A Newton-Galerkin-ADI Method for Large-Scale Algebraic Riccati Equations

Peter Benner Jens Saak

Max-Planck-Institute for Dynamics of Complex Technical Systems
Computational Methods in Systems and Control Theory Group
Magdeburg, Germany

Technische Universität Chemnitz
Fakultät für Mathematik
Mathematik in Industrie und Technik
Chemnitz, Germany

Applied Linear Algebra 2010
GAMM Workshop Applied and Numerical Linear Algebra
Novi Sad, May 27, 2010

Outline

Introduction

Large-Scale Algebraic Lyapunov and Riccati Equations

General form of algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

Introduction

Large-Scale Algebraic Lyapunov and Riccati Equations

General form of algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Introduction

Large-Scale Algebraic Lyapunov and Riccati Equations

General form of algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Typical situation in model reduction and optimal control problems for semi-discretized PDEs:

- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),

Introduction

Large-Scale Algebraic Lyapunov and Riccati Equations

General form of algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Typical situation in model reduction and optimal control problems for semi-discretized PDEs:

- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}S$ for FEM),

Introduction

Large-Scale Algebraic Lyapunov and Riccati Equations

General form of algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Typical situation in model reduction and optimal control problems for semi-discretized PDEs:

- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}S$ for FEM),
- G, W low-rank with $G, W \in \{BB^T, C^T C\}$, where $B \in \mathbb{R}^{n \times m}$, $m \ll n$, $C \in \mathbb{R}^{p \times n}$, $p \ll n$.

Introduction

Large-Scale Algebraic Lyapunov and Riccati Equations

General form of algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n\times n}$ given and $X \in \mathbb{R}^{n\times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Typical situation in model reduction and optimal control problems for semi-discretized PDEs:

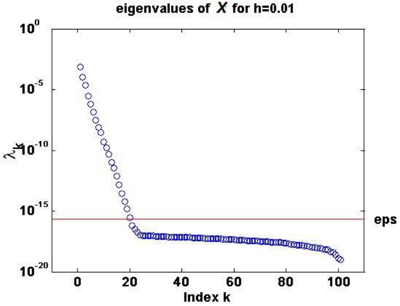
- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}S$ for FEM),
- G, W low-rank with $G, W \in \{BB^T, C^T C\}$, where $B \in \mathbb{R}^{n\times m}$, $m \ll n$, $C \in \mathbb{R}^{p\times n}$, $p \ll n$.
- Standard (eigenproblem-based) $\mathcal{O}(n^3)$ methods are not applicable!

Introduction
Low-Rank Approximation

Consider spectrum of ARE solution (analogous for Lyapunov equations).

Example:

- Linear 1D heat equation with point control,
- $\Omega = [0, 1]$,
- FEM discretization using linear B-splines,
- $h = 1/100 \implies n = 101$.



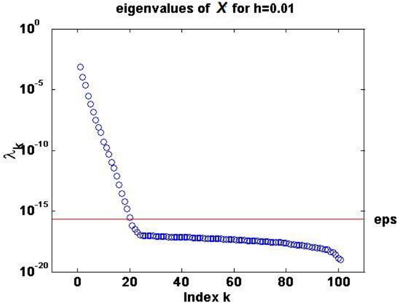
Introduction

Low-Rank Approximation

Consider spectrum of ARE solution (analogous for Lyapunov equations).

Example:

- Linear 1D heat equation with point control,
- $\Omega = [0, 1]$,
- FEM discretization using linear B-splines,
- $h = 1/100 \implies n = 101$.



Idea: $X = X^T \geq 0 \implies$

$$X = ZZ^T = \sum_{k=1}^{n} \lambda_k z_k z_k^T \approx Z^{(r)}(Z^{(r)})^T = \sum_{k=1}^{r} \lambda_k z_k z_k^T.$$

$\implies$ Goal: compute $Z^{(r)} \in \mathbb{R}^{n \times r}$ directly w/o ever forming X!

Introduction

Review: LRCF-ADI for Lyapunov Equations

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

ADI iteration for the Lyapunov equation (LE) [WACHSPRESS '95]

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

Review: LRCF-ADI for Lyapunov Equations

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

ADI iteration for the Lyapunov equation (LE) [Wachspress '95]

For $j = 1, \ldots, J$

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

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

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

Introduction

Review: LRCF-ADI for Lyapunov Equations

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

[Penzl '00]

Introduction

Review: LRCF-ADI for Lyapunov Equations

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

[Penzl '00]

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

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

[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

Review: LRCF-ADI for Lyapunov Equations

Algorithm 1 Low-rank Cholesky factor ADI iteration (LRCF-ADI)
[Penzl '97/'00, Li/White '99/'02, B./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 $ZZ^H \approx X$

```
For V_1 solve (F + p_1 I) V_1 = sqrt(-2 Re(p_1)) G
Z_1 = V_1
for i = 2, 3, ..., i_max do
    For Ṽ solve (F + p_i I) Ṽ = V_{i-1}
    V_i = sqrt(Re(p_i) / Re(p_{i-1})) (V_{i-1} - (p_i + conj(p_{i-1})) Ṽ)
    Z_i = [Z_{i-1}  V_i]
end for
```

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

Introduction

Review: LRCF-ADI for Lyapunov Equations

Algorithm 1 General. Low-rank Cholesky factor ADI iteration (G-LRCF-ADI)
[B. '04, B./Saak '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 $ZZ^H \approx X$

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

Introduction

Krylov Subspace Based Solvers for Lyapunov Equations

Consider Schur/singular value decomposition $X = U\Sigma U^T$,
$U \in \mathbb{R}^{n\times n}$, $U^T U = I$, $\Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_n)$ 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 $X = U\Sigma U^T$,
$U \in \mathbb{R}^{n\times n}$, $U^T U = I$, $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ 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

$$(U_m^T F U_m) Y_m + Y_m (U_m^T F^T U_m) = -U_m^T G G^T U_m, \tag{1}$$

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

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

Note that a factorization

$$Z_m Z_m^T = X_m$$

can easily be computed from a Cholesky factorization of

$$Y_m = \tilde{Z}_m \tilde{Z}_m^T$$

as

$$Z_m = U_m \tilde{Z}_m.$$

Introduction

Krylov Subspace Based Solvers for Lyapunov Equations

Algorithm 2 Basic Krylov Subspace Method for the Lyapunov Equation

Input: F,G defining $FX + XF^T = -GG^T$, an initial Krylov subspace $\mathcal{V}$, e.g., $\mathcal{V} = \mathcal{K}_p(F, G)$ with orthogonal basis $V \in \mathbb{C}^{n \times p}$.

Output: $Z \in \mathbb{C}^{n \times t}$, such that $ZZ^H \approx X$

repeat

  if not first step **then**

    increase dimension of $\mathcal{V}$ and update V.

  end if

  Solve the "small" LE for $\tilde{Z}$ with a classical solver:

$$(V^T FV)\tilde{Z}\tilde{Z}^T + \tilde{Z}\tilde{Z}^T(V^T F^T V) = -V^T GG^T V,$$

  Lift $\tilde{Z}$ to the full space: $Z = U\tilde{Z}$

until res(Z)$<$ TOL

LRCF-ADI with Galerkin-Projection-Acceleration

ADI and Rational Krylov

[Li '00; Theorem 2] interprets the column span of the ADI solution as a certain rational Krylov subspace

$$\mathcal{L}(F,G,\mathbf{p}) := span\left\{ \ldots, \prod_{i=-j}^{-1}(F+p_iI)^{-1}G, \ldots, (F+p_{-2}I)^{-1}(F+p_{-1}I)^{-1}G, (F+p_{-1}I)^{-1}G,\ G,\ (F+p_1I)G, (F+p_2I)(F+p_1I)G, \ldots, \prod_{i=1}^{j}(F+p_iI)G \ \ldots \right\}$$

LRCF-ADI with Galerkin-Projection-Acceleration

ADI and Rational Krylov

[Li '00; Theorem 2] interprets the column span of the ADI solution as a certain rational Krylov subspace

$$\mathcal{L}(F,G,\mathbf{p}) := span\left\{ \dots, \prod_{i=-j}^{-1}(F+p_iI)^{-1}G, \dots, (F+p_{-2}I)^{-1}(F+p_{-1}I)^{-1}G, \right.$$
$$(F+p_{-1}I)^{-1}G,\ G,\ (F+p_1I)G,$$
$$\left. (F+p_2I)(F+p_1I)G, \dots, \prod_{i=1}^{j}(F+p_iI)G \dots \right\}$$

Idea

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

LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step

Projected ADI Step $\longrightarrow$ LRCF-ADI-GP [B./Li/Truhar'09, Saak'09, B./Saak'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 rank deficient.

LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step

Projected ADI Step → LRCF-ADI-GP [B./Li/Truhar'09, Saak'09, B./Saak'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 $Z_i := Q_i \tilde{Z}$

- Need to ensure that projected systems remain stable, e.g., $F + F^T < 0$
- may perform projected ADI step only every k-th step (e.g. $k = 5$) $\rightsquigarrow$ restarted ADI with shifts $\Lambda(Q_i^T F Q_i)$.

LRCF-ADI with Galerkin-Projection-Acceleration

Projected ADI Step

Projected ADI Step →G-LRCF-ADI-GP [B./Li/Truhar'09, Saak'09, B./Saak'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 $Z_i := Q_i \tilde{Z}$

LRCF-ADI with Galerkin-Projection-Acceleration

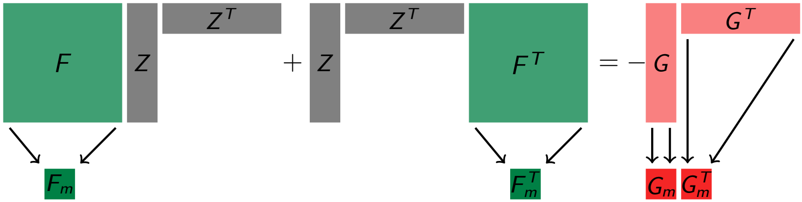
Projected ADI Step

Legend:

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

LRCF-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-ADI with Galerkin-Projection-Acceleration

Projected ADI Step

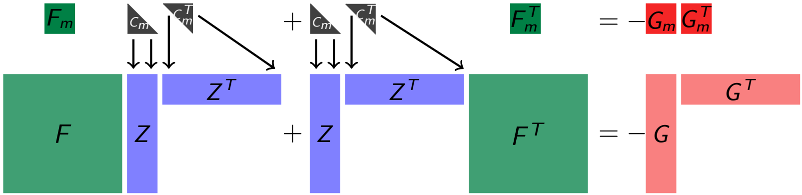
$$F_m \quad C_m C_m^T \quad + C_m C_m^T \quad F_m^T \quad = -G_m G_m^T$$

Legend:

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

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-ADI with Galerkin-Projection-Acceleration

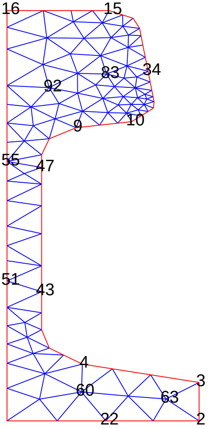
Test Example: Optimal Cooling of Steel Profiles

- Mathematical model: boundary control for linearized 2D heat equation.

$$\begin{aligned} c \cdot \rho \frac{\partial}{\partial t} x &= \lambda \Delta x, & \xi \in \Omega \\ \lambda \frac{\partial}{\partial n} x &= \kappa (u_k - x), & \xi \in \Gamma_k,\ 1 \leq k \leq 7, \\ \frac{\partial}{\partial n} x &= 0, & \xi \in \Gamma_0. \end{aligned}$$

$\Longrightarrow q = 7, p = 6.$

- FEM Discretization, different models for initial mesh ($n = 371$), 1, 2, 3, 4 steps of mesh refinement $\Rightarrow$ $n = 1\,357, 5\,177, 20\,209, 79\,841.$

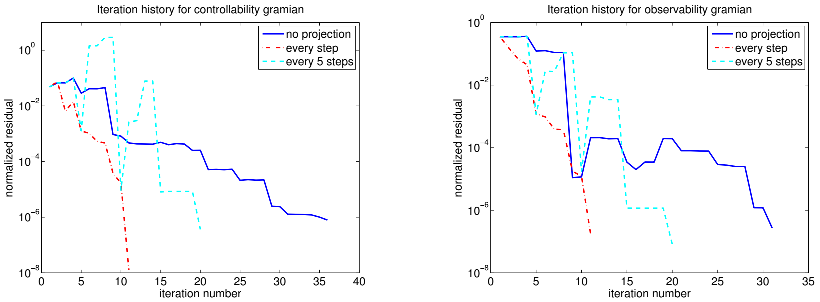


Source: Physical model: courtesy of Mannesmann/Demag.
Math. model: Tröltzsch/Unger '99/'01, Penzl '99, S. '03.

LRCF-ADI with Galerkin-Projection-Acceleration

Numerical Results

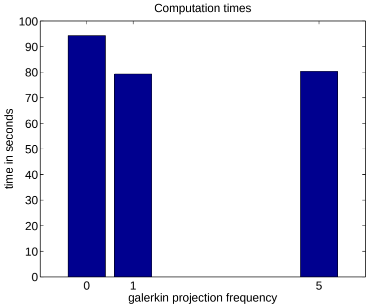
steel profile n=20 209 good shifts



LRCF-ADI with Galerkin-Projection-Acceleration

Numerical Results

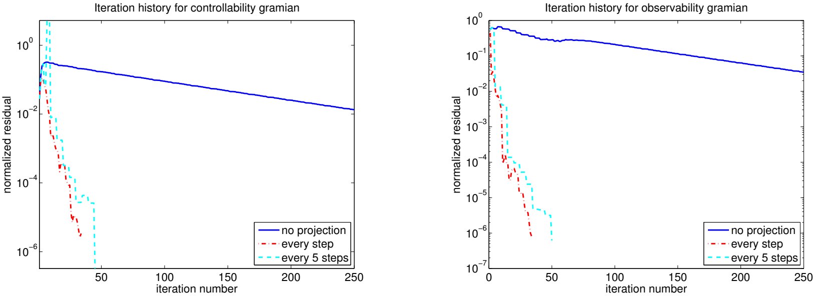
steel profile n=20 209 good shifts



LRCF-ADI with Galerkin-Projection-Acceleration

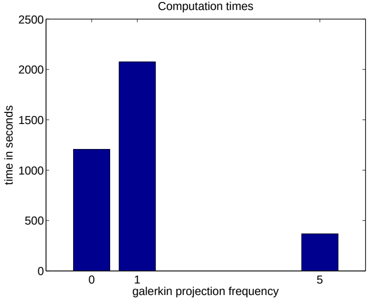
Numerical Results

steel profile n=20 209 bad shifts



LRCF-ADI with Galerkin-Projection-Acceleration
Numerical Results

steel profile n=20 209 bad shifts



LRCF-NM for the ARE

1. Introduction
2. LRCF-ADI with Galerkin-Projection-Acceleration
3. LRCF-NM for the ARE
   - Newton's Method for AREs
   - Low-Rank Newton-ADI (LRCF-NM) for AREs
   - Test Examples
   - Test Results (ADI-loop)
   - Test Results (both-loops)
   - Computation Time Scaling with Problem Size

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

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

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]

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

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

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

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]

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

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]

$F_\ell = A - BB^T X_\ell =: A - BK_\ell$ 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/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

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

LRCF-NM for the ARE

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

Algorithm 3 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 Algorithm 1 (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 3 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

$$C^TC + A^TXE + E^TXA - E^TXBB^TXE = 0.$$

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 $F^{(k)} = A^TE^{-T} - K^{(k-1)}B^TE^{-T}$.
3: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: Compute $Z^{(k)}$ using Algorithm 1 (G-LRCF-ADI) such that

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

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

LRCF-NM for the ARE

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

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

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

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

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

LRCF-NM for the ARE

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

Algorithm 4 Simpl. Low-Rank Cholesky Factor Newton Method (LRCF-NM-S)

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

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

1: Determine (sub)optimal ADI shift parameters $p_1, p_2, \dots$ with respect to the matrix $F^{(k)} = A^T - K^{(0)} B^T$.
2: **for** $k = 1, 2, \dots, k_{max}$ **do**
3: $G^{(k)} = \left[\begin{array}{cc} C^T & K^{(k-1)} \end{array}\right]$
4: Compute $Z^{(k)}$ using Algorithm 1 (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**

LRCF-NM for the ARE

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

Algorithm 5 Low-Rank Cholesky Factor Galerkin-Newton Method (LRCF-NM-GP)

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

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

1: **for** $k = 1, 2, \dots, k_{max}$ **do**
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \dots$ with respect to the matrix $F^{(k)} = A^T - K^{(k-1)}B^T$.
3: $G^{(k)} = \left[\begin{array}{cc} C^T & K^{(k-1)} \end{array}\right]$
4: Compute $Z^{(k)}$ using Algorithm 1 (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: Project ARE, solve and prolongate solution
6: $K^{(k)} = Z^{(k)}(Z^{(k)^H}B)$
7: **end for**

LRCF-NM for the ARE

Test Examples

Example 1: 3d Convection-Diffusion Equation

- FDM for 3D convection-diffusion equation on $[0,1]^3$
- proposed in [Simoncini '07], $q = p = 1$
- non-symmetric $A \in \mathbb{R}^{n \times n}$, $n = 10\,648$

Example 2: 2d Convection-Diffusion Equation

- FDM for 2D convection-diffusion equations on $[0,1]^2$
- LyaPack benchmark, $q = p = 1$, e.g., `demo_l1`
- non-symmetric $A \in \mathbb{R}^{n \times n}$, $n = 22\,500$.

- 16 shift parameters
- Penzl's heuristic from 50/25 Ritz/harmonic Ritz values of A

LRCF-NM for the ARE

Test Results (ADI-loop): Example 1

Newton-ADI

NWT	rel. change	rel. residual	ADI
1	$9.97 \cdot 10^{-01}$	$9.27 \cdot 10^{-01}$	100
2	$3.67 \cdot 10^{-02}$	$9.58 \cdot 10^{-02}$	94
3	$1.36 \cdot 10^{-02}$	$1.09 \cdot 10^{-03}$	98
4	$3.48 \cdot 10^{-04}$	$1.01 \cdot 10^{-07}$	97
5	$6.41 \cdot 10^{-08}$	$1.34 \cdot 10^{-10}$	97
6	**$7.47 \cdot 10^{-16}$**	$1.34 \cdot 10^{-10}$	97

CPU time: 4 805.8 sec.

Newton-Galerkin-ADI

LRCF-ADI-GP(5)

NWT	rel. change	rel. residual	ADI
1	$9.97 \cdot 10^{-01}$	$9.29 \cdot 10^{-01}$	80
2	$3.67 \cdot 10^{-02}$	$9.60 \cdot 10^{-02}$	30
3	$1.36 \cdot 10^{-02}$	$1.09 \cdot 10^{-03}$	28
4	$3.47 \cdot 10^{-04}$	$1.01 \cdot 10^{-07}$	35
5	$6.41 \cdot 10^{-08}$	$1.03 \cdot 10^{-10}$	25
6	$1.23 \cdot 10^{-11}$	**$1.98 \cdot 10^{-11}$**	27

CPU time: 1 460.1 sec.

test system: Intel® Xeon® 5160 3.00GHz ; 16 GB RAM;
64Bit-MATLAB® (R2010a) using threaded BLAS (`romulus`)
stopping criterion tolerances: 10^{-10}

LRCF-NM for the ARE

Test Results (ADI-loop): Example 2

Newton-ADI

NWT	rel. change	rel. residual	ADI
1	1	$1.70 \cdot 10^{+02}$	46
2	$2.88 \cdot 10^{-01}$	$4.25 \cdot 10^{+01}$	39
3	$2.13 \cdot 10^{-01}$	$1.06 \cdot 10^{+01}$	43
4	$1.77 \cdot 10^{-01}$	$2.58 \cdot 10^{+00}$	46
5	$2.47 \cdot 10^{-01}$	$5.15 \cdot 10^{-01}$	43
6	$3.04 \cdot 10^{-01}$	$3.26 \cdot 10^{-02}$	52
7	$1.78 \cdot 10^{-02}$	$6.90 \cdot 10^{-05}$	50
8	$2.60 \cdot 10^{-05}$	$1.08 \cdot 10^{-10}$	46
9	$\mathbf{2.75 \cdot 10^{-11}}$	$1.07 \cdot 10^{-10}$	50

CPU time: 493.81 sec.

Newton-Galerkin-ADI

LRCF-ADI-GP(5)

NWT	rel. change	rel. residual	ADI
1	1	$1.70 \cdot 10^{+02}$	35
2	$2.88 \cdot 10^{-01}$	$4.25 \cdot 10^{+01}$	15
3	$2.13 \cdot 10^{-01}$	$1.06 \cdot 10^{+01}$	20
4	$1.77 \cdot 10^{-01}$	$2.58 \cdot 10^{+00}$	20
5	$2.47 \cdot 10^{-01}$	$5.15 \cdot 10^{-01}$	20
6	$3.04 \cdot 10^{-01}$	$3.26 \cdot 10^{-02}$	17
7	$1.78 \cdot 10^{-02}$	$6.90 \cdot 10^{-05}$	20
8	$2.60 \cdot 10^{-05}$	$1.10 \cdot 10^{-10}$	20
9	$2.75 \cdot 10^{-11}$	$\mathbf{1.92 \cdot 10^{-12}}$	20

CPU time: 280.55 sec.

test system: Intel® Core™2 Quad Q9400 2.66 GHz; 4 GB RAM;
64Bit-MATLAB® (R2009a) using threaded BLAS (`reynolds`)
stopping criterion tolerances: 10^{-10}

LRCF-NM for the ARE

Test Results (both-loops): Example 1

Newton-ADI

NWT	rel. change	rel. residual	ADI
1	$9.97 \cdot 10^{-01}$	$9.27 \cdot 10^{-01}$	100
2	$3.67 \cdot 10^{-02}$	$9.58 \cdot 10^{-02}$	94
3	$1.36 \cdot 10^{-02}$	$1.09 \cdot 10^{-03}$	98
4	$3.48 \cdot 10^{-04}$	$1.01 \cdot 10^{-07}$	97
5	$6.41 \cdot 10^{-08}$	$1.34 \cdot 10^{-10}$	97
6	$7.47 \cdot 10^{-16}$	$1.34 \cdot 10^{-10}$	97

CPU time: 4 805.8 sec.

NG-ADI inner= 5, outer= 1

NWT	rel. change	rel. residual	ADI
1	$9.98 \cdot 10^{-01}$	$5.04 \cdot 10^{-11}$	80

CPU time: 497.6 sec.

NG-ADI inner= 1, outer= 1

NWT	rel. change	rel. residual	ADI
1	$9.98 \cdot 10^{-01}$	$7.42 \cdot 10^{-11}$	71

CPU time: 856.6 sec.

NG-ADI inner= 0, outer= 1

NWT	rel. change	rel. residual	ADI
1	$9.98 \cdot 10^{-01}$	$6.46 \cdot 10^{-13}$	100

CPU time: 506.6 sec.

test system: Intel® Xeon® 5160 3.00GHz ; 16 GB RAM;
64Bit-MATLAB® (R2010a) using threaded BLAS (`romulus`)
stopping criterion tolerances: 10^{-10}

LRCF-NM for the ARE

Test Results (both-loops): Example 2

Newton-ADI

NWT	rel. change	rel. residual	ADI
1	1	$1.70 \cdot 10^{+02}$	46
2	$2.88 \cdot 10^{-01}$	$4.25 \cdot 10^{+01}$	39
3	$2.13 \cdot 10^{-01}$	$1.06 \cdot 10^{+01}$	43
4	$1.77 \cdot 10^{-01}$	$2.58 \cdot 10^{+00}$	46
5	$2.47 \cdot 10^{-01}$	$5.15 \cdot 10^{-01}$	43
6	$3.04 \cdot 10^{-01}$	$3.26 \cdot 10^{-02}$	52
7	$1.78 \cdot 10^{-02}$	$6.90 \cdot 10^{-05}$	50
8	$2.60 \cdot 10^{-05}$	$1.08 \cdot 10^{-10}$	46
9	**$2.75 \cdot 10^{-11}$**	$1.07 \cdot 10^{-10}$	50

CPU time: 493.81 sec.

NG-ADI inner= 5, outer= 1

NWT	rel. change	rel. residual	ADI
1	1	$3.30 \cdot 10^{-11}$	35

CPU time: 24.1 sec.

NG-ADI inner= 1, outer= 1

NWT	rel. change	rel. residual	ADI
1	1	$1.31 \cdot 10^{-11}$	34

CPU time: 26.8 sec.

NG-ADI inner= 0, outer= 1

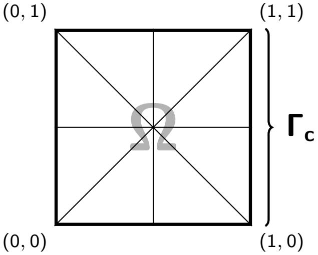
NWT	rel. change	rel. residual	ADI
1	1	$3.27 \cdot 10^{-15}$	46

CPU time: 24.0 sec.

test system: Intel® Core™2 Quad Q9400 2.66 GHz; 4 GB RAM;
64Bit-MATLAB® (R2009a) using threaded BLAS (`reynolds`)
stopping criterion tolerances: 10^{-10}

LRCF-NM for the ARE

Computation Time Scaling with Problem Size

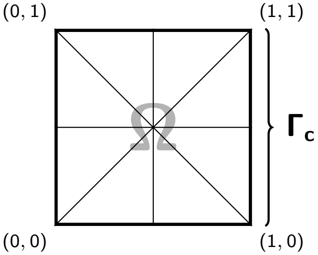


$$\begin{aligned} \partial_t x(\xi,t) &= \Delta x(\xi,t) && \text{in } \Omega \\ \partial_\nu x &= b(\xi)\cdot u(t) - x && \text{on } \Gamma_c \\ \partial_\nu x &= -x && \text{on } \partial\Omega \setminus \Gamma_c \\ x(\xi,0) &= 1 \end{aligned}$$

Note:
Here $b(\xi) = 4\,(1-\xi_2)\,\xi_2$ for $\xi \in \Gamma_c$ and 0 otherwise, thus $\forall t \in \mathbb{R}_{>0}$, we have $u(t) \in \mathbb{R}$.

LRCF-NM for the ARE

Computation Time Scaling with Problem Size



$$
\begin{aligned}
\partial_t x(\xi, t) &= \Delta x(\xi, t) && \text{in } \Omega \\
\partial_\nu x &= b(\xi) \cdot u(t) - x && \text{on } \Gamma_c \\
\partial_\nu x &= -x && \text{on } \partial\Omega \setminus \Gamma_c
\end{aligned}
$$

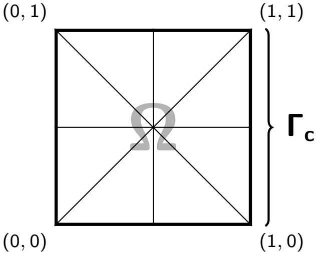
$$x(\xi, 0) = 1$$

Note:
Here $b(\xi) = 4\,(1 - \xi_2)\,\xi_2$ for $\xi \in \Gamma_c$ and 0 otherwise, thus $\forall t \in \mathbb{R}_{>0}$, we have $u(t) \in \mathbb{R}$.

$$\Rightarrow B_h = M_{\Gamma,h} \cdot b.$$

LRCF-NM for the ARE

Computation Time Scaling with Problem Size



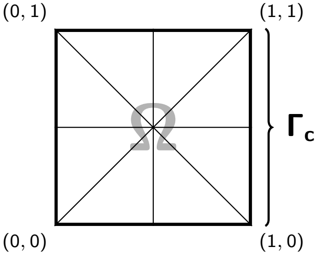
$$\begin{aligned} \partial_t x(\xi,t) &= \Delta x(\xi,t) && \text{in } \Omega \\ \partial_\nu x &= b(\xi)\cdot u(t) - x && \text{on } \Gamma_c \\ \partial_\nu x &= -x && \text{on } \partial\Omega \setminus \Gamma_c \\ x(\xi,0) &= 1 \end{aligned}$$

Consider: output equation $y = Cx$, where

$$\begin{aligned} C : \mathcal{L}^2(\Omega) &\rightarrow \mathbb{R} \\ x(\xi,t) &\mapsto y(t) = \int_\Omega x(\xi,t)\,d\xi. \end{aligned}$$

LRCF-NM for the ARE

Computation Time Scaling with Problem Size



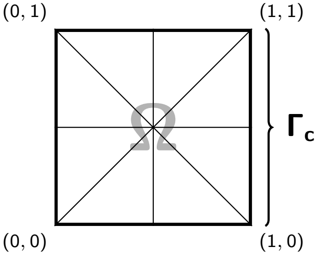
$$\begin{aligned} \partial_t x(\xi, t) &= \Delta x(\xi, t) && \text{in } \Omega \\ \partial_\nu x &= b(\xi) \cdot u(t) - x && \text{on } \Gamma_c \\ \partial_\nu x &= -x && \text{on } \partial\Omega \setminus \Gamma_c \\ x(\xi, 0) &= 1 \end{aligned}$$

Consider: output equation $y = Cx$, where

$$\begin{aligned} C : \mathcal{L}^2(\Omega) &\to \mathbb{R} \\ x(\xi, t) &\mapsto y(t) = \int_\Omega x(\xi, t)\, d\xi, \end{aligned} \quad \Rightarrow C_h = \underline{1} \cdot M_h.$$

LRCF-NM for the ARE

Computation Time Scaling with Problem Size



$$\begin{aligned} \partial_t x(\xi,t) &= \Delta x(\xi,t) && \text{in } \Omega \\ \partial_\nu x &= b(\xi)\cdot u(t) - x && \text{on } \Gamma_c \\ \partial_\nu x &= -x && \text{on } \partial\Omega \setminus \Gamma_c \end{aligned}$$

$$x(\xi,0) = 1$$

Cost Function:

$$\mathcal{J}(u) = \int_0^\infty y^2(t) + u^2(t)\,dt.$$

LRCF-NM for the ARE

Computation Time Scaling with Problem Size

simplified Low Rank Newton-Galerkin ADI

- generalized state space form implementation
- Penzl shifts (16/50/25) with respect to initial matrices
- projection acceleration in every outer iteration step
- projection acceleration in every 5-th inner iteration step

test system: Intel®Xeon® 5160 @ 3.00 GHz; 16 GB RAM;
64Bit-MATLAB® (R2010a) using threaded BLAS (`romulus`)
stopping criterion tolerances: 10^{-10}

LRCF-NM for the ARE

Computation Time Scaling with Problem Size

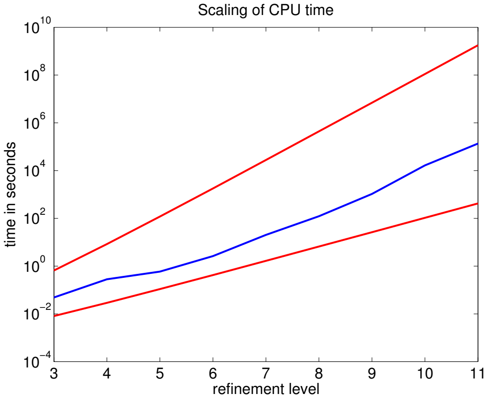
Computation Times

discretization level	problem size	time in seconds
3	81	$4.87\,10^{-2}$
4	289	$2.81\,10^{-1}$
5	1 089	$5.87\,10^{-1}$
6	4 225	2.63
7	16 641	$2.03\,10^{+1}$
8	66 049	$1.22\,10^{+2}$
9	263 169	$1.05\,10^{+3}$
10	1 050 625	$1.65\,10^{+4}$
11	4 198 401	$1.35\,10^{+5}$

test system: Intel®Xeon® 5160 @ 3.00 GHz; 16 GB RAM;
64Bit-MATLAB® (R2010a) using threaded BLAS (`romulus`)
stopping criterion tolerances: 10^{-10}

LRCF-NM for the ARE

Computation Time Scaling with Problem Size



test system: Intel®Xeon® 5160 @ 3.00 GHz; 16 GB RAM; 64Bit-MATLAB® (R2010a) using threaded BLAS (`romulus`)
stopping criterion tolerances: 10^{-10}