Control and Optimization
with Differential-Algebraic Constraints
Banff, October 24–29, 2010

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

Friedrich-Alexander-Universität Erlangen-Nürnberg

supported by DFG SPP 1253

MAX PLANCK INSTITUTE FOR DYNAMICS OF COMPLEX TECHNICAL SYSTEMS MAGDEBURG

Control and Optimization
with Differential-Algebraic Constraints
Banff, October 24–29, 2010

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

Friedrich-Alexander-Universität Erlangen-Nürnberg

supported by DFG SPP 1253 Optimization with partial differential equations

MAX PLANCK INSTITUTE FOR DYNAMICS OF COMPLEX TECHNICAL SYSTEMS MAGDEBURG

Overview

Introduction

Optimal control-based stabilization for Navier-Stokes equations

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

$$\partial_t v + v \cdot \nabla v - \frac{1}{Re}\Delta v + \nabla \chi = f \tag{1a}$$

$$\operatorname{div} v = 0 \tag{1b}$$

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

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

- 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 χ), described by incompressible Navier-Stokes equations

$$\partial_t v + v \cdot \nabla v - \frac{1}{Re}\Delta v + \nabla \chi = f \tag{1a}$$

$$\operatorname{div} v = 0 \tag{1b}$$

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

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

- 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).

Optimal control-based stabilization for NSEs

Analytical solution [Raymond''05–'07]

Assume w solves the stationary Navier-Stokes equations

$$w \cdot \nabla w - \frac{1}{Re}\Delta w + \nabla \chi_s = f, \qquad \operatorname{div} w = 0, \tag{2}$$

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

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

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

is stable for "small" initial values $z_0 \in X(\Omega) \subset V_n^0(\Omega)$, where

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

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

$\Longrightarrow \left\{ \begin{array}{l} \text{Solution to instationary Navier-Stokes equations with } v = w + z, \\ \chi = \chi_s + p, \text{ and } v(0) = w + z_0 \text{ in } \Omega \text{ is controlled to } w. \end{array}\right.$

Optimal control-based stabilization for NSEs

Analytical solution [Raymond''05–'07]

Assume w solves the stationary Navier-Stokes equations

$$w \cdot \nabla w - \frac{1}{Re}\Delta w + \nabla \chi_s = f, \qquad \operatorname{div} w = 0, \tag{2}$$

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

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

is stable for "small" initial values $z_0 \in X(\Omega) \subset V_n^0(\Omega)$, where

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

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

$\Longrightarrow \left\{ \begin{array}{l} \text{Solution to instationary Navier-Stokes equations with } v = w + z, \\ \chi = \chi_s + p, \text{ and } v(0) = w + z_0 \text{ in } \Omega \text{ is controlled to } w. \end{array} \right.$

Optimal control-based stabilization for NSEs

Analytical solution [Raymond''05–'07]

Assume w solves the stationary Navier-Stokes equations

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

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

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

is stable for "small" initial values $z_0 \in X(\Omega) \subset V_n^0(\Omega)$, where

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

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

$\Longrightarrow$ Solution to instationary Navier-Stokes equations with $v = w + z$, $\chi = \chi_s + p$, and $v(0) = w + z_0$ in Ω is controlled to w.

Optimal control-based stabilization for NSEs

Analytical solution [RAYMOND '05–'07]

Linearized Navier-Stokes control system:

$$\partial_t z + (z \cdot \nabla) w + (w \cdot \nabla) z - \frac{1}{Re} \Delta z - \omega z + \nabla p = 0 \text{ in } Q_\infty \qquad (3a)$$

$$\operatorname{div} z = 0 \text{ in } Q_\infty \qquad (3b)$$

$$z = bu \text{ in } \Sigma_\infty \qquad (3c)$$

$$z(0) = z_0 \text{ in } \Omega, \qquad (3d)$$

ωz with $\omega > 0$ de-stabilizes the system further, needed to guarantee exponential stabilization, ω controls decay rate!

Cost functional (with P = Helmholtz projector)

$$J(z, u) = \frac{1}{2} \int_0^\infty \langle Pz, Pz \rangle_{L_2(\Omega)} + \rho u(t)^2 \, dt, \qquad (4)$$

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

$$\inf \{ J(z, u) \mid (z, u) \text{ satisfies (3)}, \; u \in L_2(0, \infty) \} . \qquad (5)$$

Optimal control-based stabilization for NSEs

Analytical solution [RAYMOND '05–'07]

Linearized Navier-Stokes control system:

$$\partial_t z + (z \cdot \nabla)w + (w \cdot \nabla)z - \frac{1}{Re}\Delta z - \omega z + \nabla p = 0 \ \text{ in } Q_\infty \qquad \text{(3a)}$$

$$\operatorname{div} z = 0 \ \text{ in } Q_\infty \qquad \text{(3b)}$$

$$z = bu \text{ in } \Sigma_\infty \qquad \text{(3c)}$$

$$z(0) = z_0 \ \text{ in } \Omega, \qquad \text{(3d)}$$

ωz with $\omega > 0$ de-stabilizes the system further, needed to guarantee exponential stabilization, ω controls decay rate!

Cost functional (with P = Helmholtz projector)

$$J(z, u) = \frac{1}{2}\int_0^\infty \langle Pz, Pz\rangle_{L_2(\Omega)} + \rho u(t)^2 \, dt, \qquad (4)$$

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

$$\inf \{J(z, u) \mid (z, u) \text{ satisfies (3)}, \ u \in L_2(0, \infty)\}. \qquad (5)$$

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 ($\rightsquigarrow \operatorname{div} z_H \equiv 0$);
- $\Pi = \Pi^* \in \mathcal{L}(V_n^0(\Omega))$ 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_τ 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 $P : L_2(\Omega) \mapsto V_n^0(\Omega)$ being the Helmholtz projector ($\rightsquigarrow \operatorname{div} z_H \equiv 0$);
- $\Pi = \Pi^* \in \mathcal{L}(V_n^0(\Omega))$ 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_τ 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 $P : L_2(\Omega) \mapsto V_n^0(\Omega)$ being the Helmholtz projector ($\rightsquigarrow \operatorname{div} z_H \equiv 0$);
- $\Pi = \Pi^* \in \mathcal{L}(V_n^0(\Omega))$ 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_τ 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 $P : L_2(\Omega) \mapsto V_n^0(\Omega)$ being the Helmholtz projector ($\rightsquigarrow \operatorname{div} z_H \equiv 0$);
- $\Pi = \Pi^* \in \mathcal{L}(V_n^0(\Omega))$ 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_τ 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.

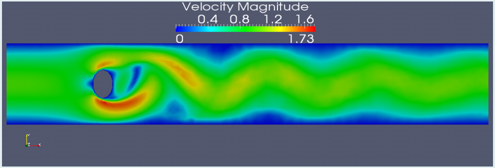
Introduction

Proof of concept: von Kármán vortex street

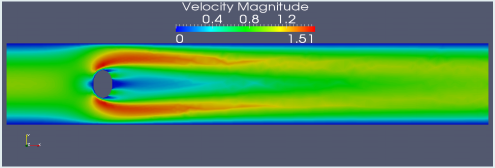
von Kármán vortex street ($Re = 300$)

Vortex suppression by blowing in at upper end of cylinder,

without ...



and with stabilizing feedback



Computations by Heiko Weichelt

Solving Large-Scale AREs

Low-Rank Newton-ADI for AREs

Consider

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

Re-write Newton's method for AREs ($A_j := A - BB^T X_j$)

$$D\mathcal{R}(X_j)\,(N_j) = -\mathcal{R}(X_j)$$

$$\Longleftrightarrow$$

$$A_j^T \underbrace{(X_j + N_j)}_{=X_{j+1}} + \underbrace{(X_j + N_j)}_{=X_{j+1}} A_j = \underbrace{-C^T C - X_j BB^T X_j}_{=:-W_j W_j^T}$$

Set $X_j = Z_j Z_j^T$ for $\operatorname{rank}(Z_j) \ll n \Longrightarrow$

$$A_j^T \left(Z_{j+1} Z_{j+1}^T\right) + \left(Z_{j+1} Z_{j+1}^T\right) A_j = -W_j W_j^T$$

Factored Newton Iteration [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.

Solving Large-Scale AREs

Low-Rank Newton-ADI for AREs

Consider

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

Re-write Newton's method for AREs ($A_j := A - BB^T X_j$)

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

$$\Longleftrightarrow$$

$$A_j^T \underbrace{(X_j + N_j)}_{=X_{j+1}} + \underbrace{(X_j + N_j)}_{=X_{j+1}} A_j = \underbrace{-C^T C - X_j BB^T X_j}_{=:-W_j W_j^T}$$

Set $X_j = Z_j Z_j^T$ for $\operatorname{rank}(Z_j) \ll n \Longrightarrow$

$$A_j^T \left(Z_{j+1} Z_{j+1}^T\right) + \left(Z_{j+1} Z_{j+1}^T\right) A_j = -W_j W_j^T$$

Factored Newton Iteration [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.

Solving Large-Scale AREs

Low-Rank Newton-ADI for AREs

Consider

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

Re-write Newton's method for AREs ($A_j := A - BB^T X_j$)

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

$$\Longleftrightarrow$$

$$A_j^T \underbrace{(X_j + N_j)}_{=X_{j+1}} + \underbrace{(X_j + N_j)}_{=X_{j+1}} A_j = \underbrace{-C^T C - X_j BB^T X_j}_{=:-W_j W_j^T}$$

Set $X_j = Z_j Z_j^T$ for $\operatorname{rank}(Z_j) \ll n \Longrightarrow$

$$A_j^T \left(Z_{j+1} Z_{j+1}^T\right) + \left(Z_{j+1} Z_{j+1}^T\right) A_j = -W_j W_j^T$$

Factored Newton Iteration [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.

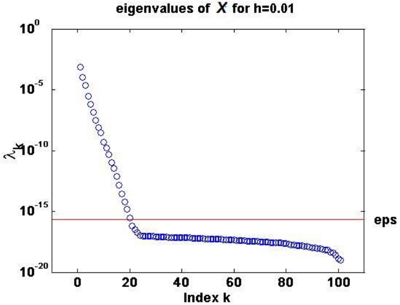
Low-Rank Newton-ADI for AREs

Low-Rank Approximation

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

Example:

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



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

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

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

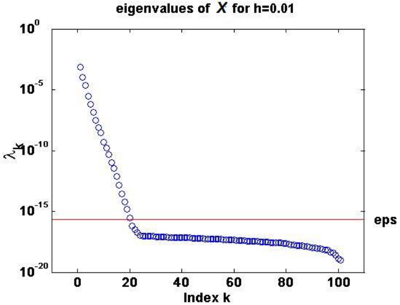
Low-Rank Newton-ADI for AREs

Low-Rank Approximation

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

Example:

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



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

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

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

Review: LRCF-ADI for Lyapunov Equations

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

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

For $j = 1, \ldots, J$

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

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

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

$\ldots \rightsquigarrow$ low-rank Cholesky factor ADI

[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) [Wachspress '95]

For $j = 1, \ldots, J$

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

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

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

$\ldots \rightsquigarrow$ low-rank Cholesky factor ADI

[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) [Wachspress '95]

For $j = 1, \ldots, J$

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

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

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

$\ldots \rightsquigarrow$ low-rank Cholesky factor ADI

[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)

[Penzl '97/'00, Li/White '99/'02, B./Li/Penzl '99/'08]

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

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

Krylov Subspace Based Solvers for Lyapunov Equations

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

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

Krylov projection idea [Saad '90, Jaimoukha/Kasenally '94]

Solve

$$(U_m^T F U_m) Y_m + Y_m (U_m^T F^T U_m) = -U_m^T G G^T U_m, \tag{6}$$

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

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

Krylov projection idea [Saad '90, Jaimoukha/Kasenally '94]

Solve

$$(U_m^T F U_m) Y_m + Y_m (U_m^T F^T U_m) = -U_m^T G G^T U_m, \tag{6}$$

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

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

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

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

repeat
 if not first step **then**
  increase dimension of $\mathcal{V}$ and update V.
 end if
 Solve the "small" LE for $\tilde{Z}$ with a classical solver:

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

 Lift $\tilde{Z}$ to the full space: $Z = V\tilde{Z}$
until res(Z)$<$ TOL

[1](K-PIK, [Simoncini '07])

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

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

Idea

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

LRCF-ADI with Galerkin Projection

ADI and Rational Krylov

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

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

Idea

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

LRCF-ADI with Galerkin Projection

Projected ADI Step $\longrightarrow$ LRCF-ADI-GP

[B./Li/Truhar'09, Saak'09, B./Saak'10]

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

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

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

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

LRCF-ADI with Galerkin Projection

Projected ADI Step $\longrightarrow$ LRCF-ADI-GP

[B./Li/Truhar'09, Saak'09, B./Saak'10]

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

LRCF-ADI with Galerkin Projection

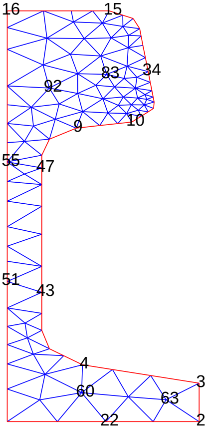
Test Example: Optimal Cooling of Steel Profiles

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

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

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

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

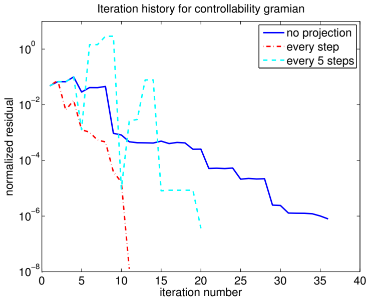


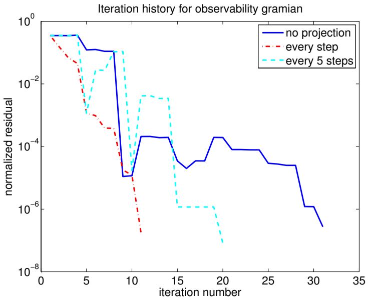
Source: Physical model: courtesy of Mannesmann/Demag.
Math. model: Tröltzsch/Unger '99/'01, Penzl '99, S. '03.

LRCF-ADI with Galerkin Projection

Numerical Results

Steel profile n=20 209 good shifts

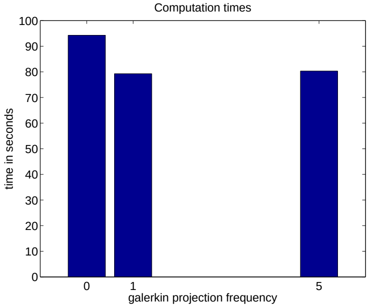




LRCF-ADI with Galerkin Projection

Numerical Results

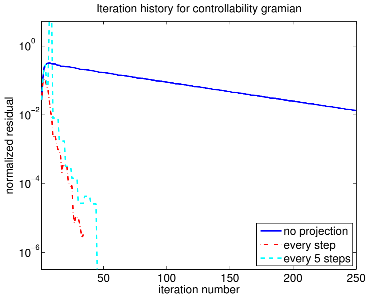
Steel profile n=20 209 good shifts

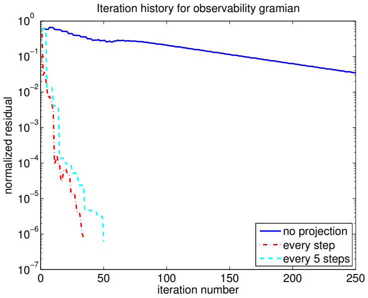


LRCF-ADI with Galerkin Projection

Numerical Results

Steel profile n=20 209 bad shifts

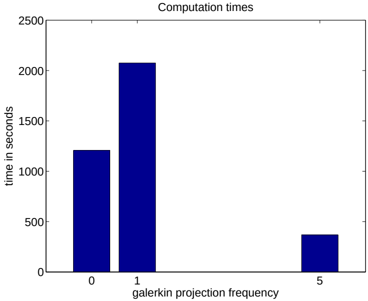




LRCF-ADI with Galerkin Projection

Numerical Results

Steel profile n=20 209 bad shifts



Solving Large-Scale AREs

LRCF-NM for the ARE

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

Newton's Iteration for the ARE

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

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

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

i.e., in every Newton step solve a

Lyapunov Equation [Kleinman '68]

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

where $F_\ell := A - BB^T X_\ell$, $G := [\,-C^T,\ -X_\ell B\,]$.

Solving Large-Scale AREs

LRCF-NM for the ARE

Factored Newton-Kleinman Iteration [B./Li/Penzl '99/'08]

$$F_\ell = A - BB^T X_\ell =: A - BK_\ell \qquad \text{is "sparse + low rank",}$$
$$G_\ell = [\, C^T,\ K_\ell^T \,] \qquad \text{is low rank factor.}$$

- Apply LRCF-ADI in every Newton step;
- exploit structure of F_ℓ using Sherman-Morrison-Woodbury formula:

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

Solving Large-Scale AREs

LRCF-NM for the ARE

Factored Newton-Kleinman Iteration [B./Li/Penzl '99/'08]

$$F_\ell = A - BB^T X_\ell =: A - BK_\ell \quad \text{is "sparse + low rank"},$$
$$G_\ell = [\, C^T,\ K_\ell^T \,] \quad \text{is low rank factor.}$$

- Apply LRCF-ADI in every Newton step;
- exploit structure of F_ℓ using Sherman-Morrison-Woodbury formula:

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

Solving Large-Scale AREs

LRCF-NM for the ARE

Factored Newton-Kleinman Iteration [B./Li/Penzl '99/'08]

$$F_\ell = A - BB^T X_\ell =: A - BK_\ell \quad \text{is "sparse + low rank"},$$
$$G_\ell = [\, C^T, \ K_\ell^T \,] \quad \text{is low rank factor.}$$

- Apply LRCF-ADI in every Newton step;
- exploit structure of F_ℓ using Sherman-Morrison-Woodbury formula:

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

LRCF-NM for the ARE

Algorithms

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

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

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

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

LRCF-NM for the ARE

Algorithms

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

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

1: Determine (sub)optimal ADI shift parameters $p_1, p_2, \dots$ with respect to $F^{(0)} = A^T - K^{(0)} B^T$ or $F^{(\infty)} = \lim_{k\to\infty} F^{(k)}$.

2: **for** $k = 1, 2, \dots, k_{max}$ **do**

3: $\quad G^{(k)} = \left[\begin{array}{cc} C^T & K^{(k-1)} \end{array}\right]$

4: $\quad$ Compute $Z^{(k)}$ using Algorithm 1 (LRCF-ADI) or (LRCF-ADI-GP) such that $\quad F^{(k)} Z^{(k)} Z^{(k)^H} + Z^{(k)} Z^{(k)^H} F^{(k)^T} \approx -G^{(k)} G^{(k)^T}$.

5: $\quad K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$

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

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

1: **for** $k = 1, 2, \ldots, k_{max}$ **do**
2: Determine (sub)optimal ADI shift parameters $p_1^{(k)}, p_2^{(k)}, \ldots$ with respect to the matrix $F^{(k)} = A^T - K^{(k-1)} B^T$.
3: $G^{(k)} = \begin{bmatrix} C^T & K^{(k-1)} \end{bmatrix}$
4: Compute $Z^{(k)}$ using Algorithm 1 (LRCF-ADI) or (LRCF-ADI-GP) such that $F^{(k)} Z^{(k)} Z^{(k)^H} + Z^{(k)} Z^{(k)^H} F^{(k)^T} \approx -G^{(k)} G^{(k)^T}$.
5: Project ARE, solve and prolongate solution.
6: $K^{(k)} = Z^{(k)} (Z^{(k)^H} B)$
7: **end for**

Solving for the Feedback Operator

Feedback Iteration

$Z_{j,k} = [Z_{j,k-1}, V_{j,k}]$,

j: Newton index,

k: ADI index.

Optimal feedback

$$K_* = B^T X_* = B^T Z_* Z_*^T$$

can be computed by direct feedback iteration:

- jth Newton iteration:

$$K_j = B^T Z_j Z_j^T = \sum_{k=1}^{k_{\max}} (B^T V_{j,k}) V_{j,k}^T \xrightarrow{j\rightarrow\infty} K_* = B^T Z_* Z_*^T$$

- 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].

Feedback Iteration

Test Examples

Example 1: 3d Convection-Diffusion Equation

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

Example 2: 2d Convection-Diffusion Equation

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

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

Feedback Iteration

Test Results (ADI-loop): Example 1

Newton-ADI

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

CPU time: 4 805.8 sec.

Newton-Galerkin-ADI LRCF-ADI-GP(5)

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

CPU time: 1 460.1 sec.

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

Feedback Iteration

Test Results (ADI-loop): Example 2

Newton-ADI

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

CPU time: 493.81 sec.

Newton-Galerkin-ADI LRCF-ADI-GP(5)

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

CPU time: 280.55 sec.

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

Feedback Iteration

Test Results (both-loops): Example 1

Newton-ADI

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

CPU time: 4 805.8 sec.

NG-ADI inner= 5, outer= 1

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

CPU time: 497.6 sec.

NG-ADI inner= 1, outer= 1

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

CPU time: 856.6 sec.

NG-ADI inner= 0, outer= 1

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

CPU time: 506.6 sec.

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

Feedback Iteration

Test Results (both-loops): Example 2

Newton-ADI

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

CPU time: 493.81 sec.

NG-ADI inner= 5, outer= 1

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

CPU time: 24.1 sec.

NG-ADI inner= 1, outer= 1

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

CPU time: 26.8 sec.

NG-ADI inner= 0, outer= 1

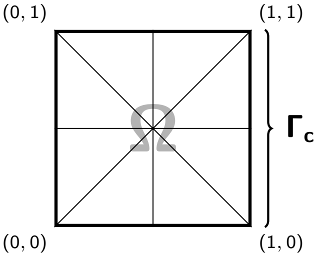
NWT	rel. change	rel. residual	ADI
1	1	$3.27 \cdot 10^{-15}$	46

CPU time: 24.0 sec.

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

Feedback Iteration

Computation Time Scales Linearly with Problem Size



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

Control operator: Here $b(\xi) = 4\,(1-\xi_2)\,\xi_2$ for $\xi \in \Gamma_c$ and 0 otherwise.
Output equation: $y = Cx$, where

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

Cost functional:

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

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

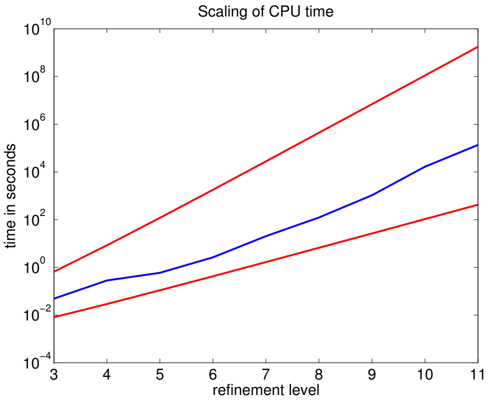
discretization level	problem size	time in seconds
3	81	$5.53 \cdot 10^{-2}$
4	289	$1.33 \cdot 10^{-1}$
5	1 089	$2.84 \cdot 10^{-1}$
6	4 225	$1.51 \cdot 10^{+0}$
7	16 641	$9.52 \cdot 10^{+0}$
8	66 049	$5.97 \cdot 10^{+1}$
9	263 169	$4.72 \cdot 10^{+2}$
10	1 050 625	$6.89 \cdot 10^{+3}$
11	4 198 401	$8.08 \cdot 10^{+4}$

(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

   $$A_j^T Z_{j+1} Z_{j+1}^T M + M Z_{j+1} Z_{j+1}^T A_j = -M - K_j^T K_j$$

   $n_v := \mathrm{rank}\,(M) =$ 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.

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

Problems with Newton-Kleinman

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

   $$\boxed{0 = \mathbf{I} + (\mathbf{A} + \omega \mathbf{I})^* \mathbf{X} + \mathbf{X}(\mathbf{A} + \omega \mathbf{I}) - \mathbf{X}(\mathbf{B}_\tau \mathbf{B}_\tau^* + \rho^{-1} \mathbf{B}_n \mathbf{B}_n^*)\mathbf{X}}$$

   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

   $$A_j^T Z_{j+1} Z_{j+1}^T M + M Z_{j+1} Z_{j+1}^T A_j = -M - K_j^T K_j$$

   $n_v := \operatorname{rank}(M) =$ 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.

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

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

$$A_j^T Z_{j+1} Z_{j+1}^T M + M Z_{j+1} Z_{j+1}^T A_j = -M - K_j^T K_j$$

   $n_v := \operatorname{rank}(M) =$ 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.

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

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

   n_v

   $\rightsquigarrow$

   Newton-ADI iteration!

   [B. '08]*Partial Stabilization of Descriptor Systems Using Spectral Projectors*; to appear in V. Olshevsky et al (eds.), Numerical Linear Algebra in Signals, Systems and Control, Lecture Notes in Electrical Engineering, Springer-Verlag.

   [HEIN '10]*MPC/LQG-Based Optimal Control of Nonlinear Parabolic PDEs*; PhD thesis Chemnitz UT.

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.

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

- incompressible Navier-Stokes-Equations

$$\frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\mathrm{Re}}\Delta \mathbf{v} + \mathbf{v}.\nabla \mathbf{v} + \nabla \mathbf{p} = 0 \quad + \text{B.C.}$$
$$\nabla.\mathbf{v} = 0 \qquad \text{(NSE)}$$

- Spatial FE discretization

$$M\dot{v}(t) = K(v)v(t) - Gp(t) + B_1\mathbf{u}(t)$$
$$0 = G^T v(t) \qquad \text{(dNSE)}$$

- Linearization and change of notation

$$E_{11}\dot{v}(t) = A_{11}v(t) + A_{12}p(t) + B_1\mathbf{u}(t)$$
$$0 = A_{12}^T v(t) \qquad \text{(NSDAE)}$$

Solving AREs for Linearized Navier-Stokes Eqns.

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

- incompressible Navier-Stokes-Equations

$$\frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\mathrm{Re}}\Delta \mathbf{v} + \mathbf{v}.\nabla \mathbf{v} + \nabla \mathbf{p} = 0 \quad + \mathrm{B.C.}$$
$$\nabla.\mathbf{v} = 0 \tag{NSE}$$

- Spatial FE discretization

$$\begin{aligned} M\dot{v}(t) &= K(v)v(t) - Gp(t) + B_1\mathbf{u}(t) \\ 0 &= G^T v(t) \end{aligned} \tag{dNSE}$$

- Linearization and change of notation

$$\begin{aligned} E_{11}\dot{v}(t) &= A_{11}v(t) + A_{12}p(t) + B_1\mathbf{u}(t) \\ 0 &= A_{12}^T v(t) \end{aligned} \tag{NSDAE}$$

Solving AREs for Linearized Navier-Stokes Eqns.

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

- incompressible Navier-Stokes-Equations

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\mathrm{Re}}\Delta \mathbf{v} + \mathbf{v}.\nabla \mathbf{v} + \nabla \mathbf{p} &= 0 \quad + \text{B.C.} \\ \nabla . \mathbf{v} &= 0 \end{aligned} \tag{NSE}$$

- Spatial FE discretization

$$\begin{aligned} M\dot{v}(t) &= K(v)v(t) - Gp(t) + B_1\mathbf{u}(t) \\ 0 &= G^T v(t) \end{aligned} \tag{dNSE}$$

- Linearization and change of notation

$$\begin{aligned} E_{11}\dot{v}(t) &= A_{11}v(t) + A_{12}p(t) + B_1\mathbf{u}(t) \\ 0 &= A_{12}^T v(t) \end{aligned} \tag{NSDAE}$$

Solving AREs for Linearized Navier-Stokes Eqns.

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

- incompressible Navier-Stokes-Equations

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\mathrm{Re}}\Delta \mathbf{v} + \mathbf{v}.\nabla \mathbf{v} + \nabla \mathbf{p} &= 0 \qquad + \text{B.C.} \\ \nabla . \mathbf{v} &= 0 \end{aligned} \tag{NSE}$$

- Spatial FE discretization

$$\begin{aligned} M\dot{v}(t) &= K(v)v(t) - Gp(t) + B_1\mathbf{u}(t) \\ 0 &= G^T v(t) \end{aligned} \tag{dNSE}$$

- Linearization and change of notation

$$\begin{aligned} E_{11}\dot{v}(t) &= A_{11}v(t) + A_{12}p(t) + B_1\mathbf{u}(t) \\ 0 &= A_{12}^T v(t) \end{aligned} \tag{NSDAE}$$

Solving AREs for Linearized Navier-Stokes Eqns.

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

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

Solving AREs for Linearized Navier-Stokes Eqns.

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

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

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

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

Solving AREs for Linearized Navier-Stokes Eqns.

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

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

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

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

which implies

$$p(t) = -\left(A_{12}^T E_{11}^{-1} A_{12}\right)^{-1} A_{12}^T E_{11}^{-1} A_{11} v(t) - \left(A_{12}^T E_{11}^{-1} A_{12}\right)^{-1} A_{12}^T E_{11}^{-1} B_1 \mathbf{u}(t).$$

Solving AREs for Linearized Navier-Stokes Eqns.

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

Inserting p we find

$$\begin{aligned} E_{11}\dot{v}(t) &= \left(I - A_{12}\left(A_{12}^T E_{11}^{-1} A_{12}\right)^{-1} A_{12}^T E_{11}^{-1}\right) A_{11} v(t) \\ &\quad + \left(I - A_{12}\left(A_{12}^T E_{11}^{-1} A_{12}\right)^{-1} A_{12}^T E_{11}^{-1}\right) B_1 \mathbf{u}(t) \end{aligned}$$

Definition [Heinkenschloss/Sorensen/Sun '08]

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

Solving AREs for Linearized Navier-Stokes Eqns.

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

Inserting p we find

$$\begin{aligned} E_{11}\dot{v}(t) &= \left(I - A_{12}\left(A_{12}^T E_{11}^{-1} A_{12}\right)^{-1} A_{12}^T E_{11}^{-1}\right) A_{11} v(t) \\ &\quad + \left(I - A_{12}\left(A_{12}^T E_{11}^{-1} A_{12}\right)^{-1} A_{12}^T E_{11}^{-1}\right) B_1 \mathbf{u}(t) \end{aligned}$$

Definition [HEINKENSCHLOSS/SORENSEN/SUN '08]

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

Solving AREs for Linearized Navier-Stokes Eqns.

Derivation of the Projected State Space System and Matrix Equations

Definition [Heinkenschloss/Sorensen/Sun '08]

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

Properties

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

This implies

Lemma 1 [Heinkenschloss/Sorensen/Sun '08]

- Π is an oblique projector.
- $A_{12}^T z = 0 \Leftrightarrow \Pi^T z = z$
- $\Rightarrow \Pi^T v(t) = v(t)$

Solving AREs for Linearized Navier-Stokes Eqns.

Derivation of the Projected State Space System and Matrix Equations

Definition [Heinkenschloss/Sorensen/Sun '08]

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

Properties

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

This implies

Lemma 1 [Heinkenschloss/Sorensen/Sun '08]

- Π is an oblique projector.
- $A_{12}^T z = 0 \Leftrightarrow \Pi^T z = z$
- $\Rightarrow \Pi^T v(t) = v(t)$

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

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

Leads to

projected Riccati equation

$$\begin{aligned} \Pi\Pi^T + \Pi A_{11}^T \Pi^T X \Pi E_{11} \Pi^T + \Pi E_{11}^T \Pi^T X \Pi A_{11} \Pi^T & \\ -\Pi E_{11}^T \Pi^T X \Pi B_1 B_1^T \Pi^T X \Pi E_{11} \Pi^T &= 0 \\ \Pi^T X \Pi &= X. \end{aligned}$$

If necessary p can be determined from

$$p(t) = -\left(A_{12}^T E_{11}^{-1} A_{12}\right)^{-1} A_{12}^T E_{11}^{-1} A_{11} v(t) - \left(A_{12}^T E_{11}^{-1} A_{12}\right)^{-1} A_{12}^T E_{11}^{-1} B_1 \mathbf{u}(t).$$

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

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

Leads to

projected Riccati equation

$$\begin{aligned}\Pi\Pi^T + \Pi A_{11}^T\Pi^T X\Pi E_{11}\Pi^T + \Pi E_{11}^T\Pi^T X\Pi A_{11}\Pi^T&\\ -\Pi E_{11}^T\Pi^T X\Pi B_1 B_1^T\Pi^T X\Pi E_{11}\Pi^T &= 0\\ \Pi^T X\Pi &= X.\end{aligned}$$

If necessary p can be determined from

$$p(t) = -\left(A_{12}^T E_{11}^{-1} A_{12}\right)^{-1} A_{12}^T E_{11}^{-1} A_{11} v(t) - \left(A_{12}^T E_{11}^{-1} A_{12}\right)^{-1} A_{12}^T E_{11}^{-1} B_1\mathbf{u}(t).$$

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 $(A_\ell := A_{11} - BK_\ell)$

$$Z = \Pi^T Z, \qquad \Pi \left(E_{11} + p_\ell A_\ell\right) \Pi^T Z = \Pi \tilde{G}$$

in the (inner) ADI steps avoiding the computation of Π?

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

Lemma [Heinkenschloss/Sorensen/Sun '08]

$$\begin{aligned} Z &= \Pi^T Z \\ \Pi \left(E_{11} + p_\ell A_{11}\right) \Pi^T Z &= \Pi \tilde{G} \end{aligned} \quad \Leftrightarrow \quad \begin{bmatrix} E_{11} + p_\ell & A_{12} \\ A_{12}^T & 0 \end{bmatrix} \begin{bmatrix} Z \\ \Lambda \end{bmatrix} = \begin{bmatrix} \tilde{G} \\ 0 \end{bmatrix}$$

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 $\quad (A_\ell := A_{11} - BK_\ell)$

$$Z = \Pi^T Z, \qquad \Pi \left(E_{11} + p_\ell A_\ell \right) \Pi^T Z = \Pi \tilde{G}$$

in the (inner) ADI steps avoiding the computation of Π?

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

Lemma [Heinkenschloss/Sorensen/Sun '08]

$$\begin{aligned} Z &= \Pi^T Z \\ \Pi \left(E_{11} + p_\ell A_{11} \right) \Pi^T Z &= \Pi \tilde{G} \end{aligned} \quad \Leftrightarrow \begin{bmatrix} E_{11} + p_\ell A_{11} & A_{12} \\ A_{12}^T & 0 \end{bmatrix} \begin{bmatrix} Z \\ \Lambda \end{bmatrix} = \begin{bmatrix} \tilde{G} \\ 0 \end{bmatrix}$$

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 $(A_\ell := A_{11} - BK_\ell)$

$$Z = \Pi^T Z, \qquad \Pi\left(E_{11} + p_\ell A_\ell\right)\Pi^T Z = \Pi \tilde{G}$$

in the (inner) ADI steps avoiding the computation of Π?

For $A_\ell = A_{11}$, i.e., $K_\ell = 0$

tasks

- exploit "sparse + low rank" structure of A_ℓ,
- precondition our saddle point problem.
(joint work with A. Wathen/M. Stoll)

Lemma [Heinkenschloss/Sorensen/Sun '08]

$$\begin{aligned} Z &= \Pi^T Z \\ \Pi\left(E_{11} + p_\ell A_{11}\right)\Pi^T Z &= \Pi \tilde{G} \end{aligned} \quad \Leftrightarrow \quad \begin{bmatrix} E_{11} + p_\ell A_\ell & A_{12} \\ A_{12}^T & 0 \end{bmatrix} \begin{bmatrix} Z \\ \Lambda \end{bmatrix} = \begin{bmatrix} \tilde{G} \\ 0 \end{bmatrix}$$

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:

$$A_j^T \underbrace{(X_j + N_j)}_{=X_{j+1}} + X_{j+1}A_j = -W - \underbrace{(X_j B)}_{=K_j^T} \underbrace{B^T X_j}_{=K_j} \qquad \text{for } j = 1, 2, \ldots$$

Subtract two consecutive equations $\Longrightarrow$

$$A_j^T N_j + N_j A_j = N_{j-1}^T B B^T N_{j-1} \qquad \text{for } j = 1, 2, \ldots$$

See [BANKS/ITO '91, B./HERNÁNDEZ/PASTOR '03, MORRIS/NAVASCA '05] for details and applications of this variant.

But: need $B^T N_0 = K_1 - K_0$!

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:

$$\begin{aligned} K_1 &= B^T X_1 \\ &= B^T \int_0^\infty e^{(A-BK_0)^T t}(W + K_0^T K_0)e^{(A-BK_0)t}\,dt \\ &= \int_0^\infty g(t)\,dt \approx \sum_{\ell=0}^{N} \gamma_\ell g(t_\ell), \end{aligned}$$

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

[Borgggaard/Stoyanov '08]:
evaluate $g(t_\ell)$ using ODE solver applied to $\dot{x} = (A - BK_0)x$ + adjoint eqn.

Solving AREs for Linearized Navier-Stokes Eqns.

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

Better solution idea:

(related to frequency domain POD [Willcox/Peraire '02])

$$\begin{aligned} K_1 &= B^T X_1 \qquad\qquad (\text{Notation: } A_0 := A - BK_0) \\ &= B^T \cdot \frac{1}{2\pi} \int_{-\infty}^{\infty} (\jmath\omega I_n - A_0)^{-H} (W + K_0^T K_0)(\jmath\omega I_n - A_0)^{-1}\, d\omega \\ &= \int_{-\infty}^{\infty} f(\omega)\, d\omega \approx \sum_{\ell=0}^{N} \gamma_\ell f(\omega_\ell), \end{aligned}$$

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

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

- 1 sparse LU decmposition (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:

$$\begin{aligned} \partial_t \mathbf{v} - \frac{1}{Re}\Delta \mathbf{v} + \mathbf{v}.\nabla \mathbf{v} + \nabla \mathbf{p} &= \mathbf{f} \\ \operatorname{div} \mathbf{v} &= 0 \\ \partial_t \mathbf{c} + \mathbf{v}.\nabla \mathbf{c} - \frac{1}{Re \cdot Sc}\Delta \mathbf{c} &= 0 \end{aligned}$$

with boundary conditions:

$$\begin{aligned} \mathbf{v} &= \mathbf{v}_0 & \mathbf{c} = \mathbf{c}_0 &= const & &\text{on } \Gamma_{in} \\ \mathbf{v} &= 0 & \partial_\nu \mathbf{c} &= 0 & &\text{on } \Gamma_{wall} \\ \mathbf{v} &= 0 & \mathbf{c} &= 0 & &\text{on } \Gamma_r, \end{aligned}$$

Further Applications

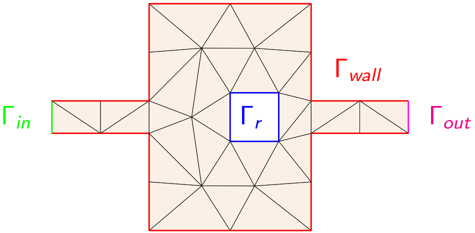
Navier-Stokes Coupled with (Passive) Transport of (Reactive) Species

Goal: stabilize concentration at certain level

Model equations:

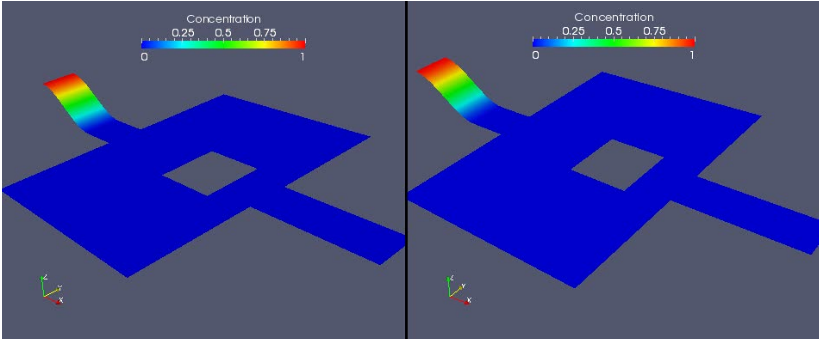
$$\begin{aligned} \partial_t \mathbf{v} - \frac{1}{Re}\Delta \mathbf{v} + \mathbf{v}.\nabla \mathbf{v} + \nabla \mathbf{p} &= \mathbf{f} \\ \operatorname{div} \mathbf{v} &= 0 \\ \partial_t \mathbf{c} + \mathbf{v}.\nabla \mathbf{c} - \frac{1}{Re \cdot Sc}\Delta \mathbf{c} &= 0 \end{aligned}$$

Domain:



Further Applications

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



movie

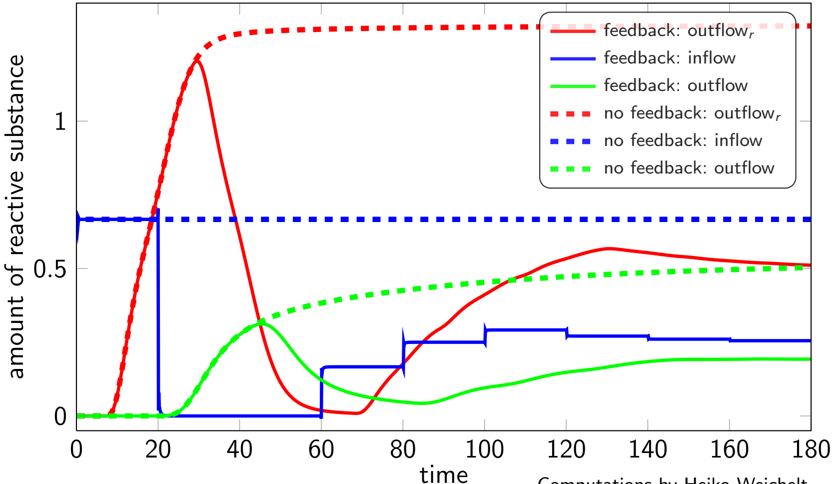
no control

piecewise constant feedback

Computations by Heiko Weichelt

Further Applications

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



Computations by Heiko Weichelt

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

1. P. Benner.
   Partial Stabilization of Descriptor Systems Using Spectral Projectors.
   In V. Olshevsky et al (eds.), *Numerical Linear Algebra in Signals, Systems, and Control*, Lecture Notes in Electrical Engineering, Springer-Verlag (to appear).

2. E. Bänsch and P. Benner
   Stabilization of Incompressible Flow Problems by Riccati-Based Feedback
   In G. Leugering et al. (eds.), *Constrained Optimization and Optimal Control for Partial Differential Equations*, ISNM, Birkhäuser, Basel, to appear.

3. P. Benner and J. Saak
   A Galerkin-Newton-ADI Method for Solving Large-Scale Algebraic Riccati Equations, *Preprint SPP1253-090* (January 2010)
   Submitted to SIMAX

4. P. Benner, J.-R. Li, and T. Penzl.
   Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems.
   Numer. Lin. Alg. Appl., vol. 15, no. 9, pp. 755–777, 2008.

5. P. Benner and T. Stykel.
   Numerical algorithms for projected generalized Riccati equations.
   Preprint, 2010.

6. R. Schneider, T. Rothaug and P. Benner.
   Flow stabilisation by Dirichlet boundary control.
   Proc. Appl. Math. Mech., vol. 8, pp. 10961–10962, 2008.