Efficient numerical solution of large scale matrix equations arising in LQR/LQG design for parabolic PDEs

Jens Saak

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

PDE Constrained Optimization -
recent challenges and future developments
Hamburg March 27-29, 2008

Department of Mathematics
CHEMNITZ UNIVERSITY OF TECHNOLOGY

Outline

Origin of the Matrix Equations

Origin of the Matrix Equations

LQR for linear parabolic PDEs

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

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

and the linear quadratic regulator problem is

LQR problem

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

Origin of the Matrix Equations

LQR for linear parabolic PDEs

semi discrete parabolic PDE

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

output equation

$$y(t) = Cx(t) \tag{output}$$

cost function

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

and the linear quadratic regulator problem is

LQR problem

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

Origin of the Matrix Equations
LQR for linear parabolic PDEs

In the open literature[1] it is well understood that the

optimal feedback control

is given as

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

where in case $T_f = \infty$, X_∞ is the minimal, positive semidefinite, selfadjoint solution of the

algebraic Riccati equation (ARE)

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

[1] e.g. [Lions '71; Lasiecka/Triggiani '00; Bensoussan et al. '92; Pritchard/Salamon '87; Curtain/Zwart '95]

Origin of the Matrix Equations
LQR for linear parabolic PDEs

In the open literature[1] it is well understood that the

optimal feedback control

is given as

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

where in case $T_f < \infty$, X_∞ is the minimal, positive semidefinite, selfadjoint solution of the

differential Riccati equation (DRE)

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

[1] e.g. [Lions '71; Lasiecka/Triggiani '00; Bensoussan et al. '92; Pritchard/Salamon '87; Curtain/Zwart '95]

Origin of the Matrix Equations

MPC/LQG design for Nonlinear Optimal Control Problems

nonlinear parabolic PDE with noise

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + B\,u(t) + F\,v(t) \quad \text{for } t > 0, \quad x(0) = x_0 + \eta_0, \\ y(t) &= C\,x(t) + w(t). \end{aligned}$$

Here,

- $v(t)$ is the input noise
- $w(t)$ is the output noise
- η_0 is the noise in the initial condition.

Origin of the Matrix Equations

MPC/LQG design for Nonlinear Optimal Control Problems

nonlinear parabolic PDE with noise

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + B\,u(t) + F\,v(t) \quad \text{for } t > 0, \quad x(0) = x_0 + \eta_0, \\ y(t) &= C\,x(t) + w(t). \end{aligned}$$

Strategy [Benner, Hein (geb. Görner) 2006] (based on [Ito, Kunisch 2006])

1. Linearize the nonlinear state equation on sub-intervals (Model Predictive Control (MPC) or Receding Horizon Control (RHC)).
2. Find estimates of the states (Linear Quadratic Gaussian Design (LQG)) on the sub-intervals.

Origin of the Matrix Equations

MPC/LQG design for Nonlinear Optimal Control Problems

nonlinear parabolic PDE with noise

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + B\,u(t) + F\,v(t) \quad \text{for } t > 0, \quad x(0) = x_0 + \eta_0, \\ y(t) &= C\,x(t) + w(t). \end{aligned}$$

Strategy [Benner, Hein (geb. Görner) 2006] (based on [Ito, Kunisch 2006])

1. Linearize the nonlinear state equation on sub-intervals (Model Predictive Control (MPC) or Receding Horizon Control (RHC)).
2. Find estimates of the states (Linear Quadratic Gaussian Design (LQG)) on the sub-intervals.

Needs the additional solution of the Filter Algebraic Riccati Equation (FARE)

$$0 = A\Sigma + \Sigma A^T - \Sigma C^T W^{-1} C \Sigma + FVF^T.$$

Origin of the Matrix Equations
MPC/LQG design for Nonlinear Optimal Control Problems

nonlinear parabolic PDE with noise

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + B\,u(t) + F\,v(t) \quad \text{for } t > 0, \quad x(0) = x_0 + \eta_0, \\ y(t) &= C\,x(t) + w(t). \end{aligned}$$

Needs the additional solution of the Filter Algebraic Riccati Equation (FARE)

$$0 = A\Sigma + \Sigma A^T - \Sigma C^T W^{-1} C \Sigma + FVF^T.$$

- Here V, W are the symmetric and positive definite covariance matrices.
- Σ is used to compute the best approximation to the state for the feedback loop

Origin of the Matrix Equations

Exponential Stabilization of Navier-Stokes and Oseen Equations

SPP 1253 Project Benner/Bänsch (Researcher: A. Heubner)

[Raymond 2006]

Navier Stokes equation exponentially stabilizable by boundary feedback control for sufficiently small initial conditions.

Origin of the Matrix Equations

Exponential Stabilization of Navier-Stokes and Oseen Equations

SPP 1253 Project Benner/Bänsch (Researcher: A. Heubner)

[RAYMOND 2006]

Navier Stokes equation exponentially stabilizable by boundary feedback control for sufficiently small initial conditions.

Main Ingredients of the proof

- Helmholtz decomposition of the state $y = Py + (I - P)y \Rightarrow$

Origin of the Matrix Equations

Exponential Stabilization of Navier-Stokes and Oseen Equations

SPP 1253 Project Benner/Bänsch (Researcher: A. Heubner)

[Raymond 2006]

Navier Stokes equation exponentially stabilizable by boundary feedback control for sufficiently small initial conditions.

Main Ingredients of the proof

- Helmholtz decomposition of the state $y = Py + (I - P)y \Rightarrow$
  - Cauchy equation for Py
  - elliptic equation for $(I - P)y$

Origin of the Matrix Equations

Exponential Stabilization of Navier-Stokes and Oseen Equations

SPP 1253 Project Benner/Bänsch (Researcher: A. Heubner)

[Raymond 2006]

Navier Stokes equation exponentially stabilizable by boundary feedback control for sufficiently small initial conditions.

Main Ingredients of the proof

- Helmholtz decomposition of the state $y = Py + (I - P)y \Rightarrow$
  - Cauchy equation for Py
  - elliptic equation for $(I - P)y$
- elimination of $(I - P)y$ from the cost functional

Origin of the Matrix Equations

Exponential Stabilization of Navier-Stokes and Oseen Equations

SPP 1253 Project Benner/Bänsch (Researcher: A. Heubner)

[Raymond 2006]

Navier Stokes equation exponentially stabilizable by boundary feedback control for sufficiently small initial conditions.

Main Ingredients of the proof

- Helmholtz decomposition of the state $y = Py + (I - P)y \Rightarrow$
  - Cauchy equation for Py
  - elliptic equation for $(I - P)y$
- elimination of $(I - P)y$ from the cost functional
- LQR design for the Cauchy equation

Origin of the Matrix Equations

Exponential Stabilization of Navier-Stokes and Oseen Equations

SPP 1253 Project Benner/Bänsch (Researcher: A. Heubner)

[RAYMOND 2006]

Navier Stokes equation exponentially stabilizable by boundary feedback control for sufficiently small initial conditions.

origin of saddle-point formulation

- Problem: Test space of divergence free functions not directly FE discretizable.

Origin of the Matrix Equations

Exponential Stabilization of Navier-Stokes and Oseen Equations

SPP 1253 Project Benner/Bänsch (Researcher: A. Heubner)

[Raymond 2006]

Navier Stokes equation exponentially stabilizable by boundary feedback control for sufficiently small initial conditions.

origin of saddle-point formulation

- Problem: Test space of divergence free functions not directly FE discretizable.
- Strategies:
  1. Matrix assembly after Helmholtz projection of the basis functions (expensive for reasonable grids)
  2. projections on matrix level after standard Galerkin discretization following [Heinkenschloss,Sorenson,Sun 2007]

Numerical methods for DRE

Numerical methods for DRE

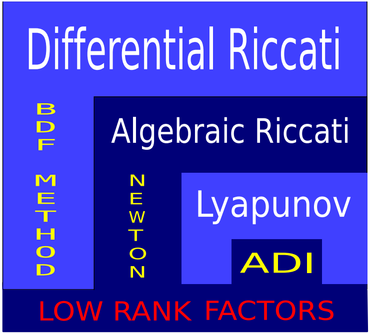
Matrix versions of the ODE solvers

[Mena 2007] showed that ODE solvers of BDF and Rosenbrock type can efficiently be applied to matrix valued problems.

Numerical methods for DRE

Matrix versions of the ODE solvers

[Mena 2007] showed that ODE solvers of BDF and Rosenbrock type can efficiently be applied to matrix valued problems.



Numerical methods for DRE

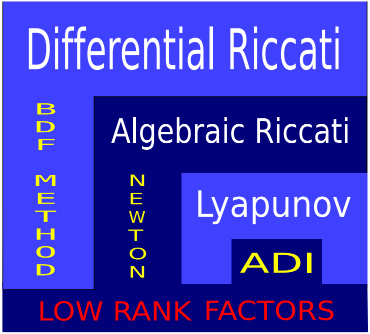
Matrix versions of the ODE solvers

[Mena 2007] showed that ODE solvers of BDF and Rosenbrock type can efficiently be applied to matrix valued problems.

Numerical methods for DRE

Matrix versions of the ODE solvers

[Mena 2007] showed that ODE solvers of BDF and Rosenbrock type can efficiently be applied to matrix valued problems.



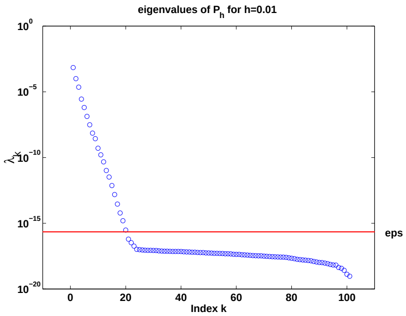
Low Rank Approximation guarantees efficiency in terms of computational effort and memory usage

Motivation of the low rank approximation

The spectrum of an AREs solution

Motivating example

- Linear 1D heat equation with point control.
- $\Omega = [0, 1]$.
- FEM discretization using linear B-splines.
- h=0.01.



$$X = X^T \geq 0 \Longrightarrow X = ZZ^T = \sum_{k=1}^{n} \lambda_k z_k z_k^T \approx \sum_{k=1}^{r} \lambda_k z_k z_k^T = Z_{(r)} Z_{(r)}^T.$$

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

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

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

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

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

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

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

$$\mathfrak{R}'|_X(N_l) = -\mathfrak{R}(X_l), \qquad X_{l+1} = X_l + N_l,$$

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

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

can be rewritten as

one iteration step

$$(A - BB^T X_l)^T X_{l+1} + X_{l+1}(A - BB^T X_l) = -C^T C - X_l BB^T X_l$$

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**[2]**:**
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

$$(H + pI)v = r$$
$$(V + pI)w = t$$

can be solved easily/efficiently.

[2] [Peaceman & Rachford 1954], see also [Wachspress 1966]

LRCF Newton Method for the ARE

Cholesky factor ADI for Lyapunov equations

Recall **Peaceman Rachford ADI**[2]**:**
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 $\Rightarrow \exists p_j,\ j = 1, 2, \ldots J$ 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 $Au = s$.

[2] [Peaceman & Rachford 1954], see also [Wachspress 1966]

LRCF Newton Method for the ARE

Cholesky factor ADI for Lyapunov equations

The Lyapunov operator

$$\mathcal{L}: \quad X \quad \mapsto \quad F^T X + XF$$

can be decomposed into the linear operators

$$\mathcal{L}_H : X \mapsto F^T X \qquad \mathcal{L}_V : X \mapsto XF.$$

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

ADI iteration for the Lyapunov equation (LE)

$$\begin{array}{rcl} X_0 & = & 0 \\ (F^T + p_j I)X_{j-\frac{1}{2}} & = & -GG^T - X_{j-1}(F - p_j I) \\ (F^T + p_j I)X_j^T & = & -GG^T - X_{j-\frac{1}{2}}^T(F - p_j I) \end{array} \qquad \text{(LE-ADI)}$$

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 X_j to exploit $\mathrm{rk}(X_j) \ll n$. [Li & White 2002; Penzl 1999; Benner, Li, 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 X_j to exploit $\mathrm{rk}(X_j) \ll n$. [Li & White 2002; Penzl 1999; Benner, Li, 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. [Penzl 1999; Benner, Li, Penzl 2000]

Recent Improvements in the Software

Recent Improvements in the Software

Reordering Strategies

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 2.0 let's you choose between:

- symmetric reverse Cuthill-McKee (RCM[3]) reordering
- approximate minimum degree (AMD[4]) reordering
- symmetric AMD[4]

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

Recent Improvements in the Software
Reordering Strategies

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 MESS 1.0 let's you choose between:

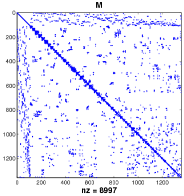
- symmetric reverse Cuthill-McKee (RCM[3]) reordering
- approximate minimum degree (AMD[4]) reordering
- symmetric AMD[4]

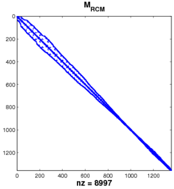
[3] [A. George and J. W.-H. Liu 1981]
[4] [P. Amestoy, T. A. Davis, and I. S. Duff 1996.]

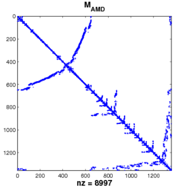
Recent Improvements in the Software

Reordering Strategies

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



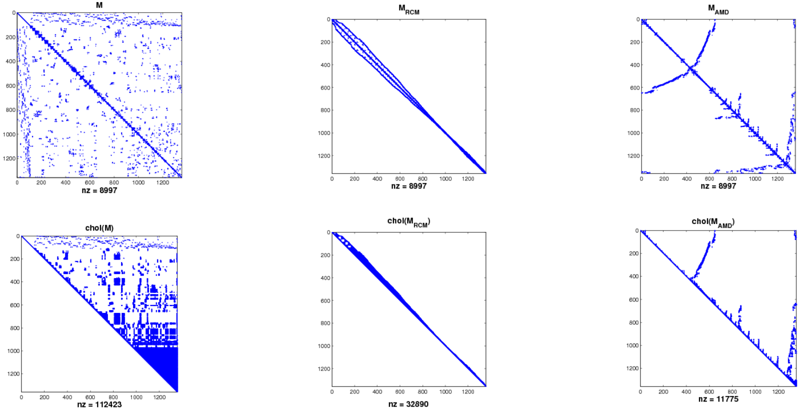




Recent Improvements in the Software

Reordering Strategies

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



Recent Improvements in the Software

ADI Shift Parameters

Optimal ADI parameters solve the

min-max-problem

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

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.

Recent Improvements in the Software

ADI Shift Parameters

Optimal ADI 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 (see also [Sabino 2006])

Recent Improvements in the Software

ADI Shift Parameters

ADI shift parameter choices in upcoming MESS 1.0

1. heuristic parameters [Penzl 1999]
   - use selected Ritz values as shifts
   - suboptimal ⇒ convergence might be slow
   - in general complex for complex spectra
2. approximate Wachspress parameters [Benner, Mena, S. 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"

Recent Improvements in the Software

ADI Shift Parameters

ADI shift parameter choices in upcoming MESS 1.0

1. heuristic parameters [Penzl 1999]
   - use selected Ritz values as shifts
   - suboptimal ⇒ convergence might be slow
   - in general complex for complex spectra
2. approximate Wachspress parameters [Benner, Mena, S. 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"

Recent Improvements in the Software

ADI Shift Parameters

ADI shift parameter choices in upcoming MESS 1.0

1. heuristic parameters [Penzl 1999]
   - use selected Ritz values as shifts
   - suboptimal ⇒ convergence might be slow
   - in general complex for complex spectra
2. approximate Wachspress parameters [Benner, Mena, S. 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"

Recent Improvements in the Software

ADI Shift Parameters

ADI shift parameter choices in upcoming MESS 1.0

1. heuristic parameters [Penzl 1999]
   - use selected Ritz values as shifts
   - suboptimal ⇒ convergence might be slow
   - in general complex for complex spectra
2. approximate Wachspress parameters [Benner, Mena, S. 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”

Recent Improvements in the Software

ADI Shift Parameters

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`)

Recent Improvements in the Software

ADI Shift Parameters

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

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

Recent Improvements in the Software

ADI Shift Parameters

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

Recent Improvements in the Software

Column Compression for the low rank factors

Problem

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

Recent Improvements in the Software

Column Compression for the low rank factors

Problem

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

Idea: Column compression using rank revealing QR factorization (RRQR)

Consider $X = ZZ^T$ and $\text{rk}(Z) = r$. Compute the RRQR[5] 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 $\tilde{Z}\tilde{Z}^T =: \tilde{X} = X$.

[5] [Bischof & Quintana-Ortí 1998]

Recent Improvements in the Software

Column Compression for the low rank factors

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`)

Recent Improvements in the Software

Column Compression for the low rank factors

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

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

Recent Improvements in the Software

Column Compression for the low rank factors

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 & 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

Recent Improvements in the Software

Generalized Systems

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.

Recent Improvements in the Software

Generalized Systems

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)

Recent Improvements in the Software

Generalized Systems

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 (BDF) or 1 Lyapunov (Rosenbrock) 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 (BDF) or 1 Lyapunov (Rosenbrock) 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 [GALLIVAN, RAO, VAN DOOREN 2006] can be implemented exploiting sparse matrix arithmetics.

Conclusions and Outlook

Outlook

Implementation TODOs

- User supplied functions and saddle point solvers for B
- 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.
- ...

Conclusions and Outlook

Outlook

Implementation TODOs

- User supplied functions and saddle point solvers for B (with Anne Heubner in SPP 1253)
- Introduce solvers for DREs (with Hermann Mena (EPN Quito))
- 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.
- ...

Conclusions and Outlook

Outlook

Implementation TODOs

- User supplied functions and saddle point solvers for B (with Anne Heubner in SPP 1253)
- Introduce solvers for DREs (with Hermann Mena (EPN Quito))
- 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.
- ...

Thank you for your attention!