Efficient Implementation of Large Scale Lyapunov and Riccati Equation Solvers

Jens Saak
joint work with
Peter Benner (MiIT)

Professur Mathematik in Industrie und Technik (MiIT)
Fakultät für Mathematik
Technische Universität Chemnitz

Computational Methods with Applications
Harrachov August 24, 2007

Aim of this talk

What is the aim of this talk?

Promote the upcoming release 1.1 of the LyaPack software package.

Aim of this talk

What is the aim of this talk?

Promote the upcoming release 1.1 of the LyaPack software package.

What is LyaPack?

Aim of this talk

What is the aim of this talk?

Promote the upcoming release 1.1 of the LyaPack software package.

What is LyaPack?

MATLAB toolbox for solving large scale

- Lyapunov equations (applications like in M. Embrees plenary talk on Tuesday)
- Riccati equations
- linear quadratic optimal control problems

Origin of the Riccati equations

semi discrete parabolic PDE

$$\dot{x}(t) = Ax(t) + Bu(t) \quad x(0) = x_0 \in \mathcal{X}. \qquad \text{(Cauchy)}$$

output equation

$$y(t) = Cx(t) \qquad \text{(output)}$$

cost function

$$\mathcal{J}(u) = \frac{1}{2}\int_0^\infty < y, y > + < u, u > dt \qquad \text{(cost)}$$

and the linear quadratic regulator problem is

LQR problem

Minimize the **quadratic** (cost) with respect to the **linear** constraints (Cauchy),(output).

Origin of the Riccati equations

semi discrete parabolic PDE

$$\dot{x}(t) = Ax(t) + Bu(t) \quad x(0) = x_0 \in \mathcal{X}. \qquad \text{(Cauchy)}$$

output equation

$$y(t) = Cx(t) \qquad \text{(output)}$$

cost function

$$\mathcal{J}(u) = \frac{1}{2}\int_0^\infty < Cx, Cx > + < u, u > dt \qquad \text{(cost)}$$

and the linear quadratic regulator problem is

LQR problem

Minimize the **quadratic** (cost) with respect to the **linear** constraints (Cauchy),(output).

Origin of the Riccati equations

In the open literature it is well understood that the

optimal feedback control

is given as

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

where X_∞ is the minimal, positive semidefinite, selfadjoint solution of the

algebraic Riccati equation

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

Outline

LRCF Newton Method for the ARE

LRCF Newton Method for the ARE

Large Scale Riccati and Lyapunov Equations

We are interested in solving

algebraic Riccati equations

$$0 = A^T P + PA - PBB^T P + C^T C =: \mathfrak{R}(P). \qquad \text{(ARE)}$$

where

- $A \in \mathbb{R}^{n \times n}$ sparse, $n \in \mathbb{N}$ "large"
- $B \in \mathbb{R}^{n \times m}$ and $m \in \mathbb{N}$ with $m \ll n$
- $C \in \mathbb{R}^{p \times n}$ and $p \in \mathbb{N}$ with $p \ll n$

LRCF Newton Method for the ARE

Large Scale Riccati and Lyapunov Equations

We are interested in solving

algebraic Riccati equations

$$0 = A^T P + PA - PBB^T P + C^T C =: \mathfrak{R}(P). \qquad \text{(ARE)}$$

where

- $A \in \mathbb{R}^{n \times n}$ sparse, $n \in \mathbb{N}$ "large"
- $B \in \mathbb{R}^{n \times m}$ and $m \in \mathbb{N}$ with $m \ll n$
- $C \in \mathbb{R}^{p \times n}$ and $p \in \mathbb{N}$ with $p \ll n$

and

Lyapunov equations

$$F^T P + PF = -GG^T. \qquad \text{(LE)}$$

with

- $F \in \mathbb{R}^{n \times n}$ sparse or sparse + low rank update, $n \in \mathbb{N}$ "large"
- $G \in \mathbb{R}^{n \times m}$ and $m \in \mathbb{N}$ with $m \ll n$

LRCF Newton Method for the ARE

Newton's method for solving the ARE

Newton's iteration for the ARE

$$\mathfrak{R}'|_P(N_l) = -\mathfrak{R}(P_l), \qquad P_{l+1} = P_l + N_l,$$

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

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

can be rewritten as

one iteration step

$$(A - BB^T P_l)^T P_{l+1} + P_{l+1}(A - BB^T P_l) = -C^T C - P_l BB^T P_l$$

i.e. in every Newton step we have to solve a Lyapunov equation of the form (LE)

LRCF Newton Method for the ARE

Cholesky factor ADI for Lyapunov equations

Recall **Peaceman Rachford ADI:**
Consider $Au = s$ where $A \in \mathbb{R}^{n \times n}$ spd, $s \in \mathbb{R}^n$. ADI Iteration Idea:
Decompose $A = H + V$ with $H, V \in \mathbb{R}^{n \times n}$ such that

$$\begin{aligned}(H + pI)v &= r \\ (V + pI)w &= t\end{aligned}$$

can be solved easily/efficiently.

LRCF Newton Method for the ARE

Cholesky factor ADI for Lyapunov equations

Recall **Peaceman Rachford ADI:**
Consider $Au = s$ where $A \in \mathbb{R}^{n \times n}$ spd, $s \in \mathbb{R}^n$. ADI Iteration Idea:
Decompose $A = H + V$ with $H, V \in \mathbb{R}^{n \times n}$ such that

$$\begin{aligned}(H + pI)v &= r\\ (V + pI)w &= t\end{aligned}$$

can be solved easily/efficiently.

ADI Iteration

If H, V spd $\Rightarrow \exists p_j$, $j = 1, 2, \dots J$ such that

$$\begin{aligned} u_0 &= 0 \\ (H + p_j I)u_{j-\frac{1}{2}} &= (p_j I - V)u_{j-1} + s \\ (V + p_j I)u_j &= (p_j I - H)u_{j-\frac{1}{2}} + s \end{aligned} \qquad \text{(PR-ADI)}$$

converges to $u \in \mathbb{R}^n$ solving $Au = s$.

LRCF Newton Method for the ARE

Cholesky factor ADI for Lyapunov equations

The Lyapunov operator

$$\mathcal{L}: \quad P \quad \mapsto \quad F^T P + PF$$

can be decomposed into the linear operators

$$\mathcal{L}_H : P \mapsto F^T P \qquad \mathcal{L}_V : P \mapsto PF.$$

Such that in analogy to (PR-ADI) we find the

ADI iteration for the Lyapunov equation (LE)

$$\begin{aligned} P_0 &= 0 \\ (F^T + p_j I)P_{j-\frac{1}{2}} &= -GG^T - P_{j-1}(F - p_j I) \\ (F^T + p_j I)P_j^T &= -GG^T - P_{j-\frac{1}{2}}^T(F - p_j I) \end{aligned} \qquad \text{(LE-ADI)}$$

LRCF Newton Method for the ARE

Cholesky factor ADI for Lyapunov equations

Remarks:

- If F is sparse or sparse + low rank update, i.e. $F = A + VU^T$ then $F^T + p_j I$ can be written as $\tilde{A} + UV^T$, where $\tilde{A} = A^T + p_j I$ and its inverse can be expressed as

$$(F^T + p_j I)^{-1} = (\tilde{A} + UV^T)^{-1} = \tilde{A}^{-1} - \tilde{A}^{-1} U (I + V^T \tilde{A}^{-1} U)^{-1} V^T \tilde{A}^{-1}$$

  by the Sherman-Morrison-Woodbury formula.

LRCF Newton Method for the ARE

Cholesky factor ADI for Lyapunov equations

Remarks:

- If F is sparse or sparse + low rank update, i.e. $F = A + VU^T$ then $F^T + p_j I$ can be written as $\tilde{A} + UV^T$, where $\tilde{A} = A^T + p_j I$ and its inverse can be expressed as

$$(F^T + p_j I)^{-1} = (\tilde{A} + UV^T)^{-1} = \tilde{A}^{-1} - \tilde{A}^{-1}U(I + V^T\tilde{A}^{-1}U)^{-1}V^T\tilde{A}^{-1}$$

  by the Sherman-Morrison-Woodbury formula.
  Note: We only need to be able to multiply with A, solve systems with A and solve shifted systems with $A^T + p_j I$

LRCF Newton Method for the ARE

Cholesky factor ADI for Lyapunov equations

Remarks:

- If F is sparse or sparse + low rank update, i.e. $F = A + VU^T$ then $F^T + p_j I$ can be written as $\tilde{A} + UV^T$, where $\tilde{A} = A^T + p_j I$ and its inverse can be expressed as

$$(F^T + p_j I)^{-1} = (\tilde{A} + UV^T)^{-1} = \tilde{A}^{-1} - \tilde{A}^{-1} U (I + V^T \tilde{A}^{-1} U)^{-1} V^T \tilde{A}^{-1}$$

  by the Sherman-Morrison-Woodbury formula.
- (LE-ADI) can be rewritten to iterate on the low rank Cholesky factors Z_j of P_j to exploit $\text{rk}(P_j) \ll n$. [J. R. Li and J. White 2002; T. Penzl 1999; P. Benner, J.R. Li and T. Penzl 2000]

LRCF Newton Method for the ARE

Cholesky factor ADI for Lyapunov equations

Remarks:

- If F is sparse or sparse + low rank update, i.e. $F = A + VU^T$ then $F^T + p_j I$ can be written as $\tilde{A} + UV^T$, where $\tilde{A} = A^T + p_j I$ and its inverse can be expressed as

$$(F^T + p_j I)^{-1} = (\tilde{A} + UV^T)^{-1} = \tilde{A}^{-1} - \tilde{A}^{-1}U(I + V^T\tilde{A}^{-1}U)^{-1}V^T\tilde{A}^{-1}$$

  by the Sherman-Morrison-Woodbury formula.
- (LE-ADI) can be rewritten to iterate on the low rank Cholesky factors Z_j of P_j to exploit $\text{rk}(P_j) \ll n$. [J. R. Li and J. White 2002; T. Penzl 1999; P. Benner, J.R. Li and T. Penzl 2000]
- When solving (ARE) to compute the feedback in an LQR-problem for a semidiscretized parabolic PDE, the LRCF-Newton-ADI can directly iterate on the feedback matrix $K \in \mathbb{R}^{n\times p}$ to save even more memory. [T. Penzl 1999; P. Benner, J. R. Li and T. Penzl 2000]

Reordering Strategies

Reordering Strategies

Introduction

Use sparse direct solvers ⇒ Store LU or Cholesky factors frequently used (e.g. for M or $A + p_j I$ in case of cyclically used shifts).

⇒ Save storage by reordering

Upcoming LyaPack 1.1 let's you choose between:

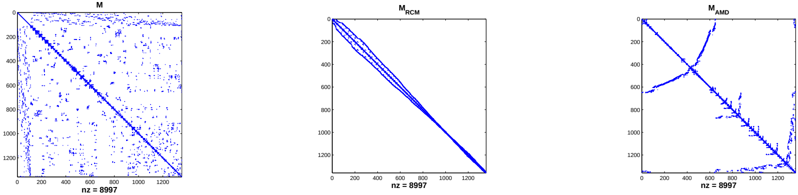
- symmetric reverse Cuthill-McKee (RCM[1]) reordering
- approximate minimum degree (AMD[2]) reordering
- symmetric AMD[2]

[1] [A. George and J. W.-H. Liu 1981]
[2] [P. Amestoy, T. A. Davis, and I. S. Duff 1996.]

Reordering Strategies

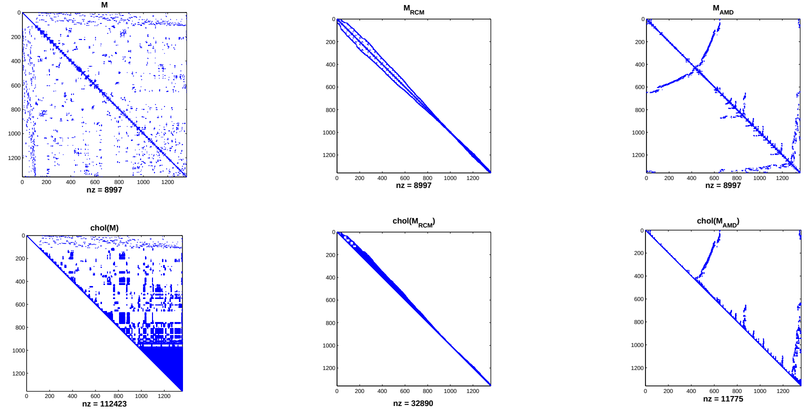
Motivation

Motivating example: Mass matrix M from a FEM semidiscretization of a 2d heat equation. Dimension of the discrete system: 1357



Reordering Strategies
Motivation

Motivating example: Mass matrix M from a FEM semidiscretization of a 2d heat equation. Dimension of the discrete system: 1357



ADI Shift Parameters

ADI Shift Parameters

Introduction

Optimal parameters solve the

min-max-problem

$$\min_{\{p_j | j=1,\dots,J\}\subset\mathbb{R}} \max_{\gamma\in\sigma(F)} \left|\prod_{j=1}^{J} \frac{(p_j-\lambda)}{(p_j+\lambda)}\right|.$$

Remark

- Also known as rational Zolotarev problem since he solved it first on real intervals enclosing the spectra in 1877.
- Another solution for the real case was presented by Wachspress/Jordan in 1963.

ADI Shift Parameters
Introduction

Optimal parameters solve the

min-max-problem

$$\min_{\{p_j | j=1,\dots,J\} \subset \mathbb{R}} \quad \max_{\gamma \in \sigma(F)} \quad \left| \prod_{j=1}^{J} \frac{(p_j - \lambda)}{(p_j + \lambda)} \right| .$$

Remark

- Wachspress and Starke presented different strategies to compute suboptimal shifts for the complex case around 1990.
- Wachspress: elliptic Integral evaluation based shifts
- Starke: Leja Point based shifts (recall M. Embrees plenary talk on Tuesday)

ADI Shift Parameters

Available (sub)optimal choices

ADI shift parameter choices in upcoming LyaPack 1.1

1. heuristic parameters [T. Penzl 1999]
   - use selected Ritz values as shifts
   - suboptimal ⇒ convergence might be slow
   - in general complex for complex spectra
2. approximate Wachspress parameters [P. Benner, H. Mena, J. Saak 2006]
   - optimal for real spectra
   - parameters real if imaginary parts are "small"
   - good approximation of the "outer" spectrum of F needed
     ⇒ possibly expensive computation
3. only real heuristic parameters
   - avoids complex computation and storage requirements
   - can be slow if many Ritz values are complex
4. real parts of heuristic parameters
   - avoids complex computation and storage requirements
   - suitable if imaginary parts are "small"

ADI Shift Parameters

Available (sub)optimal choices

ADI shift parameter choices in upcoming LyaPack 1.1

1. heuristic parameters [T. Penzl 1999]
   - use selected Ritz values as shifts
   - suboptimal ⇒ convergence might be slow
   - in general complex for complex spectra
2. approximate Wachspress parameters [P. Benner, H. Mena, J. Saak 2006]
   - optimal for real spectra
   - parameters real if imaginary parts are "small"
   - good approximation of the "outer" spectrum of F needed
     ⇒ possibly expensive computation
3. only real heuristic parameters
   - avoids complex computation and storage requirements
   - can be slow if many Ritz values are complex
4. real parts of heuristic parameters
   - avoids complex computation and storage requirements
   - suitable if imaginary parts are "small"

ADI Shift Parameters

Available (sub)optimal choices

ADI shift parameter choices in upcoming LyaPack 1.1

1. heuristic parameters [T. Penzl 1999]
   - use selected Ritz values as shifts
   - suboptimal ⇒ convergence might be slow
   - in general complex for complex spectra
2. approximate Wachspress parameters [P. Benner, H. Mena, J. Saak 2006]
   - optimal for real spectra
   - parameters real if imaginary parts are "small"
   - good approximation of the "outer" spectrum of F needed ⇒ possibly expensive computation
3. only real heuristic parameters
   - avoids complex computation and storage requirements
   - can be slow if many Ritz values are complex
4. real parts of heuristic parameters
   - avoids complex computation and storage requirements
   - suitable if imaginary parts are "small"

ADI Shift Parameters

Available (sub)optimal choices

ADI shift parameter choices in upcoming LyaPack 1.1

1. heuristic parameters [T. Penzl 1999]
   - use selected Ritz values as shifts
   - suboptimal ⇒ convergence might be slow
   - in general complex for complex spectra
2. approximate Wachspress parameters [P. Benner, H. Mena, J. Saak 2006]
   - optimal for real spectra
   - parameters real if imaginary parts are "small"
   - good approximation of the "outer" spectrum of F needed ⇒ possibly expensive computation
3. only real heuristic parameters
   - avoids complex computation and storage requirements
   - can be slow if many Ritz values are complex
4. real parts of heuristic parameters
   - avoids complex computation and storage requirements
   - suitable if imaginary parts are "small"

ADI Shift Parameters

Numerical Tests

Test example

Centered finite difference discretized 2d convection diffusion equation:

$$\dot{\mathbf{x}} = \Delta\mathbf{x} - 10\mathbf{x}_x - 100\mathbf{x}_y + \mathbf{b}(x,y)\mathbf{u}(t)$$

on the unit square with Dirichlet boundary conditions. (`demo_l1.m`)

ADI Shift Parameters

Numerical Tests

Test example

Centered finite difference discretized 2d convection diffusion equation:

$$\dot{\mathbf{x}} = \Delta\mathbf{x} - 10\mathbf{x}_x - 100\mathbf{x}_y + \mathbf{b}(x,y)\mathbf{u}(t)$$

on the unit square with Dirichlet boundary conditions. (`demo_l1.m`)

grid size: $75 \times 75 \Rightarrow$ #states $= 5625 \Rightarrow$ #unknowns in $X = 5625^2 \approx 32 \cdot 10^6$

Computations carried out on Intel Core2 Duo @2.13GHz Cache: 2048kB RAM: 2GB

ADI Shift Parameters

Numerical Tests

Test example

Centered finite difference discretized 2d convection diffusion equation:

$$\dot{\mathbf{x}} = \Delta \mathbf{x} - 10\mathbf{x}_x - 100\mathbf{x}_y + \mathbf{b}(x,y)\mathbf{u}(t)$$

on the unit square with Dirichlet boundary conditions. (`demo_l1.m`)

grid size: $75 \times 75 \Rightarrow$ #states $= 5625 \Rightarrow$ #unknowns in $X = 5625^2 \approx 32 \cdot 10^6$

heuristic parameters time: 44s residual norm: 1.0461e-11

heuristic real parts time: 13s residual norm: 9.0846e-12

appr. Wachspress time: 16s residual norm: 5.3196e-12

Remark

- heuristic parameters are complex
- problem size exceeds memory limitations in complex case

Computations carried out on Intel Core2 Duo @2.13GHz Cache: 2048kB RAM: 2GB

Column Compression for the low rank factors

Column Compression for the low rank factors

Introduction

Problem

- Low rank factors Z of the solutions X grow rapidly, since a constant number of columns is added in every ADI step.
- If convergence is weak, at some point $\#\text{columns in } Z > \text{rk}(Z)$.

Idea [Antoulas, Gugercin, Sorensen 2003]

Use sequential Karhunen-Loeve algorithm; see [Baker 2004]

- uses QR + SVD for rank truncation

Column Compression for the low rank factors

Introduction

Problem

- Low rank factors Z of the solutions X grow rapidly, since a constant number of columns is added in every ADI step.
- If convergence is weak, at some point $\#\text{columns in } Z > \text{rk}(Z)$.

Cheaper idea:
Column compression using rank revealing QR factorization (RRQR)

Consider $X = ZZ^T$ and $\text{rk}(Z) = r$. Compute the RRQR[3] of Z

$$Z^T = QR\Pi \quad \text{where} \quad R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \quad \text{and } R_{11} \in \mathbb{R}^{r \times r}$$

now set $\tilde{Z}^T = [R_{11} R_{12}]\,\Pi^T$ then $\tilde{Z}\tilde{Z}^T =: \tilde{X} = X$.

[3] [C.H. Bischof and G. Quintana-Ortí 1998]

Column Compression for the low rank factors

Numerical Tests

Test example

Centered finite difference discretized 2d convection diffusion equation:

$$\dot{\mathbf{x}} = \Delta\mathbf{x} - 10\mathbf{x}_x - 100\mathbf{x}_y + \mathbf{b}(x,y)\mathbf{u}(t)$$

on the unit square with Dirichlet boundary conditions. (`demo_l1.m`)

Column Compression for the low rank factors

Numerical Tests

Test example

Centered finite difference discretized 2d convection diffusion equation:

$$\dot{\mathbf{x}} = \Delta \mathbf{x} - 10\mathbf{x}_x - 100\mathbf{x}_y + \mathbf{b}(x, y)\mathbf{u}(t)$$

on the unit square with Dirichlet boundary conditions. (`demo_l1.m`)

grid size: $75 \times 75 \Rightarrow$ #states $= 5625 \Rightarrow$ #unknowns in $X = 5625^2 \approx 32 \cdot 10^6$

truncation TOL	# col. in LRCF	time	res. norm
–	47	13s	9.0846e-12
eps	46	14s	1.9516e-11
$\sqrt{\text{eps}}$	28	13s	1.9924e-11

Computations carried out on Intel Core2 Duo @2.13GHz Cache: 2048kB RAM: 2GB

Column Compression for the low rank factors

Numerical Tests

Test example

Centered finite difference discretized 2d convection diffusion equation:

$$\dot{\mathbf{x}} = \Delta \mathbf{x} - 10\mathbf{x}_x - 100\mathbf{x}_y + \mathbf{b}(x,y)\mathbf{u}(t)$$

on the unit square with Dirichlet boundary conditions. (`demo_l1.m`)

grid size: $75 \times 75 \Rightarrow$ #states $= 5625 \Rightarrow$ #unknowns in $X = 5625^2 \approx 32 \cdot 10^6$

truncation TOL	# col. in LRCF	time	res. norm
–	47	13s	9.0846e-12
eps	46	14s	1.9516e-11
$\sqrt{\text{eps}}$	28	13s	1.9924e-11

Observation

[Benner and Quintana-Ortí 2005] showed that truncation tolerance $\sqrt{eps}$ in the low rank factor Z is sufficient to achieve an error eps in the solution X.

Computations carried out on Intel Core2 Duo @2.13GHz Cache: 2048kB RAM: 2GB

Generalized Systems

Generalized Systems

2 Basic Ideas in Contrast

Current Method

Transform

$$\begin{aligned} M\dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

to

$$\begin{aligned} \dot{\tilde{x}} &= \tilde{A}\tilde{x} + \tilde{B}u \\ y &= \tilde{C}\tilde{x} \end{aligned}$$

where $M = M_L M_U$ and $\tilde{x} = M_U x$, $\tilde{A} = M_L^{-1} A M_U^{-1}$, $\tilde{B} = M_L^{-1} B$, $\tilde{C} = C M_U^{-1}$.

- 2 additional sparse triangular solves in every multiplication with A
- 2 additional sparse matrix vector multiplies in solution of $\tilde{A}x = b$ and $(\tilde{A} + p_j I)x = b$
- $\tilde{B}$ and $\tilde{C}$ are dense even if B and C are sparse.

\+ preserves symmetry if M, A are symmetric.

Generalized Systems

2 Basic Ideas in Contrast

Alternative Method

Transform

$$\begin{aligned} M\dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

to

$$\begin{aligned} \dot{x} &= \tilde{A}x + \tilde{B}u \\ y &= Cx \end{aligned}$$

where $\tilde{A} = M^{-1}A$ and $\tilde{B} = M^{-1}B$

+ state variable untouched $\Rightarrow$ solution to (ARE), (LE) not transformed
+ exploiting pencil structure in $(\tilde{A} + p_j I) = M^{-1}(A + p_j M)$ reduces overhead
- current user supplied function structure inefficient here
  $\Rightarrow$ rewrite of LyaPack kernel routines needed (work in progress)

Generalized Systems

2 Basic Ideas in Contrast

Alternative Method

Transform

$$\begin{aligned} M\dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

to

$$\begin{aligned} \dot{x} &= \tilde{A}x + \tilde{B}u \\ y &= Cx \end{aligned}$$

where $\tilde{A} = M^{-1}A$ and $\tilde{B} = M^{-1}B$

+ state variable untouched $\Rightarrow$ solution to (ARE), (LE) not transformed
+ exploiting pencil structure in $(\tilde{A} + p_j I)^{-1} = (A + p_j M)^{-1} M$ reduces overhead
- current user supplied function structure inefficient here
  $\Rightarrow$ rewrite of LyaPack kernel routines needed (work in progress)

Conclusions and Outlook

Conclusions and Outlook
Conlusions

- Reordering strategies can reduce memory requirements by far

Conclusions and Outlook

Conlusions

- Reordering strategies can reduce memory requirements by far
- new shift parameter selection allows easy improvements in ADI performance

Conclusions and Outlook

Conlusions

- Reordering strategies can reduce memory requirements by far
- new shift parameter selection allows easy improvements in ADI performance
- Column compression via RRQR also drastically reduces storage requirements.

Conclusions and Outlook

Conlusions

- Reordering strategies can reduce memory requirements by far
- new shift parameter selection allows easy improvements in ADI performance
- Column compression via RRQR also drastically reduces storage requirements. Especially helpful in differential Riccati equation solvers where 1 ARE solution needs to be stored in every step.

Conclusions and Outlook

Conlusions

- Reordering strategies can reduce memory requirements by far
- new shift parameter selection allows easy improvements in ADI performance
- Column compression via RRQR also drastically reduces storage requirements. Especially helpful in differential Riccati equation solvers where 1 ARE solution needs to be stored in every step.
- Optimized treatment of generalized systems is work in progress

Conclusions and Outlook

Outlook

Theoretical Outlook

- Improve stopping Criteria for the ADI process.
  e.g. inside the LRCF-Newton method by interpretation as inexact Newton method following the ideas of Sachs et al.
- Optimize truncation tolerances for the RRQR
  Investigate dependence of residual errors in X on the truncation tolerance
- Stabilizing initial feedback computation
  Investigate whether the method in [K. Gallivan, X. Rao and P. Van Dooren 2006] can be implemented exploiting sparse matrix arithmetics.

Conclusions and Outlook

Outlook

Implementation TODOs

- User supplied functions for B (and C?)
- Introduce solvers for DREs
- Initial stabilizing feedback computation
- Improve handling of generalized systems of the form $M\dot{x} = Ax + Bu$.
- Improve the current Arnoldi implementation inside the heuristic ADI Parameter computation
- RRQR and column compression for complex factors.
- Simplify calling sequences, i.e. shorten commands by grouping parameters in structures
- Improve overall performance
- ...

Conclusions and Outlook

Outlook

Implementation TODOs

- User supplied functions for B (and C?)
- Introduce solvers for DREs
- Initial stabilizing feedback computation
- Improve handling of generalized systems of the form $M\dot{x} = Ax + Bu$.
- Improve the current Arnoldi implementation inside the heuristic ADI Parameter computation
- RRQR and column compression for complex factors.
- Simplify calling sequences, i.e. shorten commands by grouping parameters in structures
- Improve overall performance
- ...

Thank you for your attention!