Recent Advances and Future Challenges in the ADI Based Numerical Solution of Large-Scale Matrix Equations

Jens Saak

1-day Workshop on Matrix Equations

Braunschweig Oct. 09, 2009

Department of Mathematics
CHEMNITZ UNIVERSITY OF TECHNOLOGY

Outline

Introduction

Introduction
(G-)LRCF-ADI

Consider $$FX + XF^T = -GG^T$$

ADI Iteration for the Lyapunov Equation (LE) [Wachspress '88]

For $j = 1, \ldots, J$

$$\begin{aligned} X_0 &= 0 \\ (F + p_j I)X_{j-\frac{1}{2}} &= -GG^T - X_{j-1}(F^T - p_j I) \\ (F + p_j I)X_j^T &= -GG^T - X_{j-\frac{1}{2}}^T(F^T - p_j I) \end{aligned}$$

Introduction
(G-)LRCF-ADI

Consider $$FX + XF^T = -GG^T$$

ADI Iteration for the Lyapunov Equation (LE) [WACHSPRESS '88]

For $j = 1, \dots, J$

$$\begin{aligned} X_0 &= 0 \\ (F + p_j I)X_{j-\frac{1}{2}} &= -GG^T - X_{j-1}(F^T - p_j I) \\ (F + p_j I)X_j^T &= -GG^T - X_{j-\frac{1}{2}}^T(F^T - p_j I) \end{aligned}$$

Rewrite as one step iteration and factorize $X_i = Z_i Z_i^T$, $i = 0, \dots, J$

$$\begin{aligned} Z_0 Z_0^T &= 0 \\ Z_j Z_j^T &= -2p_j(F + p_j I)^{-1} G G^T (F + p_j I)^{-T} \\ &\quad + (F + p_j I)^{-1}(F - p_j I) Z_{j-1} Z_{j-1}^T (F - p_j I)^T (F + p_j I)^{-T} \end{aligned}$$

Introduction
(G-)LRCF-ADI

$$Z_j = [\sqrt{-2p_j}(F + p_j I)^{-1}G,\ (F + p_j I)^{-1}(F - p_j I)Z_{j-1}]$$

[Penzl '00]

Introduction
(G-)LRCF-ADI

$$Z_j = [\sqrt{-2p_j}(F + p_j I)^{-1}G,\ (F + p_j I)^{-1}(F - p_j I)Z_{j-1}]$$

[Penzl '00]

Observing that $(F - p_i I)$, $(F + p_k I)^{-1}$ commute, we rewrite Z_J as

$$Z_J = [z_J,\ P_{J-1}z_J,\ P_{J-2}(P_{J-1}z_J),\ \dots,\ P_1(P_2 \cdots P_{J-1}z_J)],$$

[J. R. Li/White '02]

where

$$z_J = \sqrt{-2p_J}(F + p_J I)^{-1}G$$

and

$$P_i := \frac{\sqrt{-2p_i}}{\sqrt{-2p_{i+1}}}\left[I - (p_i + p_{i+1})(F + p_i I)^{-1}\right].$$

Introduction
(G-)LRCF-ADI

Algorithm 1 Low-rank Cholesky factor ADI iteration (LRCF-ADI) [Penzl '97/'00, J. R. Li/White '99/'02, Benner/J. R. Li/Penzl '99/'08]

Input: F, G defining $FX + XF^T = -GG^T$ and shifts $\{p_1, \ldots, p_{i_{max}}\}$
Output: $Z = Z_{i_{max}} \in \mathbb{C}^{n \times t_{i_{max}}}$, such that $ZZ^H \approx X$

1: $V_1 = \sqrt{-2\,\mathrm{Re}\,(p_1)}(F + p_1 I)^{-1} G$
2: $Z_1 = V_1$
3: **for** $i = 2, 3, \ldots, i_{max}$ **do**
4: $\quad V_i = \sqrt{\mathrm{Re}\,(p_i)/\,\mathrm{Re}\,(p_{i-1})}(V_{i-1} - (p_i + \overline{p_{i-1}})(F + p_i I)^{-1} V_{i-1})$
5: $\quad Z_i = [Z_{i-1} \;\; V_i]$
6: **end for**

Introduction
(G-)LRCF-ADI

Algorithm 1 Low-rank Cholesky factor ADI iteration (LRCF-ADI) [Penzl '97/'00, J. R. Li/White '99/'02, Benner/J. R. Li/Penzl '99/'08]

Input: F, G defining $FX + XF^T = -GG^T$ and shifts $\{p_1, \ldots, p_{i_{max}}\}$
Output: $Z = Z_{i_{max}} \in \mathbb{C}^{n \times t_{i_{max}}}$ such that $ZZ^H \approx X$

$V_1 = \sqrt{-2\,\mathrm{Re}\,(p_1)}(F + p_1 I)^{-1}G$
$Z_1 = V_1$
for $i = 2, 3, \ldots, i_{max}$ **do**
  $V_i = \sqrt{\mathrm{Re}\,(p_i)/\,\mathrm{Re}\,(p_{i-1})}(V_{i-1} - (p_i + \overline{p_{i-1}})(F + p_i I)^{-1}V_{i-1})$
  $Z_i = [Z_{i-1} \;\; V_i]$
end for

earlier contributions

- choice and computation of shift parameters
- column compression via RRQR (safe space, avoid rank deficiency)

Introduction
(G-)LRCF-ADI

Algorithm 2 Generalized Low-rank Cholesky factor ADI iteration (G-LRCF-ADI) [Benner '04, Benner/S. '09, S. '09]

Input: E, F, G defining $FXE^T + EXF^T = -GG^T$ and shifts $\{p_1, \ldots, p_{i_{max}}\}$
Output: $Z = Z_{i_{max}} \in \mathbb{C}^{n \times t_{i_{max}}}$, such that $ZZ^H \approx X$

1: $V_1 = \sqrt{-2\,\mathrm{Re}\,(p_1)}(F + p_1 E)^{-1} G$
2: $Z_1 = V_1$
3: **for** $i = 2, 3, \ldots, i_{max}$ **do**
4: $\quad V_i = \sqrt{\mathrm{Re}\,(p_i)/\,\mathrm{Re}\,(p_{i-1})}(V_{i-1} - (p_i + \overline{p_{i-1}})(F + p_i E)^{-1} E V_{i-1})$
5: $\quad Z_i = [Z_{i-1} \;\; V_i]$
6: **end for**

Introduction
Numerical Tests with LRCF-ADI

Convection-Diffusion Equation

- FDM for 2D convection-diffusion equations on $[0,1]^2$
- LyaPack benchmark, q = p = 1, e.g., `demo_l1`
- non-symmetric $A \in \mathbb{R}^{n \times n}$, $n = 10^2 - 10^6$
- 16 heuristic Penzl shifts from 50/25 Ritz/harmonic Ritz values
- on Dual-CPU Dual-Core Xeon 5160 3.0 GHz, 16GB RAM (`romulus`)

Software:

- LyaPack Penzl's well known MATLAB toolbox
- M.E.S.S. our successor to LyaPack
- C.M.E.S.S. the upcoming C language version of M.E.S.S.

Introduction

Numerical Tests with LRCF-ADI

Computation times (in seconds) for the three different implementations of Algorithm 1

N	LyaPack	M.E.S.S.	C.M.E.S.S.
100	0.12	0.16	0.02
625	0.10	0.23	0.04
2 500	0.70	0.99	0.16
10 000	6.22	5.64	0.97
40 000	71.48	34.55	11.09
90 000	418.50	90.49	34.67
160 000	out of mem.	219.90	109.32
250 000	out of mem.	403.80	193.67
562 500	out of mem.	1 216.70	930.14
1 000 000	out of mem.	2 428.60	2 219.95

Computations by Martin Köhler

Introduction

Numerical Tests with LRCF-ADI

Computation times (in seconds) for the three different implementations of Algorithm 1

N	LyaPack	M.E.S.S.	C.M.E.S.S.
100	0.12	0.16	0.02
625	0.10	0.23	0.04
2 500	0.70	0.99	0.16
10 000	6.22	5.64	[illegible].97
40 000	71.48	34.55	11.09
90 000	418.50	90.49	34.67
160 000	out of mem.	219.90	109.32
250 000	out of mem.	403.80	193.67
562 500	out of mem.	1 216.70	930.14
1 000 000	out of mem.	2 428.60	2 219.95

C.M.E.S.S.

speed optimization:
2 219.95s on `romulus` (16GB RAM)

minimum memory variant:
(using **cache to disk** techniques)
5 962.19s and **only 1.8GB RAM** consumption.

Computations by Martin Köhler

Introduction

Krylov Subspace Based Solvers for Lyapunov Equations

Consider Schur/singular value decomposition $X = U\Sigma U^T$,
$U \in \mathbb{R}^{n\times n}$, $U^T U = I$, $\Sigma = \mathrm{diag}\,(\sigma_1, \ldots, \sigma_n)$ and $|\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_n|$.
The best rank-m Frobenius-norm approximation to X is thus given by

$$X_m := U \begin{bmatrix} \Sigma_m & 0 \\ 0 & 0 \end{bmatrix} U^T = U_m \Sigma_m U_m^T.$$

Introduction
Krylov Subspace Based Solvers for Lyapunov Equations

Consider Schur/singular value decomposition $X = U\Sigma U^T$,
$U \in \mathbb{R}^{n\times n}$, $U^TU = I$, $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ and $|\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_n|$.
The best rank-m Frobenius-norm approximation to X is thus given by

$$X_m := U \begin{bmatrix} \Sigma_m & 0 \\ 0 & 0 \end{bmatrix} U^T = U_m \Sigma_m U_m^T.$$

Krylov Projection Idea [Saad '90, Jaimoukha/Kasenally '94]

Solve

$$(U_m^T F U_m) Y_m + Y_m (U_m^T F^T U_m) = -U_m^T G G^T U_m, \qquad (1)$$

on *colspan*(U_m) and get X_m as

$$X_m = U_m Y_m U_m^T.$$

Introduction
Krylov Subspace Based Solvers for Lyapunov Equations

Consider Schur/singular value decomposition $X = U \Sigma U^T$,
$U \in \mathbb{R}^{n \times n}$, $U^T U = I$, $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ and $|\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_n|$.
The best rank-m Frobenius-norm approximation to X is thus given by

$$X_m := U \begin{bmatrix} \Sigma_m & 0 \\ 0 & 0 \end{bmatrix} U^T = U_m \Sigma_m U_m^T.$$

Krylov Projection Idea [Saad '90, Jaimoukha/Kasenally '94]

Solve

$$F_m Y_m + Y_m F_m^T = G_m G_m^T, \qquad F_m \in \mathbb{R}^{m \times m},\ m \ll n \tag{1}$$

on *colspan*(U_m) and get X_m as

$$X_m = U_m Y_m U_m^T.$$

Introduction

Krylov Subspace Based Solvers for Lyapunov Equations

Consider Schur/singular value decomposition $X = U\Sigma U^T$,
$U \in \mathbb{R}^{n\times n}$, $U^T U = I$, $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ and $|\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_n|$.
The best rank-m Frobenius-norm approximation to X is thus given by

$$X_m := U \begin{bmatrix} \Sigma_m & 0 \\ 0 & 0 \end{bmatrix} U^T = U_m \Sigma_m U_m^T.$$

Krylov Projection Idea [Saad '90, Jaimoukha/Kasenally '94]

Solve

$$F_m Y_m + Y_m F_m^T = G_m G_m^T, \qquad F_m \in \mathbb{R}^{m\times m},\ m \ll n \tag{1}$$

on *colspan*(U_m) and get X_m as

$$X_m = U_m C_m C_m^T U_m^T \quad \text{where } Y_m = C_m C_m^T.$$

Outline

1. Introduction
2. (G-)LRCF-ADI with Galerkin-Projection-Acceleration
   - ADI and Rational Krylov
   - Projected ADI Step
3. LRCF-NM for the ARE
4. Future Challenges

(G-)LRCF-ADI with Galerkin-Projection-Acceleration

ADI and Rational Krylov

ADI space is a certain rational Krylov subspace $\mathcal{L}(F, G, \mathbf{p})$

[J. R. Li '00, J. R. Li/White '02]

(G-)LRCF-ADI with Galerkin-Projection-Acceleration

ADI and Rational Krylov

ADI space is a certain rational Krylov subspace $\mathcal{L}(F, G, \mathbf{p})$

[J. R. Li '00, J. R. Li/White '02]

Idea [Benner/R. C. Li/Truhar '08, S. '09]

Solve on current subspace of $\mathcal{L}(F, G, \mathbf{p})$ in the ADI step to increase the quality of the iterate.

(G-)LRCF-ADI with Galerkin-Projection-Acceleration
ADI and Rational Krylov

ADI space is a certain rational Krylov subspace $\mathcal{L}(F, G, \mathbf{p})$

[J. R. Li '00, J. R. Li/White '02]

Idea [Benner/R. C. Li/Truhar '08, S. '09]

Solve on current subspace of $\mathcal{L}(F, G, \mathbf{p})$ in the ADI step to increase the quality of the iterate.

Similar approach: ADI-preconditioned global Arnoldi method

[Jbilou '08]

(G-)LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step

Projected ADI Step → LRCF-ADI-GP

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

(G-)LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step

Projected ADI Step → LRCF-ADI-GP

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

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

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

- Additionally needs the standard (Krylov) assumption: $F + F^T < 0$
- orthogonalization (QR) can be avoided at the cost of solving a generalized projected LE:
$(Z_i^T F Z_i)\,\tilde{Z}\tilde{Z}^T\, Z_i^T Z_i + Z_i^T Z_i\, \tilde{Z}\tilde{Z}^T\,(Z_i^T F^T Z_i) = -Z_i^T G G^T Z_i$

(G-)LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step

Projected ADI Step →G-LRCF-ADI-GP

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

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

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

(G-)LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step

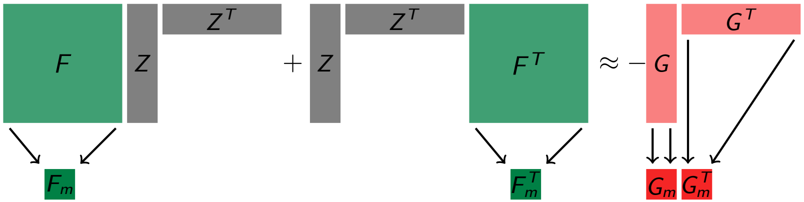
$$F Z Z^T + Z Z^T F^T \approx -G G^T$$

Legend:

new factor original matrix projected matrix projected Cholesky factor
old factor original rhs projected rhs

(G-)LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step



Legend:

new factor original matrix projected matrix projected Cholesky factor
old factor original rhs projected rhs

(G-)LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step

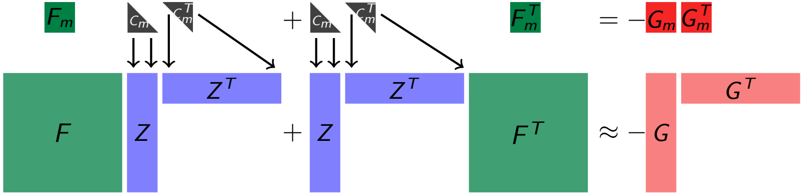
$$F_m \quad C_m C_m^T \quad + C_m C_m^T \quad F_m^T \quad = -G_m G_m^T$$

Legend:

new factor original matrix projected matrix projected Cholesky factor
old factor original rhs projected rhs

(G-)LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step



Legend:

new factor original matrix projected matrix projected Cholesky factor
old factor original rhs projected rhs

LRCF-NM for the ARE

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

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

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

Newton's Iteration for the ARE

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

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

$$\mathfrak{R}'|_X : \quad Q \mapsto (A - BB^TX)^TQ + Q(A - BB^TX),$$

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

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

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/J. R. 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

LRCF-NM for the ARE

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

Factored Newton-Kleinman Iteration [Benner/J. R. 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

- 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/J. R. 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

- 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/J. R. 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/J. R. 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/J. R. 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

$$(A - BK_\ell + p_k^{(\ell)} E)^{-1} = (I_n + (A + p_k^{(\ell)} E)^{-1} B (I_m - K_\ell (A + p_k^{(\ell)} E)^{-1} B)^{-1} K_\ell)(A + p_k^{(\ell)} E)^{-1}$$

LRCF-NM for the ARE

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

Factored Newton-Kleinman Iteration [Benner/J. R. 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

$$(A - BK_\ell + p_k^{(\ell)} E)^{-1} = (I_n + (A + p_k^{(\ell)} E)^{-1} B (I_m - K_\ell (A + p_k^{(\ell)} E)^{-1} B)^{-1} K_\ell)(A + p_k^{(\ell)} E)^{-1}$$

Factored Newton-Galerkin Iteration [S. '09]

- apply (G-)LRCF-ADI-GP in every Newton step (ADI loop)
- if necessary add Galerkin projection for ARE (Newton loop)

LRCF-NM for the ARE

Test Examples

Example 1: Heat Equation

- FDM for 2D heat equations on $[0, 1]^2$
- LyaPack benchmarks, q = p = 1, e.g., `demo_r1`
- symmetric $A \in \mathbb{R}^{n \times n}$, $n = 10\,000$

Example 2: Convection-Diffusion Equation

- FDM for 2D convection-diffusion equations on $[0, 1]^2$
- LyaPack benchmark, q = p = 1, e.g., `demo_l1`
- non-symmetric $A \in \mathbb{R}^{n \times n}$, $n = 10\,000$.

- shift parameters: Penzl's heuristic from 50/25 Ritz/harmonic Ritz values of A
- test system: Intel Core 2 Quad Q9400 2.66 GHz; 4 GB RAM; 64Bit-MATLAB (R2009a) using threaded BLAS

LRCF-NM for the ARE

Test Results: Heat Equation

Newton-ADI

NWT	rel. change	rel. residual	ADI
1	1	9.99e-01	200
2	9.99e-01	3.41e+01	23
3	5.25e-01	6.37e+00	20
4	5.37e-01	1.52e+00	20
5	7.03e-01	2.64e-01	23
6	5.57e-01	1.56e-02	23
7	6.59e-02	6.30e-05	23
8	4.02e-04	9.68e-10	23
9	8.45e-09	1.09e-11	23
10	1.52e-14	1.09e-11	23

CPU time: 76.9 sec.

Newton-Galerkin-ADI

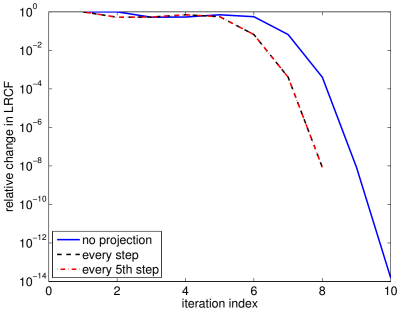
NWT	rel. change	rel. residual	ADI
1	1	3.56e-04	20
2	5.25e-01	6.37e+00	10
3	5.37e-01	1.52e+00	6
4	7.03e-01	2.64e-01	10
5	5.57e-01	1.57e-02	10
6	6.59e-02	6.30e-05	10
7	4.03e-04	9.79e-10	10
8	8.45e-09	1.43e-15	10

CPU time: 38.0 sec.

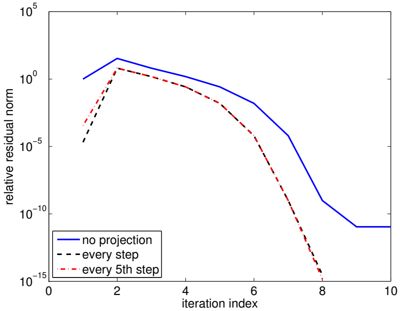
LRCF-NM for the ARE

Test Results: Heat Equation

Relative Change of LRCF



Relative Residual Norm



LRCF-NM for the ARE

Test Results: Convection-Diffusion Equation

Newton-ADI

NWT	rel. change	rel. residual	ADI
1	1	9.99e-01	200
2	9.99e-01	3.56e+01	60
3	3.11e-01	3.72e+00	39
4	2.88e-01	9.62e-01	40
5	3.41e-01	1.68e-01	45
6	1.22e-01	5.25e-03	42
7	3.88e-03	2.96e-06	47
8	2.30e-06	6.09e-13	47

CPU time: 185.9 sec.

Newton-Galerkin-ADI

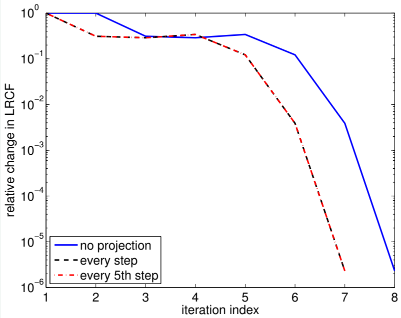
NWT	rel. change	rel. residual	ADI
1	1	1.78e-02	35
2	3.11e-01	3.72e+00	15
3	2.88e-01	9.62e-01	20
4	3.41e-01	1.68e-01	15
5	1.22e-01	5.25e-03	20
6	3.89e-03	2.96e-06	15
7	2.30e-06	6.14e-13	20

CPU time: 75.7 sec.

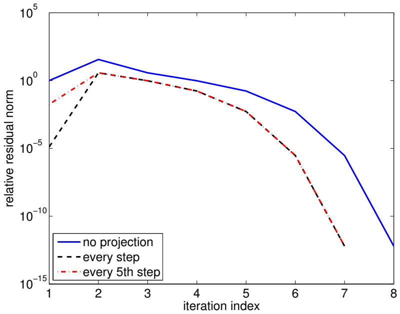
LRCF-NM for the ARE

Test Results: Convection-Diffusion Equation

Relative Change of LRCF



Relative Residual Norm



Future Challenges

Future Challenges
Newton-Kleinman-ADI vs. QADI

QADI [Wong/Balakrishnan '04-'07]

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

Future Challenges

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

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

Future Challenges
Newton-Kleinman-ADI vs. QADI

QADI [Wong/Balakrishnan '04-'07]

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

Newton-Kleinman-ADI

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

Future Challenges
Newton-Kleinman-ADI vs. QADI

QADI [Wong/Balakrishnan '04-'07]

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

Newton-Kleinman-ADI

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

Future Challenges
Newton-Kleinman-ADI vs. QADI

QADI [Wong/Balakrishnan '04-'07]

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

Newton-Kleinman-ADI

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

Future Challenges
Facing the MultiCore Challenge

Parallelization Capabilities

- threaded BLAS in shift parameter computation
- precomputation of shifted LU-decompositions
- forward-backward-solves for multiple right hand sides
- evaluation of stopping criteria

Future Challenges

Facing the MultiCore Challenge

Parallelization Capabilities

- threaded BLAS in shift parameter computation
- precomputation of shifted LU-decompositions
- forward-backward-solves for multiple right hand sides
- evaluation of stopping criteria

Properties of MultiCore Machines

Shared memory parallelization:

- 👍 no communication overhead
- 👎 shared cache hierarchy (e.g., Intel Core2)
- 👎 block divided main memory (e.g., AMD Opteron)

Future Challenges

Facing the MultiCore Challenge

Convection-diffusion example: QuadCore-OpenMP-Tests solver runtime

size	arch	4 threads	1 thread	speedup
100	Opteron	0.03765	0.05918	**1.57**
	Xeon	0.02776	0.02988	1.08
625	Opteron	0.12871	0.13237	1.03
	Xeon	0.07064	0.09797	**1.39**
2500	Opteron	1.06863	0.94622	0.89
	Xeon	0.50271	0.49864	**0.99**
10000	Opteron	6.67419	7.41584	1.11
	Xeon	4.92528	5.63425	**1.14**
40000	Opteron	47.8563	51.7201	1.08
	Xeon	31.8747	36.9881	**1.16**
90000	Opteron	142.967	175.217	**1.23**
	Xeon	98.1336	117.188	1.19
160000	Opteron	311.684	404.672	**1.30**
	Xeon	254.479	309.834	1.22
250000	Opteron	1 005.77	1 148.19	**1.14**
	Xeon	856.801	972.577	1.14
562500	Opteron	1 624.73	2 335.74	**1.44**
	Xeon	1 537.56	2 066.44	1.34
1000000	Opteron	8 453.80	9 856.93	**1.17**
	Xeon	7 265.37	7 383.35	1.02

Future Challenges
Facing the MultiCore Challenge

Convection-diffusion example: QuadCore-OpenMP-Tests LU decompositions

size	arch	1st-LU	MLU-4t	MLU-1t	speedup
100	Opteron	0.00049	0.00163	0.00138	0.85
	Xeon	0.00024	0.00065	0.00062	**0.96**
625	Opteron	0.00191	0.00559	0.00803	1.44
	Xeon	0.00121	0.00276	0.00781	**2.83**
2500	Opteron	0.01650	0.05439	0.07352	1.35
	Xeon	0.00898	0.02319	0.04373	**1.89**
10000	Opteron	0.19212	0.30134	0.78790	2.61
	Xeon	0.09576	0.12311	0.43235	**3.51**
40000	Opteron	2.57802	2.75977	8.52883	**3.09**
	Xeon	1.15189	1.24284	3.74470	3.01
90000	Opteron	12.7149	16.3483	44.6250	**2.73**
	Xeon	5.89398	11.0538	23.5014	2.13
160000	Opteron	43.5361	38.9257	122.114	**3.14**
	Xeon	18.1701	33.5309	76.3129	2.28
250000	Opteron	119.284	70.9077	224.970	**3.17**
	Xeon	41.4773	69.0541	156.055	2.26
562500	Opteron	662.555	297.201	901.092	**3.03**
	Xeon	229.656	291.314	740.166	2.54
1000000	Opteron	2 144.24	873.583	2 757.71	**3.16**
	Xeon	1 327.14	933.001	2 097.40	2.25

Future Challenges
Facing the MultiCore Challenge

Convection-diffusion example: QuadCore-OpenMP-Tests solver no res.

size	arch	4 threads	1 thread	speedup
100	Opteron	0.03133	0.04067	1.30
	Xeon	0.01139	0.01877	**1.65**
625	Opteron	0.04349	0.02667	0.61
	Xeon	0.01802	0.02210	**1.23**
2500	Opteron	0.15400	0.21844	1.42
	Xeon	0.06460	0.11350	**1.76**
10000	Opteron	0.92191	1.80707	1.96
	Xeon	0.47643	1.18958	**2.50**
40000	Opteron	7.73269	16.0154	2.07
	Xeon	3.97827	9.21411	**2.32**
90000	Opteron	37.6280	69.9033	1.86
	Xeon	21.4366	41.1459	**1.92**
16000	Opteron	96.8555	193.791	**2.00**
	Xeon	62.4670	121.631	1.95
250000	Opteron	225.104	389.816	1.73
	Xeon	127.539	239.816	**1.88**
562500	Opteron	1 012.07	1 658.35	1.64
	Xeon	565.639	1 093.80	**1.93**
1000000	Opteron	3 193.15	5 185.13	**1.62**
	Xeon	2 344.59	3 695.96	1.58

Future Challenges

Facing the MultiCore Challenge

Convection-diffusion example: QuadCore-OpenMP-Tests solver no res.

size	arch	4 threads	1 thread	speedup
100	Opteron	0.03133	0.04067	1.30
	Xeon	0.01139	0.01877	**1.65**
625	Opteron	0.04349	0.02667	0.61
	Xeon	0.01802	0.02210	**1.23**
2500	Opteron	0.15400	0.21844	1.42
	Xeon	0.06460	0.11350	**1.76**
10000	Opteron	0.92191	1.80707	1.96
	Xeon	0.47643	1.18958	**2.50**
40000	Opteron	7.73269	16.0154	2.07
	Xeon	3.97827	9.21411	**2.32**
90000	Opteron	37.6280	69.9033	1.86
	Xeon	21.4366	41.1459	**1.92**
160000	Opteron	96.8555	193.791	**2.00**
	Xeon	62.4670	121.631	1.95
250000	Opteron	225.104	389.816	1.73
	Xeon	127.539	239.816	**1.88**
562500	Opteron	[illegible]	[illegible]	[illegible]
	Xeon	565.639	1 093.80	**1.93**
1000000	Opteron	3 193.15	5 185.13	**1.62**
	Xeon	2 344.59	3 695.96	1.58

Thank you for your attention!