Numerical Solution of Descriptor Riccati Equations for Linearized Navier-Stokes Control Problems

Eberhard Bänsch\textsuperscript{1} \hspace{1cm} Peter Benner\textsuperscript{2,3} \hspace{1cm} Jens Saak\textsuperscript{3}

\textsuperscript{1}Institut für Angewandte Mathematik, Lehrstuhl 3
Friedrich-Alexander Universität Erlangen-Nürnberg, Germany

\textsuperscript{2}Fakultät für Mathematik, Professur Mathematik in Industrie und Technik
Technische Universität Chemnitz, Germany

\textsuperscript{3}Computational Methods in Systems and Control Theory
Max Planck Institute for Dynamics of Complex Technical Systems
Magdeburg, Germany

supported by
Linear Feedback Stabilization of Flow Problems

Eberhard Bänsch\(^1\)  Peter Benner\(^2,3\)  Jens Saak\(^3\)

\(^1\)Institut für Angewandte Mathematik, Lehrstuhl 3
Friedrich-Alexander Universität Erlangen-Nürnberg, Germany

\(^2\)Fakultät für Mathematik, Professur Mathematik in Industrie und Technik
Technische Universität Chemnitz, Germany

\(^3\)Computational Methods in Systems and Control Theory
Max Planck Institute for Dynamics of Complex Technical Systems
Magdeburg, Germany

supported by
Overview

1. Introduction
2. Solving Large-Scale AREs
3. Solving AREs for Linearized Navier-Stokes Eqns.
4. Further Applications
5. Conclusions and Future Work
6. References
Introduction

Optimal control-based stabilization for Navier-Stokes equations

- Stabilization to **steady-state solution** $w$ of flows (with velocity field $v$ and pressure $\chi$), described by **incompressible** Navier-Stokes equations

\[
\begin{align*}
\partial_t v + v \cdot \nabla v - \frac{1}{Re} \Delta v + \nabla \chi &= f \\
\text{div } v &= 0
\end{align*}
\]  

on $Q_\infty := \Omega \times (0, \infty)$, $\Omega \subseteq \mathbb{R}^d$, $d = 2, 3$, with smooth boundary $\Gamma := \partial \Omega$, and boundary and initial conditions

\[
\begin{align*}
v &= g \quad \text{on } \Sigma_\infty := \Gamma \times (0, \infty), \\
v(0) &= w + z(0) \quad (w \text{ given velocity field}).
\end{align*}
\]

- Existence of stabilizing linear state feedback control proved in 2D [Fernández-Cara et al 2004] and 3D [Fursikov 2004].

- Construction of stabilizing feedback control based on associated linear-quadratic optimal control problem:
  - for distributed control, see [Barbu 2003, Barbu/Sritharan 1998, Barbu/Triggiani 2004];
  - for boundary control, see [Barbu/Lasiecka/Triggiani 2006/07] (tangential) and [Raymond 2005–07, Bahdra 2009] (normal).
Introduction

Optimal control-based stabilization for Navier-Stokes equations

- Stabilization to steady-state solution $w$ of flows (with velocity field $v$ and pressure $\chi$), described by incompressible Navier-Stokes equations

$$\begin{align*}
\partial_t v + v \cdot \nabla v - \frac{1}{Re} \Delta v + \nabla \chi &= f \\
\text{div } v &= 0
\end{align*}$$

(1a)

(1b)

on $Q_\infty := \Omega \times (0, \infty)$, $\Omega \subseteq \mathbb{R}^d$, $d = 2, 3$, with smooth boundary $\Gamma := \partial \Omega$, and boundary and initial conditions

- $v = g$ on $\Sigma_\infty := \Gamma \times (0, \infty)$,

- $v(0) = w + z(0)$ ($w$ given velocity field).

- Existence of stabilizing linear state feedback control proved in 2D [Fernández-Cara et al 2004] and 3D [Fursikov 2004].

- Construction of stabilizing feedback control based on associated linear-quadratic optimal control problem:

  - for distributed control, see [Barbu 2003, Barbu/Sritharan 1998, Barbu/Triggiani 2004];

  - for boundary control, see [Barbu/Lasiecka/Triggiani 2006/07] (tangential) and [Raymond 2005–07, Bahdra 2009] (normal).
Assume $w$ solves the stationary Navier-Stokes equations

$$w \cdot \nabla w - \frac{1}{Re} \Delta w + \nabla \chi_s = f, \quad \text{div } w = 0,$$

(2)

with Dirichlet boundary condition $w = g$ on $\Gamma$, $w$ possibly unstable. If we can determine a Dirichlet boundary control $u$ so that the corresponding controlled system for $z := v - w$,

$$\partial_t z + (z \cdot \nabla)w + (w \cdot \nabla)z + (z \cdot \nabla)z - \frac{1}{Re} \Delta z + \nabla p = 0 \quad \text{in } Q_\infty,$$

$$\text{div } z = 0 \quad \text{in } Q_\infty,$$

$$z = bu \quad \text{in } \Sigma_\infty,$$

$$z(0) = z_0 \quad \text{in } \Omega,$$

is stable for “small” initial values $z_0 \in X(\Omega) \subset V^0_n(\Omega)$, where

$$V^0_n(\Omega) := L_2(\Omega) \cap \{\text{div } z = 0\} \cap \{z \cdot n = 0 \text{ on } \Gamma\},$$

then $\exists$ constants $c, \omega > 0$ so that $\|z(t)\|_{X(\Omega)} \leq ce^{-\omega t}$.

$$\Rightarrow \quad \{ \text{Solution to instationary Navier-Stokes equations with } v = w + z, \chi = \chi_s + p, \text{ and } v(0) = w + z_0 \text{ in } \Omega \text{ is controlled to } w. \}$$
Optimal control-based stabilization for NSEs

Analytical solution [Raymond ‘05–’07]

Assume \( w \) solves the stationary Navier-Stokes equations

\[
\begin{align*}
    w \cdot \nabla w - \frac{1}{Re} \Delta w + \nabla \chi_s &= f, \\
    \text{div } w &= 0,
\end{align*}
\]

with Dirichlet boundary condition \( w = g \) on \( \Gamma \), \( w \) possibly unstable.

If we can determine a Dirichlet boundary control \( u \) so that the corresponding controlled system for \( z := v - w \),

\[
\begin{align*}
    \partial_t z + (z \cdot \nabla) w + (w \cdot \nabla) z + (z \cdot \nabla) z - \frac{1}{Re} \Delta z + \nabla p &= 0 \quad \text{in } Q_{\infty}, \\
    \text{div } z &= 0 \quad \text{in } Q_{\infty}, \\
    z &= bu \quad \text{in } \Sigma_{\infty}, \\
    z(0) &= z_0 \quad \text{in } \Omega,
\end{align*}
\]

is stable for “small” initial values \( z_0 \in X(\Omega) \subset V_0^0(\Omega) \), where

\[
V_0^0(\Omega) := L_2(\Omega) \cap \{ \text{div } z = 0 \} \cap \{ z \cdot n = 0 \text{ on } \Gamma \}.
\]

then \( \exists \) constants \( c, \omega > 0 \) so that \( \|z(t)\|_{X(\Omega)} \leq ce^{-\omega t} \).

\[\implies \left\{ \begin{array}{l}
    \text{Solution to instationary Navier-Stokes equations with } v = w + z, \\
    \chi = \chi_s + p, \text{ and } v(0) = w + z_0 \text{ in } \Omega \text{ is controlled to } w.
\end{array} \right.\]
Optimal control-based stabilization for NSEs

Analytical solution [RAYMOND'05–'07]

Assume $w$ solves the stationary Navier-Stokes equations

$$w \cdot \nabla w - \frac{1}{Re} \Delta w + \nabla \chi_s = f, \quad \text{div } w = 0, \quad (2)$$

with Dirichlet boundary condition $w = g$ on $\Gamma$, $w$ possibly unstable.

If we can determine a Dirichlet boundary control $u$ so that the corresponding controlled system for $z := v - w$,

$$\partial_t z + (z \cdot \nabla)w + (w \cdot \nabla)z + (z \cdot \nabla)z - \frac{1}{Re} \Delta z + \nabla p = 0 \quad \text{in } Q_\infty,$$

$$\text{div } z = 0 \quad \text{in } Q_\infty,$$

$$z = bu \quad \text{in } \Sigma_\infty,$$

$$z(0) = z_0 \quad \text{in } \Omega,$$

is stable for “small” initial values $z_0 \in X(\Omega) \subset V_0^0(\Omega)$, where

$$V_0^0(\Omega) := L_2(\Omega) \cap \{\text{div } z = 0\} \cap \{z \cdot n = 0 \text{ on } \Gamma\},$$

then $\exists$ constants $c, \omega > 0$ so that $\|z(t)\|_{X(\Omega)} \leq ce^{-\omega t}$.

$\implies \{ \text{Solution to instationary Navier-Stokes equations with } v = w + z, \chi = \chi_s + p, \text{ and } v(0) = w + z_0 \text{ in } \Omega \text{ is controlled to } w. \}$
Optimal control-based stabilization for NSEs

Analytical solution [RAYMOND '05–'07]

**Linearized Navier-Stokes control system:**

\[
\begin{align*}
\partial_t z + (z \cdot \nabla) w + (w \cdot \nabla) z - \frac{1}{Re} \Delta z - \omega z + \nabla p &= 0 \quad \text{in } Q_\infty, \\
\text{div } z &= 0 \quad \text{in } Q_\infty \\
z &= bu \quad \text{in } \Sigma_\infty \\
z(0) &= z_0 \quad \text{in } \Omega,
\end{align*}
\]

\(\omega z\) with \(\omega > 0\) de-stabilizes the system further, needed to guarantee exponential stabilization, \(\omega\) controls decay rate!

**Cost functional (with \(P = \text{Helmholtz projector}\))**

\[
J(z, u) = \frac{1}{2} \int_0^\infty \langle Pz, Pz \rangle_{L^2(\Omega)} + \rho u(t)^2 \, dt,
\]

the linear-quadratic optimal control problem associated to (3) becomes

\[
\inf \{ J(z, u) \mid (z, u) \text{ satisfies (3), } u \in L^2(0, \infty) \}.
\]
Optimal control-based stabilization for NSEs

Analytical solution [Raymond'05–'07]

Linearized Navier-Stokes control system:

\[
\begin{align*}
\partial_t z + (z \cdot \nabla)w + (w \cdot \nabla)z - \frac{1}{Re} \Delta z - \omega z + \nabla p &= 0 \quad \text{in } Q_\infty \quad (3a) \\
\text{div } z &= 0 \quad \text{in } Q_\infty \quad (3b) \\
z &= bu \quad \text{in } \Sigma_\infty \quad (3c) \\
z(0) &= z_0 \quad \text{in } \Omega, \quad (3d)
\end{align*}
\]

\(\omega z\) with \(\omega > 0\) de-stabilizes the system further, needed to guarantee exponential stabilization, \(\omega\) controls decay rate!

Cost functional (with \(P = \text{Helmholtz projector}\))

\[
J(z, u) = \frac{1}{2} \int_0^\infty \langle Pz, Pz \rangle_{L^2(\Omega)} + \rho u(t)^2 \, dt,
\]

the linear-quadratic optimal control problem associated to (3) becomes

\[
\inf \{J(z, u) \mid (z, u) \text{ satisfies (3), } u \in L^2(0, \infty)\}.
\]
Optimal control-based stabilization for NSEs

Analytical solution [Raymond’05–’07]

Proposition [Raymond ‘05, Bahdra ‘09]

The solution to the instationary Navier-Stokes equations with perturbed initial data is exponentially controlled to the steady-state solution \( w \) by the feedback law

\[
    u = -\rho^{-1} B^* \Pi z_H,
\]

where

- \( z_H := Pz \), with \( P : L_2(\Omega) \mapsto V_n^0(\Omega) \) being the Helmholtz projector \( (\mapsto \text{div } z_H = 0) \);
- \( \Pi = \Pi^* \in \mathcal{L}(V_n^0(\Omega)) \) is the unique nonnegative semidefinite weak solution of the operator Riccati equation

\[
    0 = I + (A + \omega I)^* \Pi + \Pi (A + \omega I) - \Pi (B_\tau B_\tau^* + \rho^{-1} B_n B_n^*) \Pi,
\]

A is the linearized Navier-Stokes operator restricted to \( V_n^0 \);

\( B_\tau \) and \( B_n \) correspond to the projection of the control action in the tangential and normal directions.
Optimal control-based stabilization for NSEs

Analytical solution \[\text{[RAYMOND}’05–’07\]}

**Proposition \[\text{[RAYMOND} ‘05, BAHDRA ‘09\]**

The solution to the instationary Navier-Stokes equations with perturbed initial data is exponentially controlled to the steady-state solution \(w\) by the feedback law

\[u = -\rho^{-1}B^*\Pi z_H,\]

where

- \(z_H := Pz\), with \(P : L_2(\Omega) \mapsto V_0^0(\Omega)\) being the Helmholtz projector \((\mapsto \text{div} z_H \equiv 0)\);
- \(\Pi = \Pi^* \in \mathcal{L}(V_0^0(\Omega))\) is the unique nonnegative semidefinite weak solution of the operator Riccati equation

\[0 = I + (A + \omega I)^*\Pi + \Pi(A + \omega I) - \Pi(B_T B_T^* + \rho^{-1}B_n B_n^*)\Pi,\]

\(A\) is the linearized Navier-Stokes operator restricted to \(V_0^0\);
\(B_T\) and \(B_n\) correspond to the projection of the control action in the tangential and normal directions.
Optimal control-based stabilization for NSEs

Analytical solution \([\text{Raymond} '05–'07]\)

Proposition \([\text{Raymond} '05, \text{Bahdra} '09]\)

The solution to the instationary Navier-Stokes equations with perturbed initial data is exponentially controlled to the steady-state solution \(w\) by the feedback law

\[
u = -\rho^{-1} B^* \Pi z_H,
\]

where

- \(z_H := Pz\), with \(P : L_2(\Omega) \to V^0_n(\Omega)\) being the Helmholtz projector \((\sim \text{div } z_H \equiv 0)\);
- \(\Pi = \Pi^* \in \mathcal{L}(V^0_n(\Omega))\) is the unique nonnegative semidefinite weak solution of the operator Riccati equation

\[
0 = I + (A + \omega I)^* \Pi + \Pi (A + \omega I) - \Pi (B^*_\tau B^*_\tau + \rho^{-1} B_n B^*_n) \Pi,
\]

\(A\) is the linearized Navier-Stokes operator restricted to \(V^0_n\);

\(B_\tau\) and \(B_n\) correspond to the projection of the control action in the tangential and normal directions.
Optimal control-based stabilization for NSEs

Analytical solution [RAYMOND ‘05–’07]

Proposition [RAYMOND ‘05, BAHDRA ‘09]

The solution to the instationary Navier-Stokes equations with perturbed initial data is exponentially controlled to the steady-state solution $w$ by the feedback law

$$u = -\rho^{-1} B^* \Pi z_H,$$

where

- $z_H := Pz$, with $P : L_2(\Omega) \mapsto V^0_n(\Omega)$ being the Helmholtz projector ($\mapsto \text{div } z_H \equiv 0$);
- $\Pi = \Pi^* \in \mathcal{L}(V^0_n(\Omega))$ is the unique nonnegative semidefinite weak solution of the operator Riccati equation

$$0 = I + (A + \omega I)^* \Pi + \Pi (A + \omega I) - \Pi (B^*_\tau B^*_n + \rho^{-1} B_n B^*_n) \Pi,$$

$A$ is the linearized Navier-Stokes operator restricted to $V^0_n$; $B^*_\tau$ and $B^*_n$ correspond to the projection of the control action in the tangential and normal directions.
Long-Term Plan

Apply optimal control-based feedback stabilization to (multi-)field problems with increasing complexity:

- **Proof of concept:** Navier-Stokes with normal boundary control for model problem (von Kármán vortex shedding).
- Navier-Stokes coupled with (passive) transport of (reactive) species.
- Phase transition liquid/solid with convection.
- Stabilization of a flow with a free capillary surface.
- Control for electrically conducting fluids in presence of outer magnetic fields (MHD).

All scenarios require

- formulation as abstract parabolic Cauchy problem,
- definition of quadratic cost functional,
- formulation of corresponding ARE,
- spatial discretization (FEM),
- numerical solution of large-scale ARE.
Long-Term Plan

Apply optimal control-based feedback stabilization to (multi-)field problems with increasing complexity:

- **Proof of concept:** Navier-Stokes with normal boundary control for model problem (von Kármán vortex shedding).
- Navier-Stokes coupled with (passive) transport of (reactive) species.
- Phase transition liquid/solid with convection.
- Stabilization of a flow with a free capillary surface.
- Control for electrically conducting fluids in presence of outer magnetic fields (MHD).

All scenarios require

- formulation as abstract parabolic Cauchy problem,
- definition of quadratic cost functional,
- formulation of corresponding ARE,
- spatial discretization (FEM),
- numerical solution of large-scale ARE.
Introduction

Proof of concept: von Kármán vortex street

von Kármán vortex street \((Re = 300)\)

Vortex suppression by blowing in at upper end of cylinder, without . . .

and with stabilizing feedback

Computations by Heiko Weichelt
Consider

\[ 0 = \mathcal{R}(X) := C^T C + A^T X + X A - X B B^T X \]

Re-write Newton’s method for AREs \((A_j := A - B B^T X_j)\)

\[ D \mathcal{R}(X_j) (N_j) = -\mathcal{R}(X_j) \]

\[ \iff \]

\[ A_j^T (X_j + N_j) + (X_j + N_j) A_j = -C^T C - X_j B B^T X_j =: -W_j W_j^T \]

Set \( X_j = Z_j Z_j^T \) for rank \((Z_j) \ll n \Rightarrow \)

\[ A_j^T (Z_{j+1} Z_{j+1}^T) + (Z_{j+1} Z_{j+1}^T) A_j = -W_j W_j^T \]

**Factored Newton Iteration \([B./L/Penzl '99/’08]\)**

Solve Lyapunov equations for \(Z_{j+1}\) directly by factored ADI iteration and exploit ‘sparse + low-rank’ structure of \(A_j\).
Consider

\[ 0 = \mathcal{R}(X) := C^T C + A^T X + X A - X B B^T X \]

Re-write Newton’s method for AREs (\( A_j := A - B B^T X_j \))

\[
A_j^T N_j + N_j A_j = -\mathcal{R}(X_j)
\]

\[
\iff
A_j^T (X_j + N_j) + (X_j + N_j) A_j = -C^T C - X_j B B^T X_j =: -W_j W_j^T
\]

Set \( X_j = Z_j Z_j^T \) for rank (\( Z_j \)) \( \ll n \)

\[
A_j^T (Z_{j+1} Z_{j+1}^T) + (Z_{j+1} Z_{j+1}^T) A_j = -W_j W_j^T
\]

Factored Newton Iteration \([\text{B./Li/Penzl '99/’08}]\)

Solve Lyapunov equations for \( Z_{j+1} \) directly by factored ADI iteration and exploit ‘sparse + low-rank’ structure of \( A_j \).
Consider

\[ 0 = \mathcal{R}(X) := C^T C + A^T X + X A - X B B^T X \]

Re-write Newton’s method for AREs \((A_j := A - B B^T X_j)\)

\[
A_j^T N_j + N_j A_j = -\mathcal{R}(X_j)
\]

\[
\iff
A_j^T (X_j + N_j) + (X_j + N_j) A_j = -C^T C - X_j B B^T X_j
\]

Set \(X_j = Z_j Z_j^T\) for \(\text{rank}(Z_j) \ll n\)

\[
A_j^T (Z_{j+1} Z_{j+1}^T) + (Z_{j+1} Z_{j+1}^T) A_j = -W_j W_j^T
\]

Factored Newton Iteration \([B./L/Penzl '99/’08]\)

Solve Lyapunov equations for \(Z_{j+1}\) directly by factored ADI iteration and exploit ‘sparse + low-rank’ structure of \(A_j\).
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 $X$!
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 \( X \)!
Review: LRCF-ADI for Lyapunov Equations

Consider

\[ FX + XF^T = -GG^T \]

ADI iteration for the Lyapunov equation (LE) \[\text{[WACHSPRESS '95]}\]

For \( j = 1, \ldots, J \)

\[
\begin{align*}
X_0 &= 0 \\
(F + p_j I)X_{j-1}^{1/2} &= -GG^T - X_{j-1}(F^T - p_j I) \\
(F + p_j I)X_j^{1/2} &= -GG^T - X_{j-1}^{1/2}(F^T - p_j I)
\end{align*}
\]

Rewrite as one step iteration and factorize \( X_i = Z_iZ_i^T, i = 0, \ldots, J \)

\[
\begin{align*}
Z_0Z_0^T &= 0 \\
Z_jZ_j^T &= -2p_j(F + p_j I)^{-1}GG^T(F + p_j I)^{-T} \\
&+ (F + p_j I)^{-1}(F - p_j I)Z_{j-1}Z_{j-1}^{T}(F - p_j I)^T(F + p_j I)^{-T}
\end{align*}
\]

\( \ldots \Rightarrow \) low-rank Cholesky factor ADI

\[\text{[PENZL '97/'00, LI/WHITE '99/'02, B./LI/PENZL '99/'08, GUGERCIN/SORENSEN/ANTOULAS '03]}\]
Review: LRCF-ADI for Lyapunov Equations

Consider

\[ FX + XF^T = -GG^T \]

**ADI iteration for the Lyapunov equation (LE)**  

For \( j = 1, \ldots, J \)

\[
\begin{align*}
X_0 &= 0 \\
(F + p_j I)X_{j-1/2} &= -GG^T - X_{j-1}(F^T - p_j I) \\
(F + p_j I)X_j^T &= -GG^T - X_{j-1}^T(F^T - p_j I)
\end{align*}
\]

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

\[
\begin{align*}
Z_0 Z_0^T &= 0 \\
Z_j Z_j^T &= -2p_j(F + p_j I)^{-1}GG^T(F + p_j I)^{-T} \\
&+ (F + p_j I)^{-1}(F - p_j I)Z_{j-1}Z_{j-1}^T(F - p_j I)^T(F + p_j I)^{-T}
\end{align*}
\]

\[ \cdots \rightarrow \text{low-rank Cholesky factor ADI} \]

\[ \text{[Penzl '97/'00, Li/White '99/'02, B./Li/Penzl '99/'08, Gugercin/Sorensen/Antoulas '03]} \]
Review: LRCF-ADI for Lyapunov Equations

Consider

\[ FX + XF^T = -GG^T \]

ADI iteration for the Lyapunov equation (LE) \[ \text{[WACHSPRESS '95]} \]

For \( j = 1, \ldots, J \)

\[
\begin{align*}
X_0 &= 0 \\
(F + p_j I)X_{j-\frac{1}{2}} &= -GG^T - X_{j-1}(F^T - p_j I) \\
(F + p_j I)X_{j}^T &= -GG^T - X_{j-\frac{1}{2}}^T(F^T - p_j I)
\end{align*}
\]

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

\[
\begin{align*}
Z_0 Z_0^T &= 0 \\
Z_j Z_j^T &= -2p_j (F + p_j I)^{-1} GG^T(F + p_j I)^{-T} \\
&+ (F + p_j I)^{-1}(F - p_j I)Z_{j-1}Z_{j-1}^T(F - p_j I)^T(F + p_j I)^{-T}
\end{align*}
\]

\( \ldots \leadsto \) low-rank Cholesky factor ADI

\[ \text{[PENZL '97/'00, LI/WHITE '99/'02, B./LI/PENZL '99/'08, GUGERCIN/SORENSEN/ANTOULAS '03]} \]
Review: LRCF-ADI for Lyapunov Equations

The Work Horse

**Algorithm 1** Low-rank Cholesky factor ADI iteration (LRCF-ADI)

\[ \text{[Penzl '97/'00, Li/White '99/'02, B./Li/Penzl '99/'08]} \]

**Input:** $F, G$ defining $FX + XF^T = -GG^T$ and shifts $\{p_1, \ldots, p_{i_{\text{max}}}\}$

**Output:** $Z = Z_{i_{\text{max}}} \in \mathbb{C}^{n \times t_{i_{\text{max}}}}$, such that $ZZ^H \approx X$

1: Solve $(F + p_1 I)V_1 = \sqrt{-2\Re(p_1)G}$ for $V_1$.
2: $Z_1 = V_1$
3: for $i = 2, 3, \ldots, i_{\text{max}}$ do
4: Solve $(F + p_i I)\tilde{V} = V_{i-1}$ for $\tilde{V}$.
5: $V_i = \sqrt{\Re(p_i)/\Re(p_{i-1})} \left( V_{i-1} - (p_i + \overline{p_{i-1}})\tilde{V} \right)$
6: $Z_i = [Z_{i-1} \ V_i]$
7: end for
Krylov Subspace Based Solvers for Lyapunov Equations

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

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

on $\text{colspan}(U_m)$ and get $X_m$ as

$$X_m = U_m Y_m U_m^T.$$
Krylov Subspace Based Solvers for Lyapunov Equations

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

\[ (U_m^T F U_m) Y_m + Y_m (U_m^T F^T U_m) = -U_m^T G G^T U_m, \quad (6) \]

on $\text{colspan}(U_m)$ and get $X_m$ as

\[ X_m = U_m Y_m U_m^T. \]
Krylov Subspace Based Solvers for Lyapunov Equations

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

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

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

$$Z_m = U_m \tilde{Z}_m.$$
Krylov Subspace Based Solvers for Lyapunov Equations
Basic Algorithm

**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., \( \mathcal{V} = \mathcal{K}_p(F, G) \) or \( \mathcal{V} = \mathcal{K}_p(F, G) \cup \mathcal{K}_p(F^{-1}, G) \) with orthogonal basis \( V \in \mathbb{C}^{n \times p} \).

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

```markdown
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:

  \[
  (V^T F V) \tilde{Z} \tilde{Z}^T + \tilde{Z} \tilde{Z}^T (V^T F^T V) = -V^T G G^T V,
  \]

  Lift \( \tilde{Z} \) to the full space: \( Z = V \tilde{Z} \)

until \( \text{res}(Z) < \text{TOL} \)
```

\(^1\)\( (K-PIK, [Simoncini '07]) \)
LRCF-ADI with Galerkin Projection
ADI and Rational Krylov

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

\[ \mathcal{L}(F, G, p) := \text{span} \left\{ \ldots, \prod_{i=-j}^{-1} (F + p_i I)^{-1} G, \ldots, (F + p_{-2} I)^{-1} (F + p_{-1} I)^{-1} G, (F + p_{-1} I)^{-1} G, (F + p_1 I) G, (F + p_2 I)(F + p_1 I) G, \ldots, \prod_{i=1}^{j} (F + p_i I) G \ldots \right\} \]

**Idea**

Solve on current subspace of $\mathcal{L}(F, G, p)$ in the ADI step to increase the quality of the iterate.
LRCF-ADI with Galerkin Projection
ADI and Rational Krylov

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

\[ \mathcal{L}(F, G, p) := \text{span} \left\{ \ldots, \prod_{i=-j}^{-1} (F + p_i I)^{-1} G, \ldots, (F + p_{-2} I)^{-1} (F + p_{-1} I)^{-1} G, (F + p_{-1} I)^{-1} G, G, (F + p_1 I)G, \ldots, \prod_{i=1}^{j} (F + p_i I)G \ldots \right\} \]

Idea

Solve on current subspace of \( \mathcal{L}(F, G, p) \) in the ADI step to increase the quality of the iterate.
## LRCF-ADI with Galerkin Projection

**Projected ADI Step → LRCF-ADI-GP**

1. Compute the LRCF-ADI iterate $Z_i$
2. Compute orthogonal basis via RRQR factorization$^a$: $Q_iR_i\Pi_i = Z_i$
3. Solve (for $\tilde{Z}$) the projected Lyapunov equation
   \[
   (Q_i^T F Q_i)\tilde{Z}\tilde{Z}^T + \tilde{Z}\tilde{Z}^T (Q_i^T F^T Q_i) = -Q_i^T GG^T Q_i
   \]
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.

- 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. $k = 5$) $\leadsto$ restarted ADI with shifts $\Lambda(Q_i^T F Q_i)$. 

---

**Solving Large-Scale AREs**

- Solving AREs for lin. NSE
- Further Applications
- Conclusions
- References
**LRCF-ADI with Galerkin Projection**

**Projected ADI Step → LRCF-ADI-GP**

1. Compute the LRCF-ADI iterate $Z_i$
2. Compute orthogonal basis via RRQR factorization
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. $k = 5$) $\Rightarrow$ restarted ADI with shifts $\Lambda(Q_i^T F Q_i)$.
LRCF-ADI with Galerkin Projection

Test Example: Optimal Cooling of Steel Profiles

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

\[
c \cdot \rho \frac{\partial x}{\partial t} = \lambda \Delta x, \quad \xi \in \Omega
\]

\[
\lambda \frac{\partial x}{\partial n} = \kappa (u_k - x), \quad \xi \in \Gamma_k, \ 1 \leq k \leq 7,
\]

\[
\frac{\partial x}{\partial n} = 0, \quad \xi \in \Gamma_0.
\]

\[\Rightarrow q = 7, \ p = 6.\]

- FEM Discretization, different models for initial mesh \(n = 371\),
1, 2, 3, 4 steps of mesh refinement \(\Rightarrow n = 1357, 5177, 20209, 79841\).

Source: Physical model: courtesy of Mannesmann/Demag.
Math. model: Tröltzsch/Unger '99/'01, Penzl '99, S. '03.
LRCF-ADI with Galerkin Projection

Numerical Results

Steel profile $n=20$ 209 good shifts
LRCF-ADI with Galerkin Projection

Numerical Results

Steel profile $n=20$ 209 good shifts

![Computation times graph](chart.png)
LRCF-ADI with Galerkin Projection

Numerical Results

Steel profile n=20 209 bad shifts
LRCF-ADI with Galerkin Projection

Numerical Results

Steel profile $n=20$ 209 bad shifts

![Bar graph showing computation times for different Galerkin projection frequencies.](image-url)
Solving Large-Scale AREs

LRCF-NM for the ARE

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

Newton’s Iteration for the ARE

\[ \mathcal{R}'|_X(N_\ell) = -\mathcal{R}(X_\ell), \quad X_{\ell+1} = X_\ell + N_\ell, \quad \ell = 0, 1, \ldots \]

where the Frechét derivative of \( \mathcal{R} \) at \( X \) is the Lyapunov operator

\[ \mathcal{R}'|_X : Q \mapsto (A - BB^T X)^T Q + Q(A - BB^T X), \]

i.e., in every Newton step solve a

Lyapunov Equation \[[\text{Kleinman '68}]\]

\[ F_\ell^T X_{\ell+1} + X_{\ell+1} F_\ell = -G_\ell G_\ell^T, \]

where \( F_\ell := A - BB^T X_\ell, \ G := [-C^T, -X_\ell B] \).
Factored Newton-Kleinman Iteration

\[ F_\ell = A - BB^T X_\ell =: A - BK_\ell \]
\[ G_\ell = [ C^T, K_\ell^T ] \]

- Apply LRCF-ADI in every Newton step;
- exploit structure of \( F_\ell \) using Sherman-Morrison-Woodbury formula:

\[
(A - BK_\ell + p_k^{(\ell)} I_n)^{-1} = \\
(l_n + (A + p_k^{(\ell)} I_n)^{-1} B (l_m - K_\ell (A + p_k^{(\ell)} I_n)^{-1} B)^{-1} K_\ell) (A + p_k^{(\ell)} I_n)^{-1}
\]
Solving Large-Scale AREs

LRCF-NM for the ARE

Factored Newton-Kleinman Iteration

\[ F_\ell = A - BB^TX_\ell =: A - BK_\ell \]
\[ G_\ell = [C^T, K_\ell^T] \]

is “sparse + low rank”,
is low rank factor.

- Apply LRCF-ADI in every Newton step;
- exploit structure of \( F_\ell \) using Sherman-Morrison-Woodbury formula:

\[
(A - BK_\ell + p_k^{(\ell)}I_n)^{-1} = \\
(I_n + (A + p_k^{(\ell)}I_n)^{-1}B(I_m - K_\ell(A + p_k^{(\ell)}I_n)^{-1}B)^{-1}K_\ell)(A + p_k^{(\ell)}I_n)^{-1}
\]
Solving Large-Scale AREs

LRCF-NM for the ARE

Factored Newton-Kleinman Iteration

\[ F_\ell = A - BB^T X_\ell =: A - BK_\ell \]

\[ G_\ell = [ C^T, K_\ell^T ] \]

is “sparse + low rank”,

is low rank factor.

Apply LRCF-ADI in every Newton step;

exploit structure of \( F_\ell \) using Sherman-Morrison-Woodbury formula:

\[
(A - BK_\ell + p_k^{(\ell)} I_n)^{-1} = \\
(I_n + (A + p_k^{(\ell)} I_n)^{-1} B (I_m - K_\ell (A + p_k^{(\ell)} I_n)^{-1} B)^{-1} K_\ell) (A + p_k^{(\ell)} I_n)^{-1}
\]
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: $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of

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

1: for $k = 1, 2, \ldots, k_{\text{max}}$ do
2: Determine (sub)optimal ADI shift parameters $p^{(k)}_1, p^{(k)}_2, \ldots$ 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) 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
Algorithm 4 Simplified 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

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

1. Determine (sub)optimal ADI shift parameters $p_1, p_2, \ldots$
   with respect to $F(0) = A^T - K(0)B^T$ or $F(\infty) = \lim_{k \to \infty} F(k)$.
2. for $k = 1, 2, \ldots, k_{max}$ do
3. \hspace{1em} $G(k) = \begin{bmatrix} C^T & K(k-1) \end{bmatrix}$
4. \hspace{1em} Compute $Z(k)$ using Algorithm 1 (LRCF-ADI) or (LRCF-ADI-GP)
   such that $F(k)Z(k)Z(k)^H + Z(k)Z(k)^HF(k)^T \approx -G(k)G(k)^T$.
5. \hspace{1em} $K(k) = Z(k)(Z(k)^H)B$
6. end for
LRCF-NM for the ARE

Algorithms

**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:** $Z = Z^{(k_{\text{max}})}$, such that $ZZ^H$ approximates the solution $X$ of

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

1: \textbf{for} $k = 1, 2, \ldots, k_{\text{max}}$ \textbf{do}

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

4: \hspace{1em} 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: \hspace{1em} Project ARE, solve and prolongate solution.

6: \hspace{1em} $K^{(k)} = Z^{(k)}(Z^{(k)H}B)$

7: \textbf{end for}
Optimal feedback

\[ K_\star = B^T X_\star = B^T Z_\star Z_\star^T \]

can be computed by direct feedback iteration:

- \( j \)th Newton iteration:
  
  \[ K_j = B^T Z_j Z_j^T = \sum_{k=1}^{k_{\text{max}}} (B^T V_{j,k}) V_{j,k}^T \xrightarrow{j \to \infty} K_\star = B^T Z_\star Z_\star^T \]

- \( K_j \) can be updated in ADI iteration, \( A_j = BK_j \)
  
  \( \Rightarrow \) no need to form \( Z_j \), need only fixed workspace for \( K_j \in \mathbb{R}^{m \times n}! \)

Related to earlier work by [Banks/Ito '91].
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_11
- 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 $A$
Feedback Iteration

Test Results (ADI-loop): Example 1

<table>
<thead>
<tr>
<th>Newton-ADI</th>
<th>rel. change</th>
<th>rel. residual</th>
<th>ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>9.97 \cdot 10^{-01}</td>
<td>9.27 \cdot 10^{-01}</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>3.67 \cdot 10^{-02}</td>
<td>9.58 \cdot 10^{-02}</td>
<td>94</td>
</tr>
<tr>
<td>3</td>
<td>1.36 \cdot 10^{-02}</td>
<td>1.09 \cdot 10^{-03}</td>
<td>98</td>
</tr>
<tr>
<td>4</td>
<td>3.48 \cdot 10^{-04}</td>
<td>1.01 \cdot 10^{-07}</td>
<td>97</td>
</tr>
<tr>
<td>5</td>
<td>6.41 \cdot 10^{-08}</td>
<td>1.34 \cdot 10^{-10}</td>
<td>97</td>
</tr>
<tr>
<td>6</td>
<td>7.47 \cdot 10^{-16}</td>
<td>1.34 \cdot 10^{-10}</td>
<td>97</td>
</tr>
</tbody>
</table>

CPU time: 4805.8 sec.

<table>
<thead>
<tr>
<th>Newton-Galerkin-ADI</th>
<th>rel. change</th>
<th>rel. residual</th>
<th>ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRCF-ADI-GP(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NWT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>9.97 \cdot 10^{-01}</td>
<td>9.29 \cdot 10^{-01}</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>3.67 \cdot 10^{-02}</td>
<td>9.60 \cdot 10^{-02}</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>1.36 \cdot 10^{-02}</td>
<td>1.09 \cdot 10^{-03}</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>3.47 \cdot 10^{-04}</td>
<td>1.01 \cdot 10^{-07}</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>6.41 \cdot 10^{-08}</td>
<td>1.03 \cdot 10^{-10}</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>1.23 \cdot 10^{-11}</td>
<td>1.98 \cdot 10^{-11}</td>
<td>27</td>
</tr>
</tbody>
</table>

CPU time: 1460.1 sec.

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

Test Results (ADI-loop): Example 2

<table>
<thead>
<tr>
<th>Newton-ADI</th>
<th>Newton-Galerkin-ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NWT</strong></td>
<td><strong>rel. change</strong></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>$2.88 \cdot 10^{-01}$</td>
</tr>
<tr>
<td>3</td>
<td>$2.13 \cdot 10^{-01}$</td>
</tr>
<tr>
<td>4</td>
<td>$1.77 \cdot 10^{-01}$</td>
</tr>
<tr>
<td>5</td>
<td>$2.47 \cdot 10^{-01}$</td>
</tr>
<tr>
<td>6</td>
<td>$3.04 \cdot 10^{-01}$</td>
</tr>
<tr>
<td>7</td>
<td>$1.78 \cdot 10^{-02}$</td>
</tr>
<tr>
<td>8</td>
<td>$2.60 \cdot 10^{-05}$</td>
</tr>
<tr>
<td>9</td>
<td>$2.75 \cdot 10^{-11}$</td>
</tr>
</tbody>
</table>

CPU time: **493.81 sec.**

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: $10^{-10}$
## Feedback Iteration

### Test Results (both-loops): Example 1

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

**CPU time:** 4805.8 sec.

<table>
<thead>
<tr>
<th>NG-ADI</th>
<th>inner = 5, outer = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWT</td>
<td>rel. change</td>
</tr>
<tr>
<td>1</td>
<td>$9.98 \cdot 10^{-01}$</td>
</tr>
<tr>
<td>CPU time: 497.6 sec.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NG-ADI</th>
<th>inner = 1, outer = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWT</td>
<td>rel. change</td>
</tr>
<tr>
<td>1</td>
<td>$9.98 \cdot 10^{-01}$</td>
</tr>
<tr>
<td>CPU time: 856.6 sec.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NG-ADI</th>
<th>inner = 0, outer = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWT</td>
<td>rel. change</td>
</tr>
<tr>
<td>1</td>
<td>$9.98 \cdot 10^{-01}$</td>
</tr>
<tr>
<td>CPU time: 506.6 sec.</td>
<td></td>
</tr>
</tbody>
</table>

test system: Intel® Xeon® 5160 3.00GHz ; 16 GB RAM; 64Bit-MATLAB (R2010a) using threaded BLAS (romulus)

stopping criterion tolerances: $10^{-10}$
## Feedback Iteration

**Test Results (both-loops): Example 2**

### Newton-ADI

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

CPU time: 493.81 sec.

### NG-ADI \ inner= 5, outer= 1

<table>
<thead>
<tr>
<th>NWT</th>
<th>rel. change</th>
<th>rel. residual</th>
<th>ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3.30 \cdot 10^{-11}</td>
<td>35</td>
</tr>
</tbody>
</table>

CPU time: 24.1 sec.

### NG-ADI \ inner= 1, outer= 1

<table>
<thead>
<tr>
<th>NWT</th>
<th>rel. change</th>
<th>rel. residual</th>
<th>ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.31 \cdot 10^{-11}</td>
<td>34</td>
</tr>
</tbody>
</table>

CPU time: 26.8 sec.

### NG-ADI \ inner= 0, outer= 1

<table>
<thead>
<tr>
<th>NWT</th>
<th>rel. change</th>
<th>rel. residual</th>
<th>ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3.27 \cdot 10^{-15}</td>
<td>46</td>
</tr>
</tbody>
</table>

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: \(10^{-10}\)
Feedback Iteration

Computation Time Scales Linearly with Problem Size

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

**Control operator:** Here \( b(\xi) = 4 \left( 1 - \xi_2 \right) \xi_2 \) for \( \xi \in \Gamma_c \) and 0 otherwise.

**Output equation:** \( y = C x \), where

\[
C : L^2(\Omega) \to \mathbb{R} \\
x(\xi, t) \mapsto y(t) = \int_{\Omega} x(\xi, t) \, d\xi, \quad \Rightarrow C_h = 1 \cdot M_h.
\]

**Cost functional:**

\[
J(u) = \int_{0}^{\infty} y^2(t) + u^2(t) \, dt.
\]
Feedback Iteration

Scaling results

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: $10^{-10}$
# Feedback Iteration

## Scaling results

### Computation Times

<table>
<thead>
<tr>
<th>discretization level</th>
<th>problem size</th>
<th>time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>81</td>
<td>$5.53 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>4</td>
<td>289</td>
<td>$1.33 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>5</td>
<td>1089</td>
<td>$2.84 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>6</td>
<td>4225</td>
<td>$1.51 \cdot 10^{0}$</td>
</tr>
<tr>
<td>7</td>
<td>16641</td>
<td>$9.52 \cdot 10^{0}$</td>
</tr>
<tr>
<td>8</td>
<td>66049</td>
<td>$5.97 \cdot 10^{1}$</td>
</tr>
<tr>
<td>9</td>
<td>263169</td>
<td>$4.72 \cdot 10^{2}$</td>
</tr>
<tr>
<td>10</td>
<td>1050625</td>
<td>$6.89 \cdot 10^{3}$</td>
</tr>
<tr>
<td>11</td>
<td>4198401</td>
<td>$8.08 \cdot 10^{4}$</td>
</tr>
</tbody>
</table>

(Finest level: **8.813.287.577.601 unknowns**, taking symmetry into account.)

Test system: Intel® Xeon® 5160 @ 3.00 GHz; 16 GB RAM; 64Bit-MATLAB (R2010a) using threaded BLAS (*romulus*), stopping criterion tolerances: $10^{-10}$
Feedback Iteration

Scaling results

test system: Intel® Xeon® 5160 @ 3.00 GHz; 16 GB RAM; 64Bit-MATLAB (R2010a) using threaded BLAS (romulus)

stopping criterion tolerances: $10^{-10}$
Solving AREs for Linearized Navier-Stokes Eqns.

\[ 0 = M + (A + \omega M)^T X + X (A + \omega M) - MXBB^T XM \]

### Problems with Newton-Kleinman

1. Discretization of Helmholtz-projected linearized Navier-Stokes equations would need divergence-free finite elements.

   Here, we want to use standard discretization (Taylor-Hood elements available in flow solver NAVIER).

   **Explicit projection** of ansatz functions possible using application of Helmholtz projection, but too expensive in general.

2. Each step of Newton-Kleinman iteration: solve

   \[
   A_j^T Z_{j+1} Z_{j+1}^T M + M Z_{j+1} Z_{j+1}^T A_j = - M - K_j^T K_j
   \]

   \( n_v := \text{rank} (M) = \text{dim of ansatz space for velocities.} \)

   \( \rightsquigarrow \) need to solve \( n_v + m \) linear systems of equations in each step of Newton-ADI iteration!

3. Linearized system (i.e., \( A + \omega M \)) is unstable in general.

   But to start Newton iteration, a stabilizing initial guess is needed!
Solving AREs for Linearized Navier-Stokes Eqns.

0 = M + (A + ωM)^T X + X(A + ωM) − MXBB^T XM

Problems with Newton-Kleinman

1. Discretization of Helmholtz-projected linearized Navier-Stokes equations would need divergence-free finite elements. Here, we want to use standard discretization (Taylor-Hood elements available in flow solver NAVIER).

   Explicit projection of ansatz functions possible using application of Helmholtz projection, but too expensive in general.

2. Each step of Newton-Kleinman iteration: solve

   \[ A_j^T Z_{j+1} Z_{j+1}^T M + MZ_{j+1} Z_{j+1}^T A_j = -M - K_j^T K_j \]

   \[ n_v := \text{rank}(M) = \text{dim of ansatz space for velocities}. \]

   \[ \Rightarrow \text{need to solve } n_v + m \text{ linear systems of equations in each step of Newton-ADI iteration!} \]

3. Linearized system (i.e., A + ωM) is unstable in general.

   But to start Newton iteration, a stabilizing initial guess is needed!
Solving AREs for Linearized Navier-Stokes Eqns.

\[ 0 = M + (A + \omega M)^T X + X(A + \omega M) - MXBB^T XM \]

### Problems with Newton-Kleinman

1. Discretization of Helmholtz-projected linearized Navier-Stokes equations would need divergence-free finite elements.

   Here, we want to use standard discretization
   (Taylor-Hood elements available in flow solver \textsc{Navier}).

   Explicit projection of ansatz functions possible using application of Helmholtz projection, but too expensive in general.

2. Each step of Newton-Kleinman iteration: solve

   \[ A_j^T Z_{j+1} Z_{j+1}^T M + MZ_{j+1} Z_{j+1}^T A_j = -M - K_j^T K_j \]

   \( n_v := \text{rank}(M) = \text{dim of ansatz space for velocities.} \)

   \( \leadsto \) need to solve \( n_v + m \) linear systems of equations in each step of Newton-ADI iteration!

3. Linearized system (i.e., \( A + \omega M \)) is unstable in general.

   But to start Newton iteration, a stabilizing initial guess is needed!
Solving AREs for Linearized Navier-Stokes Eqns.

\[ 0 = M + (A + \omega M)^T X + X(A + \omega M) - MXBB^T XM \]

Problems with Newton-Kleinman

1. Discretization of Helmholtz-projected linearized Navier-Stokes equations would need divergence-free finite elements.
   Here, we want to use standard discretization (Taylor-Hood elements available in flow solver \texttt{Navier}).
   Explicit projection of ansatz functions possible using application of Helmholtz projection, but too expensive in general.

2. Each step of Newton-Kleinman iteration: solve

\[ A_j Z_{j+1} + Z_{j+1} M + M Z_{j+1} Z_{j+1}^T M = -M K_j^T K_j \]

3. Linearized system (i.e., \( A + \omega M \)) is unstable in general.
   But to start Newton iteration, a stabilizing initial guess is needed!

References


[Hein '10] \textit{MPC/LQG-Based Optimal Control of Nonlinear Parabolic PDEs}; PhD thesis Chemnitz UT.
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 1. Problem/no need for divergence free FE

- Incompressible Navier-Stokes-Equations

\[
\frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\text{Re}} \Delta \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p = 0 \quad + \text{B.C.} \quad \nabla \cdot \mathbf{v} = 0 
\]  

(NSE)

- Spatial FE discretization

\[
M \dot{\mathbf{v}}(t) = K(\mathbf{v})\mathbf{v}(t) - Gp(t) + B_1 \mathbf{u}(t) \\
0 = G^T \mathbf{v}(t) 
\]  

(dNSE)

- Linearization and change of notation

\[
E_{11} \dot{\mathbf{v}}(t) = A_{11} \mathbf{v}(t) + A_{12} p(t) + B_1 \mathbf{u}(t) \\
0 = A_{12}^T \mathbf{v}(t) 
\]  

(NSDAE)
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 1. Problem/no need for divergence free FE

- Incompressible Navier-Stokes-Equations

\[
\frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\text{Re}} \Delta \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p = 0 \quad + \text{B.C.}
\]
\[
\nabla \cdot \mathbf{v} = 0
\]

\(\text{(NSE)}\)

- Spatial FE discretization

\[
M \dot{\mathbf{v}}(t) = K(\mathbf{v}) \mathbf{v}(t) - Gp(t) + B_1 \mathbf{u}(t)
\]
\[
0 = G^T \mathbf{v}(t)
\]

\(\text{(dNSE)}\)

- Linearization and change of notation

\[
E_{11} \dot{\mathbf{v}}(t) = A_{11} \mathbf{v}(t) + A_{12} p(t) + B_1 \mathbf{u}(t)
\]
\[
0 = A_{12}^T \mathbf{v}(t)
\]

\(\text{(NSDAE)}\)
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 1. Problem/no need for divergence free FE

- incompressible Navier-Stokes-Equations

\[
\frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\text{Re}} \Delta \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla \mathbf{p} = 0 \quad + \text{B.C.} \quad (\text{NSE}) \\
\n\n\n\n
- Spatial FE discretization

\[
M \dot{\mathbf{v}}(t) = K(\mathbf{v}) \mathbf{v}(t) - G \mathbf{p}(t) + B_1 \mathbf{u}(t) \\
0 = G^T \mathbf{v}(t) \quad (\text{dNSE})
\]

- Linearization and change of notation

\[
E_{11} \dot{\mathbf{v}}(t) = A_{11} \mathbf{v}(t) + A_{12} \mathbf{p}(t) + B_1 \mathbf{u}(t) \\
0 = A_{12}^T \mathbf{v}(t) \quad (\text{NSDAE})
\]
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 1. Problem/no need for divergence free FE

- Incompressible Navier-Stokes-Equations

\[
\begin{align*}
\frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\text{Re}} \Delta \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p &= 0 \quad + \text{B.C.} \\
\nabla \cdot \mathbf{v} &= 0
\end{align*}
\] (NSE)

- Spatial FE discretization

\[
\begin{align*}
M \dot{\mathbf{v}}(t) &= K(\nu)\mathbf{v}(t) - Gp(t) + B_1 \mathbf{u}(t) \\
0 &= G^T \mathbf{v}(t)
\end{align*}
\] (dNSE)

- Linearization and change of notation

\[
\begin{align*}
E_{11} \dot{\mathbf{v}}(t) &= A_{11} \mathbf{v}(t) + A_{12} p(t) + B_1 \mathbf{u}(t) \\
0 &= A_{12}^T \mathbf{v}(t)
\end{align*}
\] (NSDAE)
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 1. Problem/no need for divergence free FE

\[
E_{11} \dot{v}(t) = A_{11} v(t) + A_{12} p(t) + B_1 u(t) \\
0 = A_{12}^T v(t)
\]
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 1. Problem/no need for divergence free FE

\begin{align*}
E_{11} \dot{v}(t) &= A_{11} v(t) + A_{12} p(t) + B_1 u(t) \\
0 &= A_{12}^T v(t)
\end{align*}

Multiplication of line one from the left by $A_{12}^T E_{11}^{-1}$ together with

\begin{align*}
0 &= A_{12}^T v(t) \Rightarrow 0 = A_{12}^T \dot{v}(t)
\end{align*}

reveals the hidden manifold

\begin{align*}
0 &= A_{12}^T E_{11}^{-1} A_{11} v(t) + A_{12}^T E_{11}^{-1} A_{12} p(t) + A_{12}^T E_{11}^{-1} B_1 u(t),
\end{align*}
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 1. Problem/no need for divergence free FE

\[
E_{11} \dot{v}(t) = A_{11} v(t) + A_{12} p(t) + B_1 u(t) \\
0 = A_{12}^T v(t)
\]

Multiplication of line one from the left by \( A_{12}^T E_{11}^{-1} \) together with

\[
0 = A_{12}^T v(t) \Rightarrow 0 = A_{12}^T \dot{v}(t)
\]

reveals the

**hidden manifold**

\[
0 = A_{12}^T E_{11}^{-1} A_{11} v(t) + A_{12}^T E_{11}^{-1} A_{12} p(t) + A_{12}^T E_{11}^{-1} B_1 u(t),
\]

which implies

\[
p(t) = - \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1} A_{11} v(t) - \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1} B_1 u(t).
\]
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 1. Problem/no need for divergence free FE

Inserting $p$ we find

$$E_{11} \dot{v}(t) = \left( I - A_{12} \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1} \right) A_{11} v(t)$$

$$+ \left( I - A_{12} \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1} \right) B_1 u(t)$$

Definition

$\Pi := I - A_{12} \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1}$

[Heinkenschloss/Sorensen/Sun ’08]
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 1. Problem/no need for divergence free FE

Inserting $p$ we find

$$E_{11} \dot{\nu}(t) = \left( I - A_{12} \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1} \right) A_{11} \nu(t)$$

$$+ \left( I - A_{12} \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1} \right) B_1 u(t)$$

Definition

$\Pi := I - A_{12} \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1}$
Solving AREs for Linearized Navier-Stokes Eqns.

Derivation of the Projected State Space System and Matrix Equations

**Definition**

\[
\Pi := I - A_{12} \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1}
\]

**Properties**

- \( \Pi^2 = \Pi \)
- \( \Pi E_{11} = E_{11} \Pi^T \)
- \( \text{null}(\Pi) = \text{range}(A_{12}) \)
- \( \text{range}(\Pi) = \text{null}(A_{12}^T E_{11}^{-1}) \)

This implies

**Lemma 1**

- \( \Pi \) is an oblique projector.
- \( A_{12}^T z = 0 \iff \Pi^T z = z \)
- \( \Rightarrow \Pi^T \nu(t) = \nu(t) \)
Solving AREs for Linearized Navier-Stokes Eqns.
Derivation of the Projected State Space System and Matrix Equations

Definition

\[ \Pi := I - A_{12} \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1} \]

Properties

- \( \Pi^2 = \Pi \)
- \( \Pi E_{11} = E_{11} \Pi^T \)
- \( \text{null}(\Pi) = \text{range}(A_{12}) \)
- \( \text{range}(\Pi) = \text{null}(A_{12}^T E_{11}^{-1}) \)

This implies

Lemma 1

- \( \Pi \) is an oblique projector.
- \( A_{12}^T z = 0 \iff \Pi^T z = z \)
- \( \Rightarrow \Pi^T \nu(t) = \nu(t) \)
Solving AREs for Linearized Navier-Stokes Eqns.

Derivation of the Projected State Space System and Matrix Equations

Thus (NSDAE) is equivalent to

**Projected state space system**

\[
\Pi E_{11} \Pi^T \frac{d}{dt} v(t) = \Pi A_{11} \Pi^T v(t) + \Pi B_1 u(t).
\]

Leads to

**Projected Riccati equation**

\[
\Pi \Pi^T + \Pi A_{11}^T \Pi^T \Pi E_{11} \Pi^T + \Pi E_{11}^T \Pi^T \Pi A_{11} \Pi^T \\
- \Pi E_{11}^T \Pi^T \Pi B_1 B_1^T \Pi^T \Pi E_{11} \Pi^T = 0 \\
\Pi^T \Pi \Pi = \Pi.
\]

If necessary, \( p \) can be determined from

\[
p(t) = - (A_{12}^T E_{11}^{-1} A_{12})^{-1} A_{12}^T E_{11}^{-1} A_{11} v(t) - (A_{12}^T E_{11}^{-1} A_{12})^{-1} A_{12}^T E_{11}^{-1} B_1 u(t).
\]
Solving AREs for Linearized Navier-Stokes Eqns.

Derivation of the Projected State Space System and Matrix Equations

Thus (NSDAE) is equivalent to

**Projected state space system**

\[ \Pi E_{11} \Pi^T \frac{d}{dt} \nu(t) = \Pi A_{11} \Pi^T \nu(t) + \Pi B_1 u(t). \]

Leads to

**Projected Riccati equation**

\[
\Pi \Pi^T + \Pi A_{11}^T \Pi^T \chi \Pi E_{11} \Pi^T + \Pi E_{11}^T \Pi^T \chi \Pi A_{11} \Pi^T \\
- \Pi E_{11}^T \Pi^T \chi \Pi B_1 B_1^T \Pi^T \chi \Pi E_{11} \Pi^T = 0 \\
\Pi^T \chi \Pi = \chi.
\]

If necessary \( p \) can be determined from

\[ p(t) = - \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1} A_{11} \nu(t) - \left( A_{12}^T E_{11}^{-1} A_{12} \right)^{-1} A_{12}^T E_{11}^{-1} B_1 u(t). \]
Solving AREs for Linearized Navier-Stokes Eqns.

Solving the Projected Matrix Equations

Apply factored-Newton-ADI

Central question

How do we solve systems of equations

\[ Z = \Pi^T Z, \quad \Pi (E_{11} + p\ell A_\ell) \Pi^T Z = \Pi \tilde{G} \]

in the (inner) ADI steps avoiding the computation of \( \Pi \)?

For \( A_\ell = A_{11} \), i.e., \( K_\ell = 0 \):

Lemma

\[
\begin{align*}
\Pi (E_{11} + p\ell A_{11}) \Pi^T Z &= \Pi \tilde{G} \\
\iff \\
\begin{bmatrix} E_{11} + p\ell & A_{12} \\ A_{12}^T & 0 \end{bmatrix} \begin{bmatrix} Z \\ \Lambda \end{bmatrix} &= \begin{bmatrix} \tilde{G} \\ 0 \end{bmatrix}
\end{align*}
\]

[Heinkenschloss/Sorensen/Sun '08]
Solving AREs for Linearized Navier-Stokes Eqns.

Solving the Projected Matrix Equations

Apply factored-Newton-ADI

Central question

How do we solve systems of equations

\[ Z = \Pi^T Z, \quad \Pi (E_{11} + p_\ell A_\ell) \Pi^T Z = \Pi \tilde{G} \]

in the (inner) ADI steps avoiding the computation of \( \Pi \)?

For \( A_\ell = A_{11} \), i.e., \( K_\ell = 0 \):

Lemma

\[
\Pi (E_{11} + p_\ell A_{11}) \Pi^T Z = \Pi \tilde{G} \iff \begin{bmatrix} E_{11} + p_\ell A_{11} & A_{12} \\ A_{12}^T & 0 \end{bmatrix} \begin{bmatrix} Z \\ \Lambda \end{bmatrix} = \begin{bmatrix} \tilde{G} \\ 0 \end{bmatrix}
\]

[Heinkenschloss/Sorensen/Sun '08]
Solving AREs for Linearized Navier-Stokes Eqns.

Solving the Projected Matrix Equations

Apply factored-Newton-ADI

Central question

How do we solve systems of equations

\[ A_\ell := A_{11} - BK_\ell \]

\[ Z = \Pi^T \Pi \]

\[ \Pi (E_{11} + p_\ell A_\ell) \Pi^T Z = \Pi \tilde{G} \]

in the (inner) ADI steps avoiding the computation of \( \Pi \)?

- exploit “sparse + low rank” structure of \( A_\ell \),
- precondition our saddle point problem.

For \( A_\ell = A_{11} \), i.e., \( K_\ell = 0 \):

Lemma

\[ \Pi (E_{11} + p_\ell A_{11}) \Pi^T Z = \Pi \tilde{G} \]

\[ \Leftrightarrow \begin{bmatrix} E_{11} + p_\ell A_{11} & A_{12} \\ A_{12}^T & 0 \end{bmatrix} \begin{bmatrix} Z \\ \Lambda \end{bmatrix} = \begin{bmatrix} \tilde{G} \\ 0 \end{bmatrix} \]

[Heinkenschloss/Sorensen/Sun ’08]

(joint work with A. Wathen/M. Stoll)
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 2. Problem: remove $W$ from r.h.s. of Lyapunov eqns. in Newton-ADI

One step of Newton-Kleinman iteration for ARE:

$$A_j^T(X_j + N_j) + X_{j+1}A_j = -W - (X_jB)B^TX_j$$

for $j = 1, 2, \ldots$

$$= X_{j+1}$$

$$= K_j^T = K_j$$

Subtract two consecutive equations $\implies$

$$A_j^TN_j + N_jA_j = N_{j-1}^TBB^TN_{j-1}$$

for $j = 1, 2, \ldots$

See [Banks/Ito '91, B./Hernández/Pastor '03, Morris/Navasca '05] for details and applications of this variant.

But: need $B^TN_0 = K_1 - K_0!$

Assuming $K_0$ is known, need to compute $K_1$. 
Solving AREs for Linearized Navier-Stokes Eqns.

Solution to 2. Problem: remove $W$ from r.h.s. of Lyapunov eqns. in Newton-ADI

Solution idea:

$$K_1 = B^T X_1$$

$$= B^T \int_0^\infty e^{(A-BK_0)T} t (W + K_0^T K_0) e^{(A-BK_0)T} dt$$

$$= \int_0^\infty g(t) dt \approx \sum_{\ell=0}^{N} \gamma_\ell g(t_\ell),$$

where $g(t) = \left( e^{(A-BK_0)t} B \right)^T (W + K_0^T K_0) e^{(A-BK_0)t}.$

[Borggaard/Stoyanov '08]:
evaluate $g(t_\ell)$ using ODE solver applied to $\dot{x} = (A - BK_0)x + \text{adjoint eqn}.$
Better solution idea:
(related to frequency domain POD [Willcox/Peraire ’02])

\[
K_1 = B^T X_1
\]

\[
= B^T \cdot \frac{1}{2\pi} \int_{-\infty}^{\infty} (j\omega I_n - A_0)^{-H} (W + K_0^T K_0) (j\omega I_n - A_0)^{-1} d\omega
\]

\[
= \int_{-\infty}^{\infty} f(\omega) d\omega \approx \sum_{\ell=0}^{N} \gamma_\ell f(\omega_\ell),
\]

where \( f(\omega) = \left( -((j\omega I_n + A_0)^{-1} B)^T (W + K_0^T K_0) (j\omega I_n - A_0)^{-1} \right. \)

Evaluation of \( f(\omega_\ell) \) requires

- 1 sparse LU decomposition (complex!),
- 2m forward/backward solves,
- m sparse and 2m low-rank matrix-vector products.

Use adaptive quadrature with high accuracy, e.g. Gauß-Kronrod (quadgk in MATLAB).
Further Applications

Navier-Stokes Coupled with (Passive) Transport of (Reactive) Species

Goal: stabilize concentration at certain level

Model equations:

\[
\partial_t \mathbf{v} - \frac{1}{Re} \Delta \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p = f
\]

\[
\text{div} \mathbf{v} = 0
\]

\[
\partial_t c + \mathbf{v} \cdot \nabla c - \frac{1}{Re \cdot Sc} \Delta c = 0
\]

with boundary conditions:

\[
\mathbf{v} = \mathbf{v}_0 \quad c = c_0 = \text{const} \quad \text{on } \Gamma_{in}
\]

\[
\mathbf{v} = 0 \quad \partial_t c = 0 \quad \text{on } \Gamma_{wall}
\]

\[
\mathbf{v} = 0 \quad c = 0 \quad \text{on } \Gamma_r,
\]
Further Applications

Navier-Stokes Coupled with (Passive) Transport of (Reactive) Species

Goal: stabilize concentration at certain level

Model equations:

\[ \partial_t \mathbf{v} - \frac{1}{Re} \Delta \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p = f \]
\[ \text{div} \mathbf{v} = 0 \]
\[ \partial_t c + \mathbf{v} \cdot \nabla c - \frac{1}{Re \cdot Sc} \Delta c = 0 \]

Domain:
Further Applications

Results for $Re = 10$, $Sc = 10$

no control

movie

piecewise constant feedback

Computations by Heiko Weichelt
Further Applications

Results for $Re = 10, Sc = 10$
Conclusions and Future Work

- Progress in solving AREs in the last decade now allows application of Riccati feedback to realistic PDE control problems.
- Implementation for Navier-Stokes and multi-field flow problems in progress, requires many details not encountered for linear convection-diffusion or beam equations.
- For 3D problems, need dedicated preconditioned iterative "saddle point" solver.
  
  "(1,1)"-term is nonsymmetric sparse matrix + low-rank perturbation $\rightsquigarrow$ joint work with A. Wathen, M. Stoll.
- Model reduction based on LQG balanced truncation for flow problems in $L_2(0, \infty; V_n(\Omega))$ can be based on derived Riccati solver.
References


