August 23, 2011

Acceleration of Newton-based Methods for Solving Large Sparse Algebraic Riccati Equations

Jens Saak
joint work with
Peter Benner and Martin Köhler

Young Researchers Minisymposium **YR4**:
Numerical Methods for the Solution of Algebraic Riccati Equations

MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

Outline

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

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 $Z \in \mathbb{K}^{n,nz}$, such that $nz \ll n$ 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 $\{p_1, ..., p_J\} \subset \mathbb{C}_{<0}$.

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

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 $Z \in \mathbb{K}^{n,nz}$, such that $nz \ll n$ 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 $\{p_1, ..., p_J\} \subset \mathbb{C}_{<0}$.

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

Task Find $Z \in \mathbb{K}^{n,nz}$, such that $nz \ll n$ 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 $\{p_1, ..., p_J\} \subset \mathbb{C}_{<0}$.

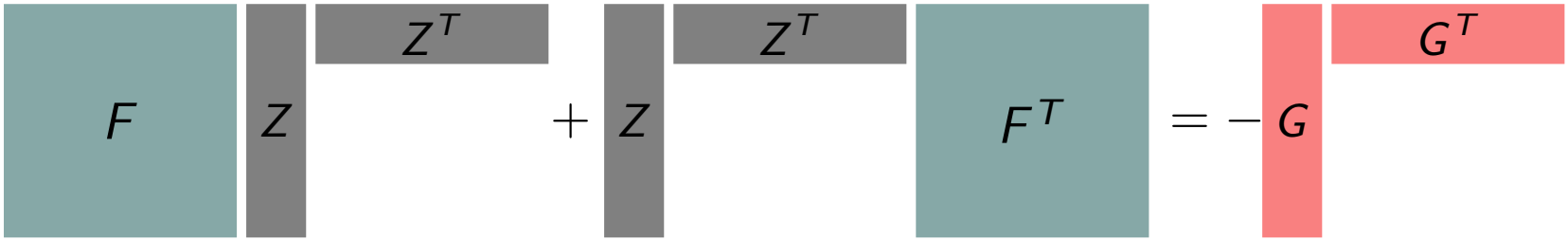
Stop if

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

Can ensure $Z \in \mathbb{R}^{n,nz}$ even if $\{p_1, ..., p_J\} \not\subset \mathbb{R}$ [Benner/Kürschner/S. '11]

Solving Large Lyapunov Equations

LRCF-ADI with Galerkin-Projection-Acceleration



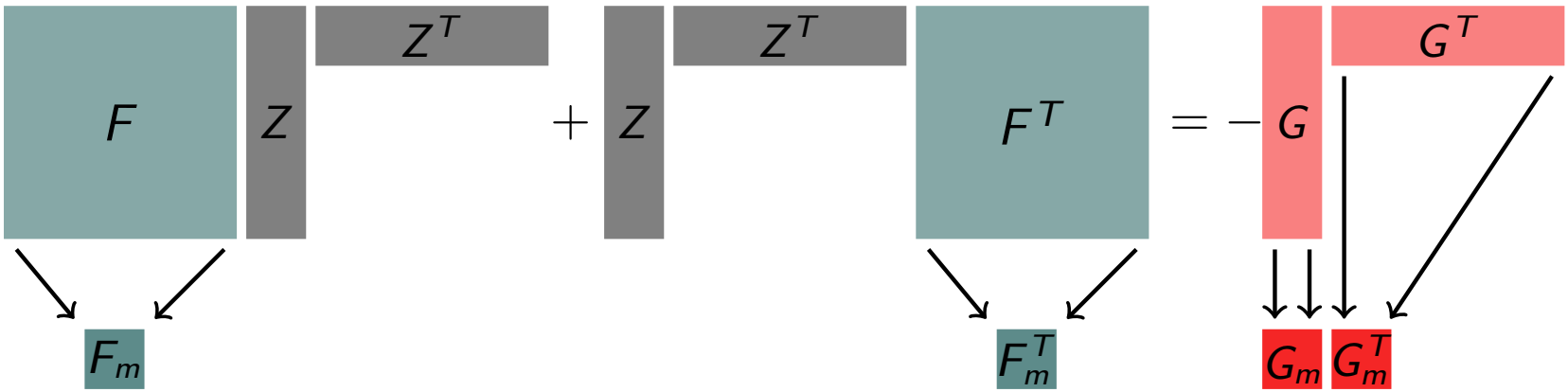
Legend:

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

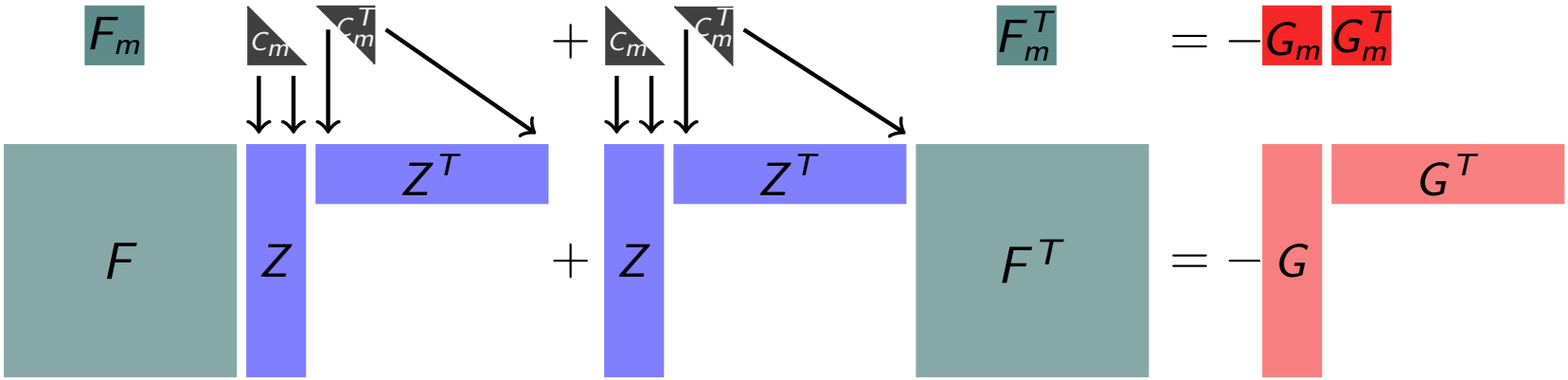
Legend:

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 $\rightarrow$ LRCF-ADI-GP [S.'09, Benner/S.'10]

1. Compute the LRCF-ADI iterate Z_i
2. Compute orthogonal basis via QR factorization: $Q_i R_i \Pi_i = Z_i$
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

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 $(Z_i^T Z_i)^{-1} = L_i L_i^T$

Projected ADI Cross Approx. LRCF-ADI-GP [S.'09, Benner/S.'10]

1. Compute the LRCF-Blocks 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 →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: $\|A\|_2 \geq \sqrt{\|M\|_2}\|A\|_E$

LRCF-NM for 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

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

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 [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

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

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

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 [Kleinman '68]

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

LRCF-NM for the ARE

Newton's Method for AREs

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

Newton's Iteration for the ARE

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

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

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

i.e., in every Newton step solve a

Lyapunov Equation [Kleinman '68]

$$F_\ell^T X_{\ell+1} E + E^T X_{\ell+1} F_\ell = -\tilde{G}_\ell \tilde{G}_\ell^T.$$

LRCF-NM for the ARE

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

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

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

Find low rank factor $Z \in \mathbb{R}^{n,nz}$, where $nz \ll n$.

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]

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

- 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]

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

- 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

Newton ADI Variants

Simplified Factored Newton Iteration

- Gradient updates are *cheap*

LRCF-NM for the ARE

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
     (see also QADI discussion)

LRCF-NM for the ARE

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
     (see also QADI discussion)

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

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
     (see also QADI discussion)

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

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
     (see also QADI discussion)

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 [Hylla/S. '08 – '11]

- Control (G-)LRCF-ADI accuracy according to Newton progress.

LRCF-NM for the ARE

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 $F^{(k)} Z^{(k)} Z^{(k)^H} + Z^{(k)} Z^{(k)^H} F^{(k)^T} \approx -G^{(k)} G^{(k)^T}$.
5: $K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$
6: **end for**

LRCF-NM for the ARE

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 $F^{(k)} Z^{(k)} Z^{(k)^H} + Z^{(k)} Z^{(k)^H} F^{(k)^T} \approx -G^{(k)} G^{(k)^T}$.
5: $\quad K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$
6: **end for**

LRCF-NM for the ARE

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 $F^{(k)} Z^{(k)} Z^{(k)^H} + Z^{(k)} Z^{(k)^H} F^{(k)^T} \approx -G^{(k)} G^{(k)^T}$.
5: Project ARE, solve and prolongate solution
6: $K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$
7: **end for**

LRCF-NM for the ARE

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

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 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

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

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: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
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 5 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

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

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

An LRCF-QADI Projection Method

The LRCF-QADI Iteration

Algorithm 5 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: $\quad V_k = \sqrt{\mathrm{Re}\,(p_k)/\,\mathrm{Re}\,(p_{k-1})}\left(V_{k-1} - (p_k + \overline{p_{k-1}})\tilde{V}\right)$
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]

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

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]

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

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]

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

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

The RicADI Projection Method

Invariance of the Krylov Subspaces

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

$$F_\ell = A - BB^T X_\ell E =: A - BK_\ell$$ 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 6 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 6 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: $Z = Z_{i_{max}} \in \mathbb{C}^{n \times i_{max}}$, 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 $(F + p_1 I)V_1 = \sqrt{-2\operatorname{Re}(p_1)}\,C$
3: $Z_1 = V_1$, $i = 2$
4: **repeat**
5: For $\tilde{V}$ solve $(F + p_i I)\tilde{V} = V_{i-1}$
6: $V_i = \sqrt{\operatorname{Re}(p_i)/\operatorname{Re}(p_{i-1})}\left(V_{i-1} - (p_i + p_{i-1})\tilde{V}\right)$
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 6 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 $C^TC + A^TXE + E^TXA - E^TXBB^TXE = 0$.

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 $(F + p_1E)\,V_1 = \sqrt{-2\,\mathrm{Re}\,(p_1)}\,G$
3: $Z_1 = V_1, \quad i = 2$
4: **repeat**
5: For $\tilde{V}$ solve $(F + p_iE)\tilde{V} = EV_{i-1}$
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** $\|C^TC + A^TZ_iZ_i^TE + E^TZ_iZ_i^TA - E^TZ_iZ_i^TBB^TZ_iZ_i^TE\| \leq TOL$

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
- Oberwohlfach 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

Conclusions and Future Perspectives

Conclusions

- LRCF-NM can be accelerated drastically
- Clever projection can help avoiding orthogonalizations

Future Work

- Combine inexact and projected Newton methods,
- Try different inner loop solvers, e.g.,
  [Simoncini '07, Vandereycken '10],
- Integrate these approaches in DRE solvers (with H. Mena)

Conclusions and Future Perspectives

Conclusions

- LRCF-NM can be accelerated drastically
- Clever projection can help avoiding orthogonalizations

Future Work

- Combine inexact and projected Newton methods,
- Try different inner loop solvers, e.g. [Simoncini '07, Vandereycken '10],
- Integrate these approaches in DRE solvers (with H. Mena)

Many thanks for your attention.