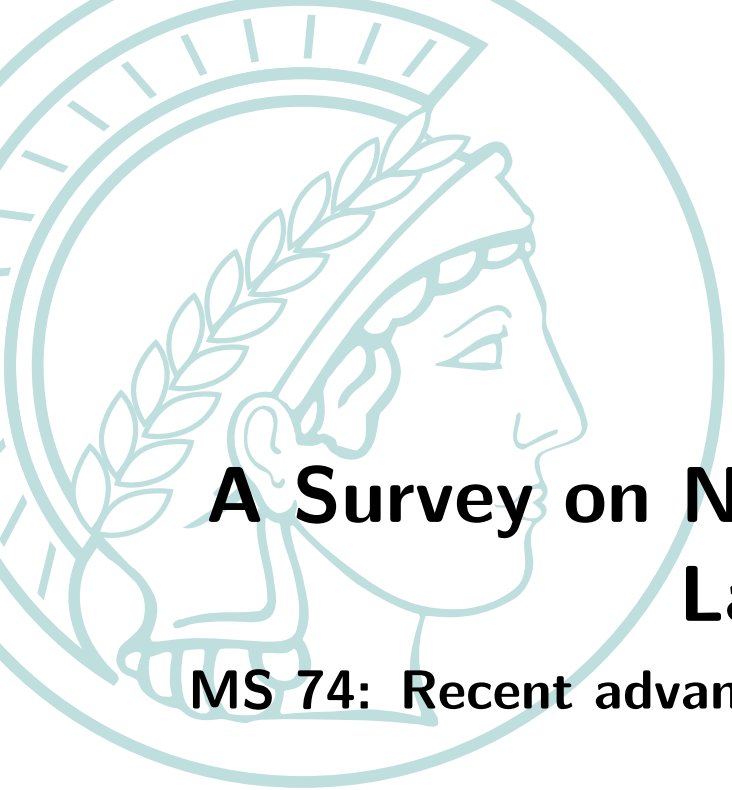


SIAM Conference on Applied Linear Algebra
Valencia, June 22, 2012

A Survey on Newton-ADI Based Solvers for Large Scale AREs

MS 74: Recent advances in the numerical solution of large scale matrix equations

Jens Saak

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

MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

Outline

1. (Inexact) Newton Methods for AREs
2. Solving Large Lyapunov Equations
3. LRCF-NM for the ARE
4. Inexact LRCF-NM for the ARE
5. Numerical Results
6. Current and future work

(Inexact) Newton Methods for AREs

Basic Concepts

[Kleinman '68, Feitzinger/Hylla/Sachs '09]

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

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

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

(Inexact) Newton Methods for AREs

Basic Concepts

[Kleinman '68, Feitzinger/Hylla/Sachs '09]

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

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

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

(Inexact) Newton Methods for AREs

Basic Concepts

[Kleinman '68, Feitzinger/Hylla/Sachs '09]

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

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

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

(Inexact) Newton Methods for AREs

Convergence Result

[Kleinman '68, Lancaster/Rodman '95, Feitzinger/Hylla/Sachs '09]

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

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

(Inexact) Newton Methods for AREs

Convergence Result (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.

(Inexact) Newton Methods for AREs

Convergence Result (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

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

can be tested efficiently in large scale applications.

(Inexact) Newton Methods for AREs

Convergence Result (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

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

can be tested efficiently in large scale applications.

Solving Large Lyapunov Equations

(G-)LRCF-ADI

e.g., [Benner/Li/Penzl '08, 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 $Z \in \mathbb{R}^{n,n_Z}$, such that $n_Z \ll n$ and $X \approx ZZ^T$.

Algorithm

$$V_1 = (F + p_1 E)^{-1} G, \qquad Z_1 = \sqrt{-2\operatorname{Re}(p_1)}\, V_1,$$
$$V_i = \left[I - (p_i + \overline{p_{i-1}})(F + p_i E)^{-1}\right] E V_{i-1}, \qquad Z_i = [Z_{i-1} \quad \sqrt{-2\operatorname{Re}(p_i)}\, V_i].$$

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

Stop if

- $\|V_i V_i^H\|$ is small, or
- $\|FZ_i Z_i^T E^T + EZ_i Z_i^T F^T + GG^T\|$ is small.

Solving Large Lyapunov Equations

(G-)LRCF-ADI

e.g., [Benner/Li/Penzl '08, 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 $Z \in \mathbb{R}^{n,n_Z}$, such that $n_Z \ll n$ and $X \approx ZZ^T$.

Algorithm

$$V_1 = (F + p_1 E)^{-1} G, \qquad Z_1 = \sqrt{-2\,\mathrm{Re}\,(p_1)}\, V_1,$$
$$V_i = \left[I - (p_i + \overline{p_{i-1}})(F + p_i E)^{-1}\right] E V_{i-1}, \qquad Z_i = \left[Z_{i-1} \quad \sqrt{-2\,\mathrm{Re}\,(p_i)}\, V_i\right].$$

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

Stop if

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

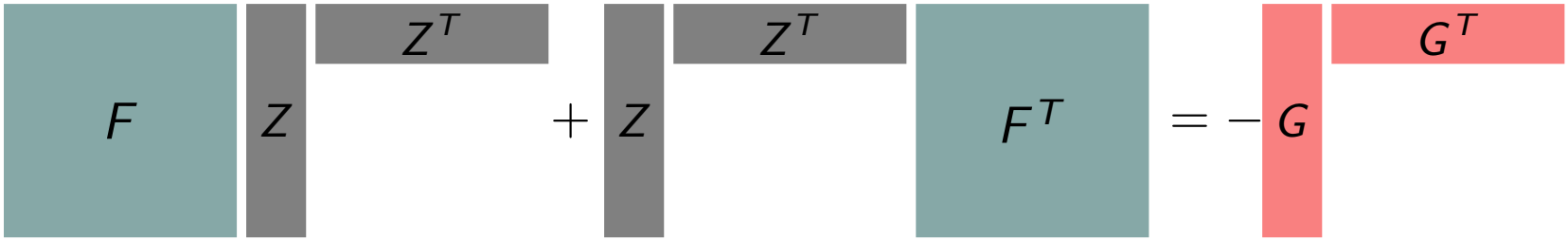
$Z_0 = [] \Rightarrow X_0 = 0$

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

[Hylla '10]: Then $\forall i : R_i \geq 0$, and $\exists i_0 \;\; \forall i \geq i_0 \;\; R_i \leq C^T C$.

Solving Large Lyapunov Equations

LRCF-ADI with Galerkin-Projection-Acceleration



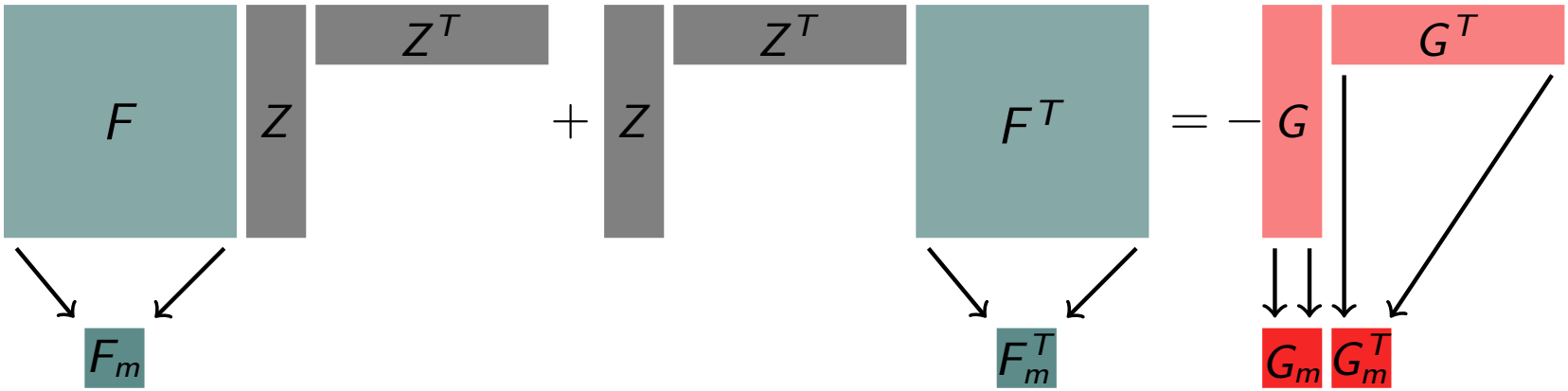
Legend:

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



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

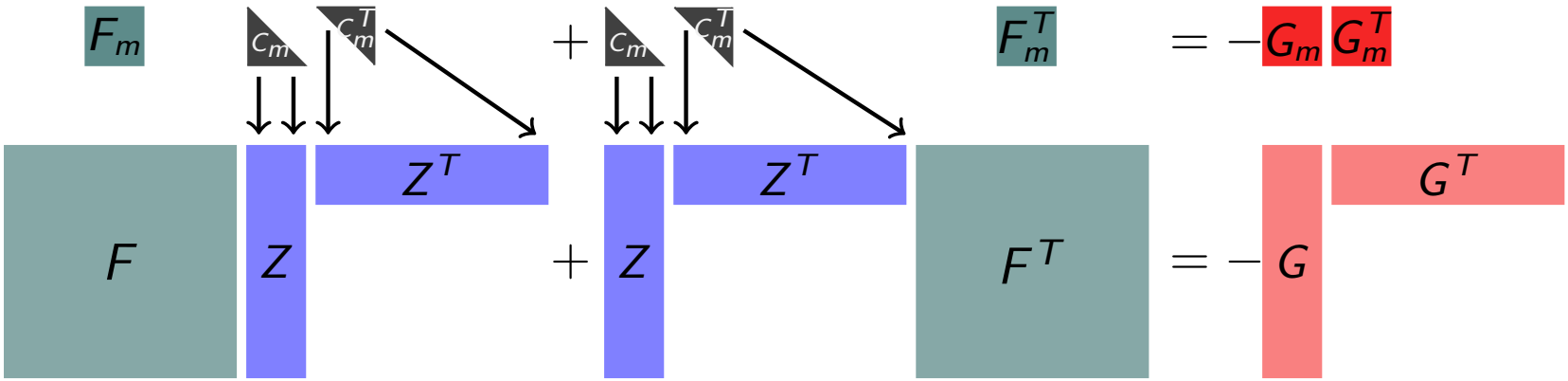
Legend:

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



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 $\rightarrow$ 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
2. Compute orthogonal basis via QR factorization: $Q_i R_i \Pi_i = Z_i$
3. Solve (for $\tilde{Z}$) the projected Lyapunov equation

$$(Q_i^T F \ldots$$

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

Compute projection basis via $Q_i = Z_i L_i$, where

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

and

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

to avoid problems with (almost) rank deficient Z_i and still avoid frequent orthogonalization.

- 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 the accuracy, when $\|.\|_2$ is used in stopping criteria.

Reason: $$\|A\|_2 \geq \sqrt{\|E\|_2}\|A\|_E$$

Recall norm discussion in R. Herzogs talk (IP 6).

(Inexact) Newton Methods for AREs

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

[Kleinman '68, Feitzinger/Hylla/Sachs '09]

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

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

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

LRCF-NM for the ARE

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

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

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

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]

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

- 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

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

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

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

5: $K^{(k)} = Z^{(k)} (Z^{(k)\,H} B)$
6: **end for**

LRCF-NM for the ARE

Low-Rank Newton-ADI (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** $k = 1, 2, \ldots, k_{max}$ **do**
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \ldots$ with respect to the matrix $F^{(k)} = A^T E^{-T} - K^{(k-1)} B^T E^{-T}$.
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

Implicit Factored Newton Iteration

- When searching for feedbacks K in PDE control avoid forming the solution X, or its factor Z, but directly update K.

LRCF-NM for the ARE

Low-Rank Newton-ADI Variants

Implicit Factored Newton Iteration

- When searching for feedbacks K in PDE control avoid forming the solution X, or its factor Z, but directly update K.

Simplified Factored Newton Iteration

1. Reuse ADI shifts from first step.
2. Compute shifts according to desired closed loop matrix.

LRCF-NM for the ARE

Low-Rank Newton-ADI Variants

Implicit Factored Newton Iteration

- When searching for feedbacks K in PDE control avoid forming the solution X, or its factor Z, but directly update K.

Simplified Factored Newton Iteration

1. Reuse ADI shifts from first step.
2. Compute shifts 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

Implicit Factored Newton Iteration

- When searching for feedbacks K in PDE control avoid forming the solution X, or its factor Z, but directly update K.

Simplified Factored Newton Iteration

1. Reuse ADI shifts from first step.
2. Compute shifts 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 $A - BK^{(0)^T}$ 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: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
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**

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 $p_1, p_2, \ldots$ 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) such that

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

5: $\quad K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$
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

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

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: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
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**

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

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

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: $\quad G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
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

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

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: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
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 Simpl. 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: $\quad G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: $\quad$ Compute $Z^{(k)}$ using (LRCF-ADI) such that

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

5: $\quad K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$
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

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

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: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
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 Simpl. 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

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

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: $\quad G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
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

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

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

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

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

Inexact LRCF-NM for the ARE

Accuracy control for the (G-)LRCF-ADI

Main Problem:

How can we ensure quadratic convergence without checking

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

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

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

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

Riccati residual

inner Lyapunov residual

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

- $\|R_\ell\|_2$ is stopping criterion in the LRCF-ADI.
- $\|\frac{1}{2} N_\ell B B^T N_\ell\|_2$ 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.

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

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:

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

Numerical Results

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`

Numerical Results

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

Numerical Results

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

Numerical Results

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

Numerical Results

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

Numerical Results

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

Numerical Results

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

Numerical Results

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

Numerical Results

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

Current and future work

Top 5 related activities

- **Inexact ADI steps:**
  Iterative solution and preconditioning of shifted linear systems in the ADI step.

(with H. Weichelt, M. Stoll, K. Ahuja, and T. Reis)

- Boundary feedback stabilization of linearized Navier-Stokes equations.(within DFG SPP1253)

$$F_\ell = \begin{bmatrix} A - BK_\ell + p_i E & G \\ G^T & 0 \end{bmatrix} \Rightarrow \text{LSC-based block preconditioners.}$$

- Recycling methods for shifted linear systems with multiple right hand sides.
- Error control and early stopping of the inner iteration.

Current and future work

Top 5 related activities

- **RicADI:**
  Rational Krylov style ADI version tackling the ARE directly.

(with M. Köhler)

- LRCF-NM-GP usually stops after 1 Newton step $\rightsquigarrow$ merge inner and outer iterations and monitor Riccati residual directly.
  $\Rightarrow$ Rational Krylov style method.
- Shift parameters?

Current and future work

Top 5 related activities

- **Hardware exploiting implementation:**
  Multicore and Manycore (i.e., GPU) accelerated implementation.

(with M. Köhler and A. Remón)

- Optimal exploitation of modern MultiCore CPUs.
- Offload expensive Linear algebra to the GPU for acceleration, wherever possible.

Current and future work

Top 5 related activities

- **Adaptive Shift strategies:**
  Update shifts during the iteration instead of pre-computation before the ADI.

(with M. Hund and T. Reis)

- Projection based updates of ADI shifts
  (recall talks by B. Beckermann (MS 8) and T.Breiten (MS 13))
  $\rightsquigarrow$ ADI counterpart of RKSM.
- (Discretization) hierarchy based selection of shifts.

Current and future work

Top 5 related activities

- **Differential Riccati equations:**
  Incorporate inexactness and Galerkin ideas in LR-DRE solvers of Rosenbrock and BDF type.

(with N. Lang and H. Mena)

$$E^T \dot{X} E = -C^T C - A^T X E - E^T X A + E^T X B B^T X E = -\mathfrak{R}(X),$$
$$X(t_f) = X_{t_f}.$$

- BDF methods solve AREs in each time step.
- Rosenbrock methods solve Lyapunov equation in each time step.
- Can apply projection ideas as above!
- Can we do so in the ODE solvers themselves?

Current and future work

Top 5 related activities

- **Differential Riccati equations:**
  Incorporate inexactness and Galerkin ideas in LR-DRE solvers of Rosenbrock and BDF type.

Many thanks for your attention.

and H. Mena)

$$E^T \dot{X} E = -C^T C - A^T X E - E^T X A + E^T X B B^T X E = -\mathfrak{R}(X),$$
$$X(t_f) = X_{t_f}.$$

- BDF methods solve AREs in each time step.
- Rosenbrock methods solve Lyapunov equation in each time step.
- Can apply projection ideas as above!
- Can we do so in the ODE solvers themselves?

An LRCF-QADI Projection Method

Newton-Kleinman-ADI vs. QADI

QADI

[Wong/Balakrishnan '04-'07]

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

An LRCF-QADI Projection Method

Newton-Kleinman-ADI vs. QADI

QADI

[Wong/Balakrishnan '04-'07]

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

Newton-Kleinman-ADI

[Kleinman '68]

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

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]

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

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]

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

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

The LRCF-QADI Iteration

Idea [Benner/S. '09]

Apply the Gau-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

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

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

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: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: Compute $Z^{(k)}$ using (LRCF-ADI) or (LRCF-ADI-GP), such that $F^{(k)} Z^{(k)} Z^{(k)H} + Z^{(k)} Z^{(k)H} F^{(k)\,T} \approx -G^{(k)} G^{(k)\,T}$.
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: $G = \begin{bmatrix} C^T & K^{(0)} \end{bmatrix}$
3: **for** $k = 1, 2, \ldots, k_{max}$ **do**
4: $\quad F^{(k)} = A^T - K^{(k-1)} B^T$
5: $\quad$ For $\tilde{V}$ solve $(F^{(k)} + p_k I)\tilde{V} = V_{k-1}$.
6: $\quad V_k = \sqrt{\operatorname{Re}(p_k)/\operatorname{Re}(p_{k-1})}\left(V_{k-1} - (p_k + \overline{p_{k-1}})\tilde{V}\right)$
7: $\quad Z_k = [Z_{k-1} \;\; V_k]$
8: $\quad K^{(k)} = K^{(k-1)} + V_k V_k^T B$
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: $\quad F^{(k)} = A^T - K^{(k-1)} B^T$
5: $\quad$ For $\tilde{V}$ solve $(F^{(k)} + p_k I)\tilde{V} = V_{k-1}$.
6: $\quad V_k = \sqrt{\operatorname{Re}(p_k)/\operatorname{Re}(p_{k-1})}\left(V_{k-1} - (p_k + \overline{p_{k-1}})\tilde{V}\right)$
7: $\quad Z_k = [Z_{k-1} \;\; V_k]$
8: $\quad$ Project ARE, solve and prolongate solution.
9: $\quad K^{(k)} = Z^{(k)}(Z^{(k)\,H} B)$
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

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]

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

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]

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

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]

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

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

The RicADI Projection Method

Invariance of the Krylov Subspaces

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

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

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]

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

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

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

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

The RicADI Projection Method

RicADI Algorithm

Algorithm 10 Low-rank Cholesky factor RicADI iteration

(LRCF-RicADI) [Benner/Kohler/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)

Input: A, B, C, $K^{(0)}$ fo
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)opt ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \ldots$ with respect to the matrix F.
2: For V_1 solve $(F + p_1 I)V_1 = \sqrt{-2\,\mathrm{Re}\,(p_1)}\,C$,
3: $Z_1 = V_1$, $i = 2$
4: **repeat**
5: For $\tilde{V}$ solve $(F + p_{i-1} I)\tilde{V} = V_{i-1}$
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** $\|C^TC + A^TZ_iZ_i^T + Z_iZ_i^TA - Z_iZ_i^TBB^TZ_iZ_i^T\| \leq TOL$

[Benner/Kohler/S. '11]

Compute projection basis via $Q_i = Z_i L_i$, where

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

and

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

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/Kher/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 $C^T C + A^T X E + E^T X A - E^T X B B^T X E = 0$.

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 $(F + p_1 E)\, V_1 = \sqrt{-2\,\mathrm{Re}\,(p_1)}\, G$,
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: $V_i = \sqrt{\mathrm{Re}\,(p_i)/\,\mathrm{Re}\,(p_{i-1})}\left(V_{i-1} - (p_i + \overline{p_{i-1}})\tilde{V}\right)$
7: $Z_i = [Z_{i-1} \;\; V_i]$, $i = i + 1$
8: Project ARE, solve and prolongate solution.
9: **until** $\|C^T C + A^T Z_i Z_i^T E + E^T Z_i Z_i^T A - E^T Z_i Z_i^T B B^T Z_i Z_i^T E\| \leq TOL$

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

Test Examples and Hardware

	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

Test Examples and Hardware

	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