^T \; F_m^T \quad = -G_m \, G_m^T$$

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



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

Projected ADI Step → LRCF-ADI-GP [S.'09, Benner/S.'10]

1. Compute the LRCF-ADI iterate Z_i
2. Compute orthogonal basis via QR factorization: $Q_i R_i \Pi_i = Z_i$[a]
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 $Z_i := Q_i \tilde{Z}$

[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 [S.'09, Benner/S.'10]

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 $Z_i := Q_i \tilde{Z}$

- 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

Solving Large Lyapunov Equations

LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step → LRCF-ADI-GP [S.'09, Benner/S.'10]

1. Compute the LRCF-ADI iterate Z_i
2. Compute orthogonal basis via QR factorization: $Q_iR_i\Pi_i = Z_i$
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 $Z_i := Q_i\tilde{Z}$

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

- solving generalized projected LE:

$$(Z_i^T F Z_i)\,\tilde{Z}\tilde{Z}^T\, Z_i^T Z_i + Z_i^T Z_i\,\tilde{Z}\tilde{Z}^T\,(Z_i^T F^T Z_i) = -Z_i^T G G^T Z_i$$

- using $Q_i := Z_i L_i$ where $(Z_i^T Z_i)^{-1} = L_i L_i^T$

- 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

Solving Large Lyapunov Equations

LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step →G-LRCF-ADI-GP [S.'09, Benner/S.'10]

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

LRCF-NM for the ARE

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

Consider $$\mathfrak{R}(X) := C^T C + A^T X + XA - XBB^T X = 0$$

Kleinman's Iteration for the ARE [Kleinman '68]

$$\mathfrak{R}'|_{X_\ell}(X_{\ell+1}) = \mathfrak{R}'|_{X_\ell}(X_\ell) - \mathfrak{R}(X_\ell), \qquad \ell = 0, 1, \ldots$$

where the Frechét derivative of $\mathfrak{R}$ at X is the Lyapunov operator

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

i.e., in every Newton step solve a

Lyapunov Equation [Kleinman '68]

$$(A - BB^T X_\ell)^T X_{\ell+1} + X_{\ell+1}(A - BB^T X_\ell) = -C^T C - X_\ell BB^T X_\ell.$$

LRCF-NM for the ARE

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

Factored Newton-Kleinman Iteration [Benner/Li/Penzl '99/'08]

$$F_\ell = A - BB^T X_\ell =: A - BK_\ell$$ is "sparse + low rank"

$$G_\ell = [C^T \;\; K_\ell^T]$$ is low rank factor

Find low rank factor $Z_\ell \in \mathbb{R}^{n,n_Z}$, where $n_Z \ll n$ and $X_\ell = Z_\ell Z_\ell^T$.

LRCF-NM for the ARE

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

Factored Newton-Kleinman Iteration [Benner/Li/Penzl '99/'08]

$$F_\ell = A - BB^T X_\ell =: A - BK_\ell \quad \text{is “sparse + low rank”}$$

$$G_\ell = [C^T \;\; K_\ell^T] \quad \text{is low rank factor}$$

- apply LRCF-ADI in every Newton step
- exploit structure of F_ℓ 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 [Benner/Li/Penzl '99/'08]

$$F_\ell = A - BB^T X_\ell =: A - BK_\ell \quad \text{is "sparse + low rank"}$$
$$G_\ell = [C^T \;\; K_\ell^T] \quad \text{is low rank factor}$$

- apply LRCF-ADI in every Newton step
- exploit structure of F_ℓ 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 [Benner/Li/Penzl '99/'08]

$$F_\ell = A - BB^T X_\ell =: A - BK_\ell \quad \text{is "sparse + low rank"}$$
$$G_\ell = [C^T \ \ K_\ell^T] \quad \text{is low rank factor}$$

- apply LRCF-ADI in every Newton step
- exploit structure of F_ℓ 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 [Benner/Li/Penzl '99/'08]

$$F_\ell = A - BB^T X_\ell =: A - BK_\ell \qquad \text{is “sparse + low rank”}$$
$$G_\ell = [C^T \; K_\ell^T] \qquad \text{is low rank factor}$$

- apply LRCF-ADI in every Newton step
- exploit structure of F_ℓ 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 [Benner/Li/Penzl '99/'08]

$$F_\ell = A - BB^T X_\ell E =: A - BK_\ell \quad \text{is “sparse + low rank”}$$
$$G_\ell = [C^T \;\; K_\ell^T] \quad \text{is low rank factor}$$

- apply LRCF-ADI in every Newton step
- exploit structure of F_ℓ 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: $Z = Z^{(k_{max})}$, such that ZZ^H approximates the solution X of

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

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

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

  $K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$
end for

LRCF-NM for the ARE

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

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: $Z = Z^{(k_{max})}$, 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** $k = 1, 2, \dots, k_{max}$ **do**
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \dots$ 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**

LRCF-NM for the ARE

Low-Rank Newton-ADI Variants

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 [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)

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

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

1: **for** $k = 1, 2, \ldots, k_{max}$ **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: $K^{(k)} = Z^{(k)}({Z^{(k)}}^HB)$
6: **end for**

LRCF-NM for the ARE

Low-Rank Newton-ADI Variants

Algorithm 2 Simpl. Low-Rank Cholesky Factor Newton Method (LRCF-NM-S)

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 (sub)optimal ADI shift parameters $p_1, p_2, \dots$ with respect to the matrix $F^{(0)} = A^T - K^{(0)} B^T$.
2: **for** $k = 1, 2, \dots, k_{max}$ **do**
3: $G^{(k)} = \left[\ C^T \quad K^{(k-1)}\ \right]$
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**

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

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

1: **for** $k = 1, 2, \dots, k_{max}$ **do**
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \dots$ with respect to the matrix $F^{(k)} = A^T - K^{(k-1)}B^T$.
3: $G^{(k)} = \left[\begin{array}{cc} C^T & K^{(k-1)} \end{array}\right]$
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**

LRCF-NM for the ARE

Low-Rank Newton-ADI Variants

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

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

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** $k = 1, 2, \ldots, k_{max}$ **do**
3: $\quad G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: $\quad$ 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: $\quad$ Project ARE, solve and prolongate solution
6: $\quad K^{(k)} = Z^{(k)}(Z^{(k)^H}B)$
7: **end for**

LRCF-NM for the ARE

Low-Rank Newton-ADI Variants

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

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

1: **for** $k = 1, 2, \dots, k_{max}$ **do**
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \dots$ 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**

LRCF-NM for the ARE

Low-Rank Newton-ADI Variants

Algorithm 6 Inexact Low-Rank Cholesky Factor Newton Method (I-LRCF-NM-S)

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 (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** $k = 1, 2, \ldots, k_{max}$ **do**

3: $G^{(k)} = \left[\begin{array}{cc} C^T & K^{(k-1)} \end{array}\right]$

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**

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

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

1: **for** $k = 1, 2, \ldots, k_{max}$ **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

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

5: Project ARE, solve and prolongate solution
6: $K^{(k)} = Z^{(k)}(Z^{(k)^H}B)$
7: **end for**

LRCF-NM for the ARE

Low-Rank Newton-ADI Variants

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

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

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** $k = 1, 2, \ldots, k_{max}$ **do**
3: $\quad G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: $\quad$ Compute $Z^{(k)}$ using (LRCF-ADI-GP) such that

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

5: $\quad$ Project ARE, solve and prolongate solution
6: $\quad K^{(k)} = Z^{(k)}(Z^{(k)}{}^H B)$
7: **end for**

Inexact LRCF-NM For the ARE

Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

Main Problem:

How can we ensure quadratic convergence without checking

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

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 $\mathfrak{R}(.)$ we have

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

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 $\mathfrak{R}(.)$ we have
$$\mathfrak{R}(Y) = \mathfrak{R}(X) + \mathfrak{R}'|_X(Y - X) + \frac{1}{2}\mathfrak{R}''|_X(Y - X, Y - X).$$

Recall the Inexact Kleinman step:
$$R_\ell = \mathfrak{R}'|_{X_\ell}(X_{\ell+1}) - \mathfrak{R}'|_{X_\ell}(X_\ell) + \mathfrak{R}(X_\ell)$$

Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

Main Problem:

How can we ensure quadratic convergence without checking

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

Due to the quadratic nature of $\mathfrak{R}(.)$ we have

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

Recall the Inexact Kleinman step:

$$R_\ell = \mathfrak{R}'|_{X_\ell}(X_{\ell+1}) - \mathfrak{R}'|_{X_\ell}(X_\ell) + \mathfrak{R}(X_\ell) = \mathfrak{R}'|_{X_\ell}(X_{\ell+1} - X_\ell) + \mathfrak{R}(X_\ell),$$

Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

Main Problem:

How can we ensure quadratic convergence without checking

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

Due to the quadratic nature of $\mathfrak{R}(.)$ we have

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

Recall the Inexact Kleinman step:

$$R_\ell = \mathfrak{R}'|_{X_\ell}(X_{\ell+1}) - \mathfrak{R}'|_{X_\ell}(X_\ell) + \mathfrak{R}(X_\ell) = \mathfrak{R}'|_{X_\ell}(X_{\ell+1} - X_\ell) + \mathfrak{R}(X_\ell),$$

and thus

$$\mathfrak{R}(X_{\ell+1}) = R_\ell + \frac{1}{2}\mathfrak{R}''|_{X_\ell}(X_{\ell+1} - X_\ell, X_{\ell+1} - X_\ell).$$

Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

Main Problem:

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

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

Recall the Inexact Kleinman step:
$$R_\ell = \mathfrak{R}'|_{X_\ell}(X_{\ell+1}) - \mathfrak{R}'|_{X_\ell}(X_\ell) + \mathfrak{R}(X_\ell) = \mathfrak{R}'|_{X_\ell}(X_{\ell+1} - X_\ell) + \mathfrak{R}(X_\ell),$$

and thus
$$\mathfrak{R}(X_{\ell+1}) = R_\ell + \frac{1}{2}\mathfrak{R}''|_{X_\ell}(X_{\ell+1} - X_\ell, X_{\ell+1} - X_\ell).$$

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 $\mathfrak{R}(.)$ we have

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

Recall the Inexact Kleinman step:

$$R_\ell = \mathfrak{R}'|_{X_\ell}(X_{\ell+1}) - \mathfrak{R}'|_{X_\ell}(X_\ell) + \mathfrak{R}(X_\ell) = \mathfrak{R}'|_{X_\ell}(X_{\ell+1} - X_\ell) + \mathfrak{R}(X_\ell),$$

and thus

$$\mathfrak{R}(X_{\ell+1}) = R_\ell + \frac{1}{2}\mathfrak{R}''|_{X_\ell}(N_\ell, N_\ell).$$

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 $\mathfrak{R}(.)$ we have

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

Recall the Inexact Kleinman step:

$$R_\ell = \mathfrak{R}'|_{X_\ell}(X_{\ell+1}) - \mathfrak{R}'|_{X_\ell}(X_\ell) + \mathfrak{R}(X_\ell) = \mathfrak{R}'|_{X_\ell}(X_{\ell+1} - X_\ell) + \mathfrak{R}(X_\ell),$$

and thus

$$\mathfrak{R}(X_{\ell+1}) = R_\ell + \frac{1}{2} N_\ell BB^T N_\ell$$

Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

New Question

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

Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

New Question

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

Riccati residual

inner Lyapunov residual

$$
\begin{aligned}
N_\ell B B^T N_\ell &= (X_{\ell+1} - X_\ell) B B^T (X_{\ell+1} - X_\ell) \\
&= X_{\ell+1} B B^T X_{\ell+1} + X_\ell B B^T X_\ell - X_\ell B B^T X_{\ell+1} - X_{\ell+1} B B^T X_\ell
\end{aligned}
$$

Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

New Question

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

Riccati residual

inner Lyapunov residual

$$\begin{aligned} N_\ell B B^T N_\ell &= (X_{\ell+1} - X_\ell) B B^T (X_{\ell+1} - X_\ell) \\ &= X_{\ell+1} B B^T X_{\ell+1} + X_\ell B B^T X_\ell - X_\ell B B^T X_{\ell+1} - X_{\ell+1} B B^T X_\ell \\ &= K_{\ell+1}^T K_{\ell+1} + K_\ell^T K_\ell - K_{\ell+1}^T K_\ell - K_\ell^T K_{\ell+1} \end{aligned}$$

Inexact LRCF-NM For the ARE

Accuracy control for the (G-)LRCF-ADI

New Question

How can we exploit $\underbrace{\mathfrak{R}(X_{\ell+1})}_{\text{Riccati residual}} = \underbrace{R_\ell}_{\text{inner Lyapunov residual}} + \frac{1}{2}N_\ell BB^T N_\ell$ to control the ADI accuracy?

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

- $\|R_\ell\|_2$ is stopping criterion in the LRCF-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

Implementation

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

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

$\Rightarrow K_{\ell+1}^{(i+1)} = B^T Z_{i+1} Z_{i+1}^T \quad = B^T Z_i Z_i^T \quad + B^T V_i V_i^T \quad = K_{\ell+1}^{(i)} + B^T V_i V_i^T$

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.

$\Rightarrow K_{\ell+1}^{(i+1)} = B^T Z_{i+1} Z_{i+1}^T E = B^T Z_i Z_i^T E + B^T V_i V_i^T E = K_{\ell+1}^{(i)} + B^T V_i V_i^T E$

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.

$\Rightarrow K_{\ell+1}^{(i+1)} = B^T Z_{i+1} Z_{i+1}^T E = B^T Z_i Z_i^T E + B^T V_i V_i^T E = K_{\ell+1}^{(i)} + B^T V_i V_i^T E$

We can force quadratic convergence via

$$\begin{aligned} \|\mathfrak{R}(X_{\ell+1})\|_2 &\leq \|R_\ell\|_2 + \frac{1}{2}\|K_{\ell+1}^T K_{\ell+1} + K_\ell^T K_\ell - K_{\ell+1}^T K_\ell - K_\ell^T K_{\ell+1}\|_2 \\ &\leq \varepsilon_\ell := \alpha \mathfrak{R}(X_\ell)^2 \end{aligned}$$

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

`otto`

Tests and Conclusions

Rail 1357

I-LRCF-NM-S

step	rel.residual	# ADI	term.flag	inex.tol (ε_ℓ)
1	6.208639 e-01	1	R	1
2	1.045024 e-01	4	R	3.854720 e-01
3	3.782191 e-03	8	R	1.092076 e-02
4	1.106234 e-05	20	R	1.430497 e-05
5	9.728725 e-11	43	L	1.223754 e-10

time: 2.73s

Tests and Conclusions

Rail 1357

I-LRCF-NM-S

step	rel.residual	# ADI	term.flag	inex.tol (ε_ℓ)
1	6.208639 e-01	1	R	1
2	1.045024 e-01	4	R	3.854720 e-01
3	3.782191 e-03	8	R	1.092076 e-02
4	1.106234 e-05	20	R	1.430497 e-05
5	9.728725 e-11	43	L	1.223754 e-10

time: 2.73s

LRCF-NM

step	rel.residual	# ADI
1	6.739966 e-05	44
2	9.423553 e-09	45
3	7.831133 e-11	45

time: 4.60s

Tests and Conclusions

Rail 1357

I-LRCF-NM-S

step	rel.residual	# ADI	term.flag	inex.tol (ε_ℓ)
1	6.208639 e-01	1	R	1
2	1.045024 e-01	4	R	3.854720 e-01
3	3.782191 e-03	8	R	1.092076 e-02
4	1.106234 e-05	20	R	1.430497 e-05
5	9.728725 e-11	43	L	1.223754 e-10

time: 2.73s

LRCF-NM

step	rel.residual	# ADI
1	6.739966 e-05	44
2	9.423553 e-09	45
3	7.831133 e-11	45

time: 4.60s

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

step	rel.residual	# ADI
1	4.361686 e-11	31

time: 3.78s

Tests and Conclusions

Rail 1357

I-LRCF-NM-S

step	rel.residual	# ADI	term.flag	inex.tol (ε_ℓ)
1	6.208639 e-01	1	R	1
2	1.045024 e-01	4	R	3.854720 e-01
3	3.782191 e-03	8	R	1.092076 e-02
4	1.106234 e-05	20	R	1.430497 e-05
5	9.728725 e-11	43	L	1.223754 e-10

time: 2.73s

LRCF-NM

step	rel.residual	# ADI
1	6.739966 e-05	44
2	9.423553 e-09	45
3	7.831133 e-11	45

time: 4.60s

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

step	rel.residual	# ADI
1	6.272812 e-12	44

time: 1.47s

Tests and Conclusions

Rail 5177

I-LRCF-NM-S

step	rel.residual	# ADI	term.flag	inex.tol (ε_ℓ)
1	6.674262 e-01	1	R	1
2	1.733687 e-01	3	R	4.454578 e-01
3	9.853188 e-03	7	R	3.005672 e-02
4	8.072432 e-05	19	R	9.708532 e-05
5	2.761797 e-09	41	R	6.516416 e-09
6	4.808556 e-11	53	L	7.627521 e-18

time: 18.79s

Tests and Conclusions

Rail 5177

I-LRCF-NM-S

step	rel.residual	# ADI	term.flag	inex.tol (ε_ℓ)
1	6.674262 e-01	1	R	1
2	1.733687 e-01	3	R	4.454578 e-01
3	9.853188 e-03	7	R	3.005672 e-02
4	8.072432 e-05	19	R	9.708532 e-05
5	2.761797 e-09	41	R	6.516416 e-09
6	4.808556 e-11	53	L	7.627521 e-18

time: 18.79s

LRCF-NM

step	rel.residual	# ADI
1	2.691724 e-05	54
2	2.321039 e-09	54
3	8.044616 e-11	54

time: 23.49s

Tests and Conclusions

Rail 5177

I-LRCF-NM-S

step	rel.residual	# ADI	term.flag	inex.tol (ε_ℓ)
1	6.674262 e-01	1	R	1
2	1.733687 e-01	3	R	4.454578 e-01
3	9.853188 e-03	7	R	3.005672 e-02
4	8.072432 e-05	19	R	9.708532 e-05
5	2.761797 e-09	41	R	6.516416 e-09
6	4.808556 e-11	53	L	7.627521 e-18

time: 18.79s

LRCF-NM

step	rel.residual	# ADI
1	2.691724 e-05	54
2	2.321039 e-09	54
3	8.044616 e-11	54

time: 23.49s

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

step	rel.residual	# ADI
1	9.762903 e-11	38

time: 11.62s

Tests and Conclusions

Rail 5177

I-LRCF-NM-S

step	rel.residual	# ADI	term.flag	inex.tol (ε_ℓ)
1	6.674262 e-01	1	R	1
2	1.733687 e-01	3	R	4.454578 e-01
3	9.853188 e-03	7	R	3.005672 e-02
4	8.072432 e-05	19	R	9.708532 e-05
5	2.761797 e-09	41	R	6.516416 e-09
6	4.808556 e-11	53	L	7.627521 e-18

time: 18.79s

LRCF-NM

step	rel.residual	# ADI
1	2.691724 e-05	54
2	2.321039 e-09	54
3	8.044616 e-11	54

time: 23.49s

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

step	rel.residual	# ADI
1	7.204343 e-12	54

time: 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

[Wong/Balakrishnan '04-'07]

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

An LRCF-QADI Projection Method

Newton-Kleinman-ADI vs. QADI

QADI [Wong/Balakrishnan '04-'07]

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

Newton-Kleinman-ADI [Kleinman '68]

$$
\begin{aligned}
(A^T - K_{j-1}^T B^T + p_k I) X_{k-\frac{1}{2}}^T &= -Q - K_{j-1}^T K_{j-1} - X_{k-1}^T (A - BK_{j-1} - p_k I),\\
(A^T - K_{j-1}^T B^T + p_k I) X_k &= -Q - K_{j-1}^T K_{j-1} - X_{k-\frac{1}{2}} (A - BK_{j-1} - p_k I).
\end{aligned}
$$

An LRCF-QADI Projection Method

Newton-Kleinman-ADI vs. QADI

QADI

[Wong/Balakrishnan '04-'07]

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

Newton-Kleinman-ADI

[Kleinman '68]

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

An LRCF-QADI Projection Method

Newton-Kleinman-ADI vs. QADI

QADI [Wong/Balakrishnan '04-'07]

$$
\begin{aligned}
(A^T - K_{j-1}^T B^T + p_j I) X_{j-\frac{1}{2}}^T &= -Q - K_{j-1}^T K_{j-1} - X_{j-1}^T (A - BK_{j-1} - p_j I),\\
(A^T - K_{j-\frac{1}{2}}^T B^T + p_j I) X_j &= -Q - K_{j-\frac{1}{2}}^T K_{j-\frac{1}{2}} - X_{j-\frac{1}{2}} (A - BK_{j-\frac{1}{2}} - p_j I).
\end{aligned}
$$

Newton-Kleinman-ADI [Kleinman '68]

$$
\begin{aligned}
(A^T - K_{j-1}^T B^T + p_k I) X_{k-\frac{1}{2}}^T &= -Q - K_{j-1}^T K_{j-1} - X_{k-1}^T (A - BK_{j-1} - p_k I),\\
(A^T - K_{j-1}^T B^T + p_k I) X_k &= -Q - K_{j-1}^T K_{j-1} - X_{k-\frac{1}{2}} (A - BK_{j-1} - p_k I).
\end{aligned}
$$

An LRCF-QADI Projection Method

Newton-Kleinman-ADI vs. QADI

QADI

[WONG/BALAKRISHNAN '04-'07]

$$
\begin{aligned}
(A^T - K_{j-1}^T B^T + p_j I) X_{j-\frac{1}{2}}^T &= -Q - K_{j-1}^T K_{j-1} - X_{j-1}^T (A - BK_{j-1} - p_j I),\\
(A^T - K_{j-\frac{1}{2}}^T B^T + p_j I) X_j &= -Q - K_{j-\frac{1}{2}}^T K_{j-\frac{1}{2}} - X_{j-\frac{1}{2}} (A - BK_{j-\frac{1}{2}} - p_j I).
\end{aligned}
$$

Newton-Kleinman-ADI

[KLEINMAN '68]

$$
\begin{aligned}
(A^T - K_{j-1}^T B^T + p_k I) X_{k-\frac{1}{2}}^T &= -Q - K_{j-1}^T K_{j-1} - X_{k-1}^T (A - BK_{j-1} - p_k I),\\
(A^T - K_{j-1}^T B^T + p_k I) X_k &= -Q - K_{j-1}^T K_{j-1} - X_{k-\frac{1}{2}} (A - BK_{j-1} - p_k I).
\end{aligned}
$$

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 $A - BK^{(0)^T}$ 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: **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: $G^{(k)} = \left[\begin{array}{cc} C^T & K^{(k-1)} \end{array}\right]$
4: Compute $Z^{(k)}$ using (LRCF-ADI) or (LRCF-ADI-GP) such that $F^{(k)} Z^{(k)} Z^{(k)^H} + Z^{(k)} Z^{(k)^H} F^{(k)^T} \approx -G^{(k)} G^{(k)^T}$.
5: $K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$
6: **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: $G = \left[\begin{array}{cc} C^T & K^{(0)} \end{array}\right]$
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: $V_k = \sqrt{\mathrm{Re}\,(p_k)/\,\mathrm{Re}\,(p_{k-1})}\left(V_{k-1} - (p_k + \overline{p_{k-1}})\tilde{V}\right)$
7: $Z_k = [Z_{k-1} \;\; V_k]$
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

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

Determine QADI shift parameters $p_1, p_2, \dots$
$G = \begin{bmatrix} C^T & K^{(0)} \end{bmatrix}$
for $k = 1, 2, \dots, k_{max}$ **do**
  $F^{(k)} = A^T - K^{(k-1)}B^T$
  For $\tilde{V}$ solve $(F^{(k)} + p_k I)\tilde{V} = V_{k-1}$
  $V_k = \sqrt{\mathrm{Re}\,(p_k)/\,\mathrm{Re}\,(p_{k-1})}\left(V_{k-1} - (p_k + \overline{p_{k-1}})\tilde{V}\right)$
  $Z_k = [Z_{k-1} \;\; V_k]$
  Project ARE, solve and prolongate solution
  $K^{(k)} = Z^{(k)}(Z^{(k)^H}B)$
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

Observation

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

"Is this the holy grail?"

Question

Can we avoid updating F in the Newton based solvers completely?

The RicADI Projection Method

Invariance of the Krylov Subspaces

Factored Newton-Kleinman Iteration [Benner/Li/Penzl '99/'08]

$$F_\ell = A - BB^T X_\ell E =: A - BK_\ell \qquad \text{is "sparse + low rank"}$$

$$G_\ell = [C^T \; K_\ell^T] \qquad \text{is low rank factor}$$

Need to solve:

$$F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T.$$

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

$$F_\ell = A - BB^T X_\ell E =: A - BK_\ell \quad \text{is "sparse + low rank"}$$
$$G_\ell = [C^T \; K_\ell^T] \quad \text{is low rank factor}$$

Need to solve:

$$F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T.$$

Feedback Invariance of Subspaces [Benner/Beckermann '11]

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

$$\begin{aligned}
\mathcal{K}_m(F_\ell^T, G_\ell) &= \mathcal{K}_m((A - BK_\ell)^T, [C^T \; K_\ell^T]) \\
&= \mathrm{sp}[C^T, K_\ell^T, A^T C^T - K_\ell^T B^T C^T, A^T K_\ell^T - K_\ell^T B^T K_\ell^T, \dots] \\
&= \mathrm{sp}[C^T, K_\ell^T, A^T C^T, A^T K_\ell^T, \dots] \\
&= \mathcal{K}_m(A^T, [C^T \; K_\ell]) = \mathcal{K}_m(A^T, G_\ell)
\end{aligned}$$

The RicADI Projection Method

Invariance of the Krylov Subspaces

Factored Newton-Kleinman Iteration [Benner/Li/Penzl '99/'08]

$F_\ell = A - BB^T X_\ell E =: A - BK_\ell$ is "sparse + low rank"

$G_\ell = [C^T \ K_\ell^T]$ is low rank factor

Need to solve:

$$F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T.$$

Feedback Invariance of Subspaces [Benner/Beckermann '11]

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

$$\mathrm{sp}[-K_\ell^T B^T C^T] \subseteq \mathrm{sp}[K_\ell^T]$$

$$\mathrm{sp}[-K_\ell^T B^T K_\ell^T] \subseteq \mathrm{sp}[K_\ell^T]$$

$$\begin{aligned}
\mathcal{K}_m(F_\ell^T, G_\ell) &= \mathcal{K}_m((A - BK_\ell)^T, [C^T \ K_\ell^T]) \\
&= \mathrm{sp}[C^T, K_\ell^T, A^T C^T - K_\ell^T B^T C^T, A^T K_\ell^T - K_\ell^T B^T K_\ell^T, \ldots] \\
&= \mathrm{sp}[C^T, K_\ell^T, A^T C^T, A^T K_\ell^T, \ldots] \\
&= \mathcal{K}_m(A^T, [C^T \ K_\ell]) = \mathcal{K}_m(A^T, G_\ell)
\end{aligned}$$

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: $Z = Z_{i_{max}} \in \mathbb{C}^{n \times t_{i_{max}}}$, such that $ZZ^H \approx X$ approximates the solution X of $C^TC + A^TX + XA - XBB^TX = 0$.

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 $(F + p_1 I)\, V_1 = \sqrt{-2\,\mathrm{Re}\,(p_1)}G$
3: $Z_1 = V_1, \quad i = 2$
4: **repeat**
5: For $\tilde{V}$ solve $(F + p_i I)\tilde{V} = V_{i-1}$
6: $V_i = \sqrt{\mathrm{Re}\,(p_i)/\,\mathrm{Re}\,(p_{i-1})}\left(V_{i-1} - (p_i + \overline{p_{i-1}})\tilde{V}\right)$
7: $Z_i = [Z_{i-1} \;\; V_i]$, $i = i + 1$
8: Project ARE, solve and prolongate solution
9: **until** $\|C^TC + A^TZ_iZ_i^T + Z_iZ_i^TA - Z_iZ_i^TBB^TZ_iZ_i^T\| \leq TOL$

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 $F = A - BK^{(0)}$ is stable.
Output: $Z = Z_{i_{max}} \in \mathbb{C}^{n \times t_{i_{max}}}$, such that $ZZ^H \approx X$ approximates the solution X of $C^TC + A^TX + XA - XBB^TX = 0$.

1: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \dots$ with respect to the matrix F
2: For V_1 solve $(F + p_1 I)\, V_1 = \sqrt{-2\,\mathrm{Re}\,(p_1)}\, C$
3: $Z_1 = V_1, \quad i = 2$
4: **repeat**
5: For $\tilde{V}$ solve $(F + p_i I)\, \tilde{V} = V_{i-1}$
6: $V_i = \sqrt{\mathrm{Re}\,(p_i) / \mathrm{Re}\,(p_{i-1})} \left(V_{i-1} - (p_i + \overline{p_{i-1}})\tilde{V} \right)$
7: $Z_i = [Z_{i-1} \;\; V_i]$, $i = i + 1$
8: Project ARE, solve and prolongate solution
9: **until** $\|C^TC + A^TZ_iZ_i^T + Z_iZ_i^TA - Z_iZ_i^TBB^TZ_iZ_i^T\| \leq TOL$

Compute projection basis via $Q_i = Z_iL_i$, where

$$L_i = U(:, 1:k)\sqrt{\Sigma(1:k, 1:k)^{-1}}$$

and

$$[U, \Sigma, V] = svd(Z^TZ)$$

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 (G-LRCF-RicADI) [BENNER/KÖHLER/S. '11]

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

Output: $Z = Z_{i_{max}} \in \mathbb{C}^{n \times t_{i_{max}}}$, such that $ZZ^H \approx X$ approximates the solution X of $C^TC + A^TXE + E^TXA - E^TXBB^TXE = 0$.

Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \dots$ with respect to the matrix F

For V_1 solve $(F + p_1 E)\, V_1 = \sqrt{-2\,\mathrm{Re}\,(p_1)}G$

$Z_1 = V_1, \quad i = 2$

repeat

For $\tilde{V}$ solve $(F + p_i E)\tilde{V} = EV_{i-1}$

$V_i = \sqrt{\mathrm{Re}\,(p_i)/\,\mathrm{Re}\,(p_{i-1})}\left(V_{i-1} - (p_i + \overline{p_{i-1}})\tilde{V}\right)$

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

Project ARE, solve and prolongate solution

until $\|C^TC + A^TZ_iZ_i^TE + E^TZ_iZ_i^TA - E^TZ_iZ_i^TBB^TZ_iZ_i^TE\| \leq TOL$

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

`otto`

Numerical Results

MATLAB

	LRCF-NM	LRCF-NM-S	LRCF-NM-GP	RicADI
FDM 40k	135.499 s	83.953 s	13.629 s	11.760 s
FDM 250k	1 912.790 s	1 168.900 s	139.102 s	121.616 s
Rail 79k	1 775.280 s	1 784.001 s	131.280 s	80.854 s

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

Numerical Results
MATLAB

	LRCF-NM	LRCF-NM-S	LRCF-NM-GP	RicADI
FDM 40k	135.499 s	83.953 s	13.629 s	11.760 s
FDM 250k	1 912.790 s	1 168.900 s	139.102 s	121.616 s
Rail 79k	1 775.280 s	1 784.001 s	131.280 s	80.854 s

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

	LRCF-NM	LRCF-NM-S	LRCF-NM-GP	RicADI
FDM 40k	6.619 e-10	6.360 e-10	2.611 e-12	3.027 e-10
FDM 250k	1.954 e-10	1.038 e-10	4.324 e-12	6.915 e-11
Rail 79k	4.709 e-10	4.710 e-10	3.004 e-10	2.136 e-09

Table: Final residuals on `otto` using MATLAB 2010b

Numerical Results

C

	LRCF-NM	LRCF-NM-S	LRCF-NM-GP	RicADI
FDM 40k	44.84 s	37.664 s	4.160 s	3.440 s
FDM 250k	1 568.94 s	462.653 s	135.854 s	127.716 s
Rail 79k	1 306.99 s	1 468.850 s	56.379 s	22.956 s

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

Numerical Results

C

	LRCF-NM	LRCF-NM-S	LRCF-NM-GP	RicADI
FDM 40k	44.84 s	37.664 s	4.160 s	3.440 s
FDM 250k	1 568.94 s	462.653 s	135.854 s	127.716 s
Rail 79k	1 306.99 s	1 468.850 s	56.379 s	22.956 s

Table: Computation time in seconds on otto using C

	LRCF-NM	LRCF-NM-S	LRCF-NM-GP	RicADI
FDM 40k	6.619 e-10	6.350 e-10	5.901 e-11	3.001 e-10
FDM 250k	1.960 e-11	5.318 e-11	7.912 e-12	8.163 e-10
Rail 79k	2.019 e-10	2.019 e-10	2.240 e-11	9.528 e-10

Table: Final residuals on otto using C