Efficient Solution of Large-Scale Saddle Point Systems Arising in Feedback Control of Flow Problems

Peter Benner\textsuperscript{1,2} Jens Saak\textsuperscript{1,2} Martin Stoll\textsuperscript{1} Heiko K. Weichelt\textsuperscript{2}

\textsuperscript{1}Computational Methods in Systems and Control Theory
Max Planck Institute for Dynamics of Complex Technical Systems Magdeburg

\textsuperscript{2}Department of Mathematics, Research group Mathematics in Industry and Technology
Chemnitz University of Technology

CSC Seminar
Magdeburg, 14 February 2012
Overview

1. Introduction
2. Discretized Stokes Control System
3. Solving Large-Scale Saddle Point Systems
4. Numerical Examples
5. Conclusions
Introduction

**Motivation**

- Asymptotic stabilization of partial differential equations
- Main application: fluid mechanics
- Later: Flow problems coupled with other field equations
- First proof of concepts:
  - "von Kármán vortex street"
  - Stokes equations to describe flow with low Reynolds number
Introduction

Motivation

- Asymptotic stabilization of partial differential equations
- Main application: fluid mechanics
- Later: Flow problems coupled with other field equations
- First proof of concepts:
  - "von Kármán vortex street"
  - Stokes equations to describe flow with low Reynolds number

Stokes Equations

\[
\begin{align*}
\frac{\partial \mathbf{v}(t, \mathbf{x})}{\partial t} - \frac{1}{\text{Re}} \Delta \mathbf{v}(t, \mathbf{x}) + \nabla p(t, \mathbf{x}) &= 0 \\
\nabla \cdot \mathbf{v}(t, \mathbf{x}) &= 0
\end{align*}
\] on \((0, \infty) \times \Omega, \quad (1)\]

with \(\Omega \subset \mathbb{R}^2\) and bounded with \(\Gamma = \partial \Omega\).
Introduction

Motivation

- Asymptotic stabilization of partial differential equations
- Main application: fluid mechanics
- Later: Flow problems coupled with other field equations
- First proof of concepts:
  - "von Kármán vortex street"
  - Stokes equations to describe flow with low Reynolds number

Stokes Equations

\[
\begin{aligned}
\frac{\partial}{\partial t} \mathbf{v} - \frac{1}{\text{Re}} \Delta \mathbf{v} + \nabla p &= 0 \\
\nabla \cdot \mathbf{v} &= 0
\end{aligned}
\]

on \((0, \infty) \times \Omega, \) \((1)\)

with \(\Omega \subset \mathbb{R}^2\) and bounded with \(\Gamma = \partial \Omega.\)
Introduction

Motivation

- Asymptotic stabilization of partial differential equations
- Main application: fluid mechanics
- Later: Flow problems coupled with other field equations
- First proof of concepts:
  - "von Kármán vortex street"
  - Stokes equations to describe flow with low Reynolds number

Navier-Stokes Equations

\[
\begin{aligned}
\frac{\partial}{\partial t} \mathbf{v} - \frac{1}{\text{Re}} \Delta \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p &= 0 \\
\nabla \cdot \mathbf{v} &= 0
\end{aligned}
\]

on \((0, \infty) \times \Omega,\)

\[
\begin{aligned}
\nabla \cdot \mathbf{v} &= 0
\end{aligned}
\]

with \(\Omega \subset \mathbb{R}^2\) and bounded with \(\Gamma = \partial \Omega.\)
Introduction

Motivation

- Asymptotic stabilization of partial differential equations
- Main application: fluid mechanics
- Later: Flow problems coupled with other field equations
- First proof of concepts:
  - "von Kármán vortex street"
  - Stokes equations to describe flow with low Reynolds number

Stokes Equations

\[
\begin{align*}
\frac{\partial}{\partial t} v - \frac{1}{Re} \Delta v + v \cdot \nabla v + \nabla p &= 0 \\
\nabla \cdot v &= 0
\end{align*}
\]

on \((0, \infty) \times \Omega, \frac{\partial}{\partial t}\nu = 0, \nabla \cdot v = 0\)

with \(\Omega \subset \mathbb{R}^2 \) and bounded with \(\Gamma = \partial \Omega\).
### Introduction

#### Basic Ideas

- Riccati-based feedback stabilization with boundary control input
- Analytical approach by **Raymond ’05–’07**
- Use *Leray projector* to project onto the correct subspace (*Helmholtz decomposition*)
- Ideas for numerical treatment based on **Bänsch/Benner ’10**
- Consider linearized Navier-Stokes equations for 2D
- Discrete projection idea by **Heinkenschloss/Sorensen/Sun ’08**
Discretized Stokes Control System

Finite Element Discretization

Applying a standard finite element discretization to (1) yields to

\[
M \frac{d}{dt} z(t) = Az(t) + Gp(t) + f(t),
\]  
\[
0 = G^T z(t).
\]
Discretized Stokes Control System

Finite Element Discretization

- Applying a standard finite element discretization to (1) yields to

\[ M \frac{d}{dt} z(t) = A z(t) + G p(t) + f(t), \]
\[ 0 = G^T z(t). \]

Properties

- velocity: \( z(t) \in \mathbb{R}^{n_v} \),
- mass matrix: \( M \in \mathbb{R}^{n_v \times n_v} \), \( M = M^T > 0 \)
- pressure: \( p(t) \in \mathbb{R}^{n_p} \),
- system matrix: \( A \in \mathbb{R}^{n_v \times n_v} \), \( A = A^T < 0 \)
- rhs: \( f(t) \in \mathbb{R}^{n_v} \),
- discretized gradient: \( G \in \mathbb{R}^{n_v \times n_p} \), \( \text{rank}(G) = n_p \)
Discretized Stokes Control System

Finite Element Discretization

- Applying a standard finite element discretization to (1) yields to

\[
M \frac{d}{dt} z(t) = Az(t) + Gp(t) + f(t),
\]

\[
0 = G^T z(t),
\]

\[
y(t) = Cz(t).
\]

Properties

- velocity: \( z(t) \in \mathbb{R}^{n_v} \),
- mass matrix: \( M \in \mathbb{R}^{n_v \times n_v}, M = M^T > 0 \)
- pressure: \( p(t) \in \mathbb{R}^{n_p} \),
- system matrix: \( A \in \mathbb{R}^{n_v \times n_v}, A = A^T < 0 \)
- rhs: \( f(t) \in \mathbb{R}^{n_v} \),
- discretized gradient: \( G \in \mathbb{R}^{n_v \times n_p}, \text{rank}(G) = n_p \)
- output: \( y(t) \in \mathbb{R}^{n_a} \),
- output matrix: \( C \in \mathbb{R}^{n_a \times n_v} \)
Discretized Stokes Control System

Finite Element Discretization

- Applying a standard finite element discretization to (1) yields to

\[
M \frac{d}{dt} z(t) = Az(t) + Gp(t) +Bu(t),
\]

(2a)

\[0 = G^T z(t),\]

(2b)

\[y(t) = Cz(t).\]

(2c)

Properties

- velocity: \( z(t) \in \mathbb{R}^{n_v}, \)
- mass matrix: \( M \in \mathbb{R}^{n_v \times n_v}, M = M^T > 0 \)
- pressure: \( p(t) \in \mathbb{R}^{n_p}, \)
- system matrix: \( A \in \mathbb{R}^{n_v \times n_v}, A = A^T < 0 \)
- rhs: \( f(t) \in \mathbb{R}^{n_v}, \)
- discretized gradient: \( G \in \mathbb{R}^{n_v \times n_p}, \text{rank}(G) = n_p \)
- output: \( y(t) \in \mathbb{R}^{n_a}, \)
- output matrix: \( C \in \mathbb{R}^{n_a \times n_v} \)
- input: \( u(t) \in \mathbb{R}^{n_r}, \)
- feedback matrix: \( B \in \mathbb{R}^{n_v \times n_r} \)
Discretized Stokes Control System

Finite Element Discretization

- Applying a standard finite element discretization to (1) yields to

\[
M \frac{d}{dt} z(t) = A z(t) + G p(t) + B u(t), \quad (2a)
\]

\[
0 = G^T z(t), \quad (2b)
\]

\[
y(t) = C z(t). \quad (2c)
\]

Properties

- Differential algebraic system (DAE) of D-index 2
- Matrix pencil:

\[
\begin{pmatrix}
A & G \\
G^T & 0
\end{pmatrix},
\begin{bmatrix}
M & 0 \\
0 & 0
\end{bmatrix}
\]
Discretized Stokes Control System

Finite Element Discretization

- Applying a standard finite element discretization to (1) yields to

\[ M \frac{d}{dt} z(t) = Az(t) + Gp(t) + Bu(t), \]  
\[ 0 = G^T z(t), \]  
\[ y(t) = Cz(t). \]

Properties

- Differential algebraic system (DAE) of D-index 2
- Matrix pencil:

\[ \left( \begin{bmatrix} A & G \\ G^T & 0 \end{bmatrix}, \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \right) \]
- Descriptor system with multiple inputs and multiple outputs (MIMO)
Discretized Stokes Control System

Finite Element Discretization

- Applying a standard finite element discretization to (1) yields to

\[ M \frac{d}{dt} z(t) = Az(t) + Gp(t) + Bu(t), \]  
\[ 0 = G^T z(t), \]  
\[ y(t) = Cz(t). \]

Properties

- Differential algebraic system (DAE) of D-index 2
- Matrix pencil:

\[
\begin{pmatrix}
A & G \\
G^T & 0
\end{pmatrix},
\begin{bmatrix}
M & 0 \\
0 & 0
\end{bmatrix}
\]

- Descriptor system with multiple inputs and multiple outputs (MIMO)
- Index reduction to apply linear quadratic control approach (LQR)
Discretized Stokes Control System

**Projection Method**

- Index reduction for balanced truncation model order reduction
- Show later why this is applicable

[Heinkenschloss/Sorensen/Sun ’08]
Discretized Stokes Control System

Projection Method

- Index reduction for balanced truncation model order reduction
- Show later why this is applicable
- Projector:

\[ \Pi := I - G(G^T M^{-1} G)^{-1} G^T M^{-1} \]
Discretized Stokes Control System

Properties of $\Pi$:

\[ \Pi := I - G(G^T M^{-1} G)^{-1} G^T M^{-1} \]

\[ \Pi^2 = \Pi \]
\[ \Pi M = M\Pi^T \]
\[ G^T x = 0 \iff \Pi^T x = x \]
\[ \text{null}(\Pi) = \text{range}(G) \]
\[ \text{range}(\Pi) = \text{null}(G^T M^{-1}) \]
Discretized Stokes Control System

Projection Method

\[ \Pi := I - G(G^T M^{-1} G)^{-1} G^T M^{-1} \]

Properties of \( \Pi \):
- \( \Pi^2 = \Pi \)
- \( \Pi M = M \Pi^T \)
- \( G^T x = 0 \iff \Pi^T x = x \)
- \( \text{null}(\Pi) = \text{range}(G) \)
- \( \text{range}(\Pi) = \text{null}(G^T M^{-1}) \)

\( \Pi^T \) seems to be discrete Leray projector
Discretized Stokes Control System

Properties of $\Pi^T$:

$\Pi^T := I - M^{-T} G (G^T M^{-1} G)^{-1} G^T$

$(\Pi^T)^2 = \Pi^T$

$\Pi M = M \Pi^T$

$G^T x = 0 \iff \Pi^T x = x$

$\text{null}(\Pi^T) = \text{range}(M^{-1} G)$

$\text{range}(\Pi^T) = \text{null}(G^T)$
Discretized Stokes Control System

Projection Method

\[(\Pi T := I - M^{-T} G (G^T M^{-1} G)^{-1} G^T)\]

Index reduction for balanced truncation model order reduction

Show later why this is applicable

Projector:

\[\Pi := I - G (G^T M - 1 G)^{-1} G^T M - 1\]

For \(G^T z(t) = 0 \Rightarrow \Pi^T z(t) = z(t)\)

\[\Pi M = M \Pi^T\]

\[\Pi^2 = \Pi\]

\[\text{null}(\Pi^T) = \text{range}(M^{-1} G)\]

\[\text{range}(\Pi^T) = \text{null}(G^T)\]

\[G^T x = 0 \iff \Pi^T x = x\]

- Projection onto divergence free functions (\(\text{div} \nu = 0\))
Discretized Stokes Control System

Properties of $\Pi^T$:

\[(\Pi^T)^2 = \Pi^T\]

\[\Pi M = M \Pi^T\]

\[G^T x = 0 \iff \Pi^T x = x\]

- Projection onto divergence free functions ($\text{div} \, \nu = 0$)
- Nullspace represents curl-free components ($\text{rot} \, \nabla p = 0$)

Index reduction for balanced truncation model order reduction

Show later why this is applicable

Projector:

\[\Pi := I - G (G^T M^{-1} G)^{-1} G^T\]

For $G^T z(t) = 0 \Rightarrow \Pi^T z(t) = z(t)$

System (2) reduces to

\[\Pi M \Pi^T \frac{\text{d}}{\text{d}t} z(t) = \Pi A \Pi^T z(t) + \Pi B u(t), \quad (3a)\]

\[y(t) = C \Pi^T z(t). \quad (3b)\]

Not invertible, because nullspace of $\Pi$ is non-trivial

Properties of $\Pi^T$:

\[\Pi^T = I - M^{-T} G (G^T M^{-1} G)^{-1} G^T\]

\[\Pi^T \Pi = \Pi \Pi^T = \Pi \text{null}(\Pi^T) = \text{range}(M^{-1} G)\]

\[\text{range}(\Pi^T) = \text{null}(G^T)\]

\[G^T x = 0 \iff \Pi^T x = x\]

- Projection onto divergence free functions ($\text{div} \, \nu = 0$)
- Nullspace represents curl-free components ($\text{rot} \, \nabla p = 0$)

Symmetric w.r.t. ($\cdot,\cdot$) (i.e., the discrete ($\cdot,\cdot$) $L_2$)

⇒ Oblique in ($\mathbb{R}^n, (\cdot,\cdot)_{2}$) but orthogonal in ($\mathbb{R}^n, (\cdot,\cdot)_M$)

Uniqueness of projectors

⇒ $\Pi^T$ is discrete version of Leray projector!
Discretized Stokes Control System

Projection Method

\[ \Pi^T := I - M^{-T} G \left( G^T M^{-1} G \right)^{-1} G^T \]

- Projection onto divergence free functions (\( \text{div} \nu = 0 \))
- Nullspace represents curl-free components (\( \text{rot} \nabla p = 0 \))
- Symmetric w.r.t. \((.,.)_M\) (i.e., the discrete \((.,.)_{L_2}\))
  \( \Rightarrow \) oblique in \((\mathbb{R}^n,.,.)_2\) but orthogonal in \((\mathbb{R}^n,.,.)_M\)

\[ \Pi \Pi^T = M \Pi^T \]
\[ \Pi^T M = M \Pi^T \]
\[ \Pi^T \Pi = \Pi^T \]
\[ \Pi^T \Xi^T = \Xi^T \Pi^T = \Xi^T \]

\( G^T \Xi = 0 \iff \Pi^T \Xi = \Xi \)
Discretized Stokes Control System

Properties of $\Pi^T$:

\[(\Pi^T)^2 = \Pi^T\]
\[\Pi M = M \Pi^T\]
\[\Pi T x = 0 \iff \Pi^T x = x\]

- Projection onto divergence free functions ($\text{div}\, v = 0$)
- Nullspace represents curl-free components ($\text{rot}\, \nabla p = 0$)
- Symmetric w.r.t. $(\cdot,\cdot)_M$ (i.e., the discrete $(\cdot,\cdot)_{L_2}$)
  \[\Rightarrow\] oblique in $(\mathbb{R}^n, (\cdot,\cdot)_2)$ but orthogonal in $(\mathbb{R}^n, (\cdot,\cdot)_M)$
- Uniqueness of projectors
  \[\Rightarrow\] $\Pi^T$ is discrete version of Leray projector!
Discretized Stokes Control System

Projection Method

- Index reduction for balanced truncation model order reduction
- Show later why this is applicable
- Projector:
  \[ \Pi := I - G(G^T M^{-1} G)^{-1} G^T M^{-1} \]
- For \( G^T z(t) = 0 \) \( \Rightarrow \Pi^T z(t) = z(t) \)

[Hinkelenschloss/Sorensen/Sun '08]
Discretized Stokes Control System

### Projection Method

**Index reduction for balanced truncation model order reduction**

- Show later why this is applicable
- **Projector:**
  \[
  \Pi := I - G(G^T M^{-1} G)^{-1} G^T M^{-1}
  \]

- For \( G^T z(t) = 0 \) \( \Rightarrow \) \( \Pi^T z(t) = z(t) \)
  \( \leftrightarrow \) resides in the correct solution manifold (*hidden manifold*):

\[
0 = G^T M^{-1} A z(t) + G^T M^{-1} G p(t) + G^T M^{-1} B u(t)
\]
**Discretized Stokes Control System**

**Projection Method**

- Index reduction for balanced truncation model order reduction
- Show later why this is applicable
- Projector:
  \[ \Pi := I - G(G^T M^{-1} G)^{-1} G^T M^{-1} \]
- For \( G^T z(t) = 0 \Rightarrow \Pi^T z(t) = z(t) \)
  \( \leftrightarrow \) resides in the correct solution manifold (hidden manifold):
  \[ 0 = G^T M^{-1} A z(t) + G^T M^{-1} G p(t) + G^T M^{-1} B u(t) \]
- System (2) reduces to
  \[ \Pi M \Pi^T \frac{d}{dt} z(t) = \Pi A \Pi^T z(t) + \Pi B u(t), \quad (3a) \]
  \[ y(t) = C \Pi^T z(t). \quad (3b) \]
Discretized Stokes Control System

**Projection Method**

- Index reduction for balanced truncation model order reduction
- Show later why this is applicable
- Projector:

\[
\Pi := I - G(G^T M^{-1} G)^{-1} G^T M^{-1}
\]

- For \( G^T z(t) = 0 \) \( \Rightarrow \) \( \Pi^T z(t) = z(t) \)
  \( \xrightarrow{\text{resides in the correct solution manifold (hidden manifold):}} \)

\[
0 = G^T M^{-1} A z(t) + G^T M^{-1} G p(t) + G^T M^{-1} B u(t)
\]

- System (2) reduces, not invertible, because nullspace of \( \Pi \) is non trivial

\[
\Pi M \Pi^T \frac{d}{dt} z(t) = \Pi A \Pi^T z(t) + \Pi B u(t), \quad (3a)
\]

\[
y(t) = C \Pi^T z(t). \quad (3b)
\]
Consider decomposition: $\Pi = \Theta_l \Theta_r^T$, with $\Theta_l, \Theta_r \in \mathbb{R}^{n_v \times (n_v-n_p)}$, such that $\Theta_l^T \Theta_r = I$. 

Projection Method

[HEINKENSCHELOSS/SORENSEN/SUN ’08]
Discretized Stokes Control System

Projection Method

Consider decomposition: \( II = \Theta_l \Theta_r^T \), with \( \Theta_l, \Theta_r \in \mathbb{R}^{n_v \times (n_v - n_p)} \), such that \( \Theta_l^T \Theta_r = I \).

Substitute the decomposition into (3) yields to

\[
\Theta_r^T M \Theta_r \frac{d}{dt} \tilde{z}(t) = \Theta_r^T A \Theta_r \tilde{z}(t) + \Theta_r^T Bu(t),
\]

\[
y(t) = C \Theta_r \tilde{z}(t),
\]

with \( \tilde{z} = \Theta_l^T z \in \mathbb{R}^{n_v - n_p} \).
Discretized Stokes Control System

Projection Method

[Heinkenschloss/Sorensen/Sun ’08]

- Consider decomposition: $\Pi = \Theta_l \Theta_r^T$, with $\Theta_l, \Theta_r \in \mathbb{R}^{nv \times (nv-np)}$, such that $\Theta_l^T \Theta_r = I$.
- Substitute the decomposition into (3) yields to

$$\Theta_r^T M \Theta_r \frac{d}{dt} \tilde{z}(t) = \Theta_r^T A \Theta_r \tilde{z}(t) + \Theta_r^T B u(t),$$

$$y(t) = C \Theta_r \tilde{z}(t),$$

with $\tilde{z} = \Theta_l^T z \in \mathbb{R}^{nv-np}$.  

Efficient Solution of Saddle Point Systems Arising in Feedback Control
Discretized Stokes Control System

Projection Method

- Consider decomposition: \( \Pi = \Theta_l \Theta_r^T \), with \( \Theta_l, \Theta_r \in \mathbb{R}^{nv \times (nv - np)} \), such that \( \Theta_l^T \Theta_r = I \).
- Substitute the decomposition into (3) yields to

\[
\mathcal{M} \frac{d}{dt} \tilde{z}(t) = \Theta_r^T A \Theta_r \tilde{z}(t) + \Theta_r^T Bu(t),
\]

\[
y(t) = C \Theta_r \tilde{z}(t),
\]

with \( \tilde{z} = \Theta_l^T z \in \mathbb{R}^{nv - np} \).
Discretized Stokes Control System

Projection Method

- Consider decomposition: \( II = \Theta_l \Theta_r^T \), with \( \Theta_l, \Theta_r \in \mathbb{R}^{n_v \times (n_v-n_p)} \), such that \( \Theta_l^T \Theta_r = I \).

- Substitute the decomposition into (3) yields to

\[
\mathcal{M} \frac{d}{dt} \tilde{z}(t) = \Theta_r^T A \Theta_r \tilde{z}(t) + \Theta_r^T B u(t),
\]
\[
y(t) = C \Theta_r \tilde{z}(t),
\]

with \( \tilde{z} = \Theta_l^T z \in \mathbb{R}^{n_v-n_p} \).
Discretized Stokes Control System

Projection Method

- Consider decomposition: \( \Pi = \Theta_l \Theta_r^T \), with \( \Theta_l, \Theta_r \in \mathbb{R}^{n_v \times (n_v-n_p)} \), such that \( \Theta_l^T \Theta_r = I \).

- Substitute the decomposition into (3) yields to

\[
\mathcal{M} \frac{d}{dt} \tilde{z}(t) = A\tilde{z}(t) + \Theta_r^T B u(t),
\]
\[
y(t) = C \Theta_r \tilde{z}(t),
\]

with \( \tilde{z} = \Theta_l^T z \in \mathbb{R}^{n_v-n_p} \).
Discretized Stokes Control System

Projection Method

Consider decomposition: $\Pi = \Theta_l \Theta_r^T$, with $\Theta_l, \Theta_r \in \mathbb{R}^{n_v \times (n_v - n_p)}$, such that $\Theta_l^T \Theta_r = I$.

Substitute the decomposition into (3) yields to

$$
\mathcal{M} \frac{d}{dt} \tilde{z}(t) = \mathcal{A} \tilde{z}(t) + \Theta_r^T B u(t),
$$

$$
y(t) = C \Theta_r \tilde{z}(t),
$$

with $\tilde{z} = \Theta_l^T z \in \mathbb{R}^{n_v - n_p}$. 

[Heinkenschloss/Sorensen/Sun '08]
Discretized Stokes Control System

Projection Method

- Consider decomposition: \( II = \Theta_l\Theta_r^T \), with \( \Theta_l, \Theta_r \in \mathbb{R}^{n_v \times (n_v - n_p)} \), such that \( \Theta_l^T\Theta_r = I \).

- Substitute the decomposition into (3) yields to

\[
\mathcal{M} \frac{d\tilde{z}(t)}{dt} = A\tilde{z}(t) + Bu(t),
\]

\[
y(t) = C\Theta_r\tilde{z}(t),
\]

with \( \tilde{z} = \Theta_l^Tz \in \mathbb{R}^{n_v-n_p} \).
Discretized Stokes Control System

**Projection Method**

- Consider decomposition: \( II = \Theta_l\Theta_r^T \), with \( \Theta_l, \Theta_r \in \mathbb{R}^{n_v \times (n_v - n_p)} \), such that \( \Theta_l^T\Theta_r = I \).

- Substitute the decomposition into (3) yields to

\[
\mathcal{M} \frac{d}{dt} \tilde{z}(t) = A\tilde{z}(t) + Bu(t),
\]

\[
y(t) = C\Theta_r\tilde{z}(t),
\]

with \( \tilde{z} = \Theta_l^Tz \in \mathbb{R}^{n_v - n_p} \).

[Heinkenschloss/Sorensen/Sun '08]
Discretized Stokes Control System

Projection Method [HEINKENSCHLOSS/SORENSEN/SUN ’08]

- Consider decomposition: \( \Pi = \Theta_l \Theta_r^T \), with \( \Theta_l, \Theta_r \in \mathbb{R}^{n_v \times (n_v - n_p)} \), such that \( \Theta_l^T \Theta_r = I \).

- Substitute the decomposition into (3) yields to

\[
\mathcal{M} \frac{d}{dt} \tilde{z}(t) = \mathcal{A} \tilde{z}(t) + \mathcal{B} u(t),
\]

\[
y(t) = \mathcal{C} \tilde{z}(t),
\]

with \( \tilde{z} = \Theta_l^T z \in \mathbb{R}^{n_v - n_p} \).
Discretized Stokes Control System

Projection Method

Consider decomposition: \( \Pi = \Theta_l \Theta_r^T \), with \( \Theta_l, \Theta_r \in \mathbb{R}^{n_v \times (n_v - n_p)} \), such that \( \Theta_l^T \Theta_r = I \).

Substitute the decomposition into (3) yields to

\[
\mathcal{M} \frac{d}{dt} \tilde{z}(t) = A \tilde{z}(t) + B u(t),
\]
\[
y(t) = C \tilde{z}(t),
\]

with \( \tilde{z} = \Theta_l^T z \in \mathbb{R}^{n_v - n_p} \).

For balanced truncation the generalized Lyapunov equations

\[
A \tilde{P} \mathcal{M}^T + \mathcal{M} \tilde{P} A^T = -BB^T,
\]
\[
A^T \tilde{Q} \mathcal{M} + \mathcal{M}^T \tilde{Q} A = -C^T C,
\]

have to be solved.
Discretized Stokes Control System

Feedback Control Approach

State space system:
\[ \mathcal{M} \dot{z} = \mathcal{A} z + \mathcal{B} u, \quad y = \mathcal{C} z \]

with \( \mathcal{M} = \mathcal{M}^T > 0 \)
Discretized Stokes Control System

Feedback Control Approach

State space system:
\[ \mathcal{M} \dot{z} = A z + B u, \quad y = C z \]
with \( \mathcal{M} = \mathcal{M}^T > 0 \)

Generalized algebraic Riccati equation:
\[ \mathcal{R}(X) = C^T C + A^T X \mathcal{M} + \mathcal{M}^T X A - \mathcal{M}^T X B B^T X \mathcal{M} = 0 \]
Discretized Stokes Control System

Feedback Control Approach

State space system:
\[ M \dot{z} = Az + Bu, \quad y = Cz \]
with \( M = M^T > 0 \)

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

Newton iteration:
\[ X^{(m+1)} = X^{(m)} + N^{(m)}, \text{ where } N^{(m)} \text{ is solution of} \]
\[ (A - BB^T X^{(m)} M) N^{(m)} M + M^T N^{(m)} (A - BB^T X^{(m)} M) = -\mathcal{R}(X^{(m)}) \]
**Discretized Stokes Control System**

**Feedback Control Approach**

State space system:
\[
\mathcal{M} \dot{z} = Az + Bu, \quad y = Cz
\]
with \( \mathcal{M} = \mathcal{M}^T > 0 \)

Generalized algebraic Riccati equation:
\[
\mathcal{R}(X) = C^T C + A^T X M + M^T X A - M^T X BB^T X M = 0
\]

Newton iteration:
\[
X^{(m+1)} = X^{(m)} + N^{(m)}, \text{ where } N^{(m)} \text{ is solution of}
\]
\[
(A - BB^T X^{(m)} M)^T N^{(m)} M + M^T N^{(m)} (A - BB^T X^{(m)} M) = -\mathcal{R}(X^{(m)})
\]

Lyapunov equation \(\Rightarrow\) ADI-Method:
\[
(A^{(m)})^T X^{(m+1)} M + M^T X^{(m+1)} A^{(m)} = -(W^{(m)})^T W^{(m)}
\]
Discretized Stokes Control System

Feedback Control Approach

State space system:
\[ \mathcal{M} \dot{z} = A z + B u, \quad y = C z \]
with \( \mathcal{M} = \mathcal{M}^T > 0 \)

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

Newton iteration:
\[ X^{(m+1)} = X^{(m)} + N^{(m)}, \quad \text{where } N^{(m)} \text{ is solution of} \]
\[ (A - B B^T X^{(m)} M)^T N^{(m)} M + M^T N^{(m)} (A - B B^T X^{(m)} M) = -\mathcal{R}(X^{(m)}) \]

Lyapunov equation \( \Rightarrow \) ADI-Method:
\[ (A^{(m)})^T X^{(m+1)} M + M^T X^{(m+1)} A^{(m)} = -(\mathcal{W}^{(m)})^T \mathcal{W}^{(m)} \]
**Avoid Projection Method**

- $\Pi$ and $\Theta_r$ are dense and non symmetric.
- Solution of $\Theta$-projected Lyapunov equation leads to solution of $\Pi$-projected Lyapunov equation.
  - $\Rightarrow$ In every ADI-step solve a system of the form

$$
\Pi (A^T - M^T X^{(m)} B B^T + p_i M^T) \Pi^T \Lambda = \Pi Y,
$$

for $\Pi^T \Lambda = \Lambda$.
- Determine $\Lambda$ as solution of the saddle point system

$$
\begin{bmatrix}
A^T - M^T X^{(m)} B B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
* 
\end{bmatrix}
= 
\begin{bmatrix}
Y \\
0
\end{bmatrix}.
$$
Discretized Stokes Control System

Avoid Projection Method

- $\Pi$ and $\Theta_r$ are dense and non symmetric.
- Solution of $\Theta$-projected Lyapunov equation leads to solution of $\Pi$-projected Lyapunov equation.
  ⇒ In every ADI-step solve a system of the form
  \[
  \Pi(A^T - M^T X^{(m)} B B^T + p_i M^T) \Pi^T \Lambda = \Pi Y,
  \]
  for $\Pi^T \Lambda = \Lambda$.
- Determine $\Lambda$ as solution of the saddle point system
  \[
  \begin{bmatrix}
    A^T - M^T X^{(m)} B B^T + p_i M^T & G \\
    & G^T & 0
  \end{bmatrix}
  \begin{bmatrix}
    \Lambda \\
    *
  \end{bmatrix}
  =
  \begin{bmatrix}
    Y \\
    0
  \end{bmatrix}.
  \]
**Discretized Stokes Control System**

### Feedback Control Approach

State space system:

\[ \mathcal{M} \dot{z} = \mathcal{A} z + \mathcal{B} u, \quad y = \mathcal{C} z \]

with \( \mathcal{M} = \mathcal{M}^T > 0 \)

Generalized algebraic Riccati equation:

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

Newton iteration:

\[ X^{(m+1)} = X^{(m)} + N^{(m)}, \text{ where } N^{(m)} \text{ is solution of} \]

\[ (\mathcal{A} - \mathcal{B} \mathcal{B}^T X^{(m)} \mathcal{M})^T N^{(m)} \mathcal{M} + \mathcal{M}^T N^{(m)} (\mathcal{A} - \mathcal{B} \mathcal{B}^T X^{(m)} \mathcal{M}) = -\mathcal{R}(X^{(m)}) \]

Lyapunov equation \( \Rightarrow \) ADI-Method:

\[ (\mathcal{A}^{(m)})^T X^{(m+1)} \mathcal{M} + \mathcal{M}^T X^{(m+1)} \mathcal{A}^{(m)} = -(\mathcal{W}^{(m)})^T \mathcal{W}^{(m)} \]

Saddle Point System:

\[
\begin{bmatrix}
\mathcal{A}^T - \mathcal{M}^T X^{(m)} \mathcal{B} \mathcal{B}^T + p_i \mathcal{M}^T \\
\mathcal{G}^T
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
* 
\end{bmatrix}
= 
\begin{bmatrix}
\mathcal{Y} \\
0
\end{bmatrix}
\]

\[ \text{[Heinkenschloss/Sorensen/Sun '08]} \]
**Discretized Stokes Control System**

**Feedback Control Approach**

State space system:
\[ \mathcal{M} \dot{z} = A z + B u, \quad y = C z \]
with \( \mathcal{M} = \mathcal{M}^T > 0 \)

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

Newton iteration:
\[ X^{(m+1)} = X^{(m)} + N^{(m)} \]
where \( N^{(m)} \) is solution of
\[ (A - B B^T X^{(m)} \mathcal{M})^T N^{(m)} \mathcal{M} + \mathcal{M}^T N^{(m)} (A - B B^T X^{(m)} \mathcal{M}) = -\mathcal{R}(X^{(m)}) \]

Lyapunov equation \( \Rightarrow \) ADI-Method:
\[ (A^{(m)})^T X^{(m+1)} \mathcal{M} + \mathcal{M}^T X^{(m+1)} A^{(m)} = -(\mathcal{W}^{(m)})^T \mathcal{W}^{(m)} \]

Saddle Point System:
\[
\begin{bmatrix}
A^T - \mathcal{M}^T X^{(m)} B B^T + p_i M^T \\
G^T
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
\star
\end{bmatrix} =
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[ \text{[HEINKENSCHLOSS/SORENSEN/SUN '08]} \]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - M^T X^{(m)} B B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
* 
\end{bmatrix}
= 
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - M^T X^{(m)} B B^T + p_i M^T \\
G^T
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
0
\end{bmatrix} =
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]
Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T \\
G^T
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
0
\end{bmatrix} =
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G^T \\
G & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix}
= \begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

- \( A = A^T \in \mathbb{R}^{n_v \times n_v} \), sparse, discretized Stokes operator, constant
- \( K^k \in \mathbb{R}^{n_v \times n_r} \), dense, feedback operator, changes in every Newton step
- \( B \in \mathbb{R}^{n_v \times n_r} \), highly sparse, boundary operator, constant
- \( p_i \in \mathbb{C}^- \), changes in every ADI step
- \( M \in \mathbb{R}^{n_v \times n_v} \), spd, sparse, mass matrix, constant
- \( G \in \mathbb{R}^{n_v \times n_p} \), sparse, full column rank, discretized gradient, constant
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G^T \\
G & 0
\end{bmatrix} \begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix} = \begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

- \( A = A^T \in \mathbb{R}^{n_v \times n_v} \), sparse, discretized Stokes operator, constant
- \( K^k \in \mathbb{R}^{n_v \times n_r} \), dense, feedback operator, changes in every Newton step
- \( B \in \mathbb{R}^{n_v \times n_r} \), highly sparse, boundary operator, constant
- \( p_i \in \mathbb{C}^- \), changes in every ADI step
- \( M \in \mathbb{R}^{n_v \times n_v} \), spd, sparse, mass matrix, constant
- \( G \in \mathbb{R}^{n_v \times n_p} \), sparse, full column rank, discretized gradient, constant
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix} =
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
\left(\begin{bmatrix}
A^T + p_i M^T & G \\
G^T & 0
\end{bmatrix} - \begin{bmatrix}
(K^{(m)})^T \\
0
\end{bmatrix} \begin{bmatrix}
B^T & 0
\end{bmatrix}\right)
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix} =
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
0 & G^T
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
* 
\end{bmatrix} =
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
\begin{pmatrix}
A - \begin{bmatrix}(K^{(m)})^T \\
0
\end{bmatrix}
& B^T \\
0 & 0
\end{pmatrix}
\begin{bmatrix}
\Lambda \\
* 
\end{bmatrix} =
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
\text{size: } n_r (n_r \ll n_v)
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
* 
\end{bmatrix}
= 
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
(A - K^T [B^T \ 0]) \begin{bmatrix}
\Lambda \\
* 
\end{bmatrix}
= 
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix}
= 
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
(A - K^T B^T)
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix}
= 
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix} \begin{bmatrix}
\Lambda \\
0
\end{bmatrix} = \begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
(A - K^T B^T) \Lambda = \begin{bmatrix}
Y \\
0
\end{bmatrix}
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
* 
\end{bmatrix}
=
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[(A - K^T B^T) \Lambda = Y\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix}
= 
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
(A - K^T B^T) \Lambda = Y
\]

Use Sherman-Morrison-Woodbury formula:

\[
(A - K^T B^T)^{-1} = (I_{n_v} + A^{-1} K^T (I_{n_r} - B^T A^{-1} K^T)^{-1} B^T) A^{-1}
\]
Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix}
=
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
(A - K^T B^T) \Lambda = Y
\]

Use Sherman-Morrison-Woodbury formula:

\[
(A - K^T B^T)^{-1} = (I_{n_v} + A^{-1} K^T (I_{n_r} - B^T A^{-1} K^T)^{-1} B^T)A^{-1}
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G^T
\end{bmatrix}
\begin{bmatrix}
\Lambda
\end{bmatrix}
= 
\begin{bmatrix}
Y
\end{bmatrix}
\]

\[
(A - K^T B^T) \Lambda = Y
\]

Use Sherman-Morrison-Woodburry formula:

\[
(A - K^T B^T)^{-1} = (I_{n_v} + A^{-1} K^T (I_{n_r} - B^T A^{-1} K^T)^{-1} B^T) A^{-1}
\]

size: \( n_r \) \((n_r \ll n_v)\)
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
* 
\end{bmatrix}
= 
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[(A - K^T B^T) \Lambda = Y\]

Use Sherman-Morrison-Woodbury formula:

\[
(A - K^T B^T)^{-1} = (I_{n_v} + A^{-1} K^T (I_{n_r} - B^T A^{-1} K^T)^{-1} B^T) A^{-1}
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix}
= \begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
(A - K^T B^T) \Lambda = Y
\]

Use Sherman-Morrison-Woodbury formula:

\[
(A - K^T B^T)^{-1} = (I_{n_v} + A^{-1} K^T (I_{n_r} - B^T A^{-1} K^T)^{-1} B^T) A^{-1}
\]

\[
\begin{bmatrix}
A^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix}
= \begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
(K^{(m)})^T
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix}
= 
\begin{bmatrix}
Y \\
0
\end{bmatrix}
\]

\[
(A - K^T B^T) \Lambda = Y
\]

Use Sherman-Morrison-Woodbury formula:

\[
(A - K^T B^T)^{-1} = (I_{n_v} + A^{-1} K^T (I_{n_r} - B^T A^{-1} K^T)^{-1} B^T) A^{-1}
\]

\[
\begin{bmatrix}
A^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
\ast
\end{bmatrix}
= 
\begin{bmatrix}
\tilde{Y} \\
0
\end{bmatrix}
\]
Solving Large-Scale Saddle Point Systems

Properties of Saddle Point System

\[
\begin{bmatrix}
A^T - (K^{(m)})^T B^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
* 
\end{bmatrix}
= 
\begin{bmatrix}
\check{Y} \\
0
\end{bmatrix}
\]

\[(A - K^T B^T) \Lambda = Y\]

Use Sherman-Morrison-Woodbury formula:

\[
(A - K^T B^T)^{-1} = (I_{n_v} + A^{-1} K^T (I_{n_r} - B^T A^{-1} K^T)^{-1} B^T) A^{-1}
\]

\[
\begin{bmatrix}
A^T + p_i M^T & G \\
G^T & 0
\end{bmatrix}
\begin{bmatrix}
\Lambda \\
* 
\end{bmatrix}
= 
\begin{bmatrix}
\check{\check{Y}} \\
0
\end{bmatrix}
\]
**Solving Large-Scale Saddle Point Systems**

<table>
<thead>
<tr>
<th>Preconditioned Iterative Solvers</th>
<th>[Elman/Silvester/Wathen ’05]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use iterative methods because sizes become quite large</td>
<td></td>
</tr>
</tbody>
</table>
Solving Large-Scale Saddle Point Systems

Preconditioned Iterative Solvers [Elman/Silvester/Wathen ’05]

- Use iterative methods because sizes become quite large
- For symmetric Stokes case: MINRES with preconditioner

\[ P = \begin{bmatrix} -P_F & 0 \\ 0 & P_{SC} \end{bmatrix}, \text{ with } P_F \approx F := A^T + p_i M^T, \]

\[ P_{SC} \approx G^T F^{-1} G \text{ (Schur complement)}. \]
Solving Large-Scale Saddle Point Systems

Preconditioned Iterative Solvers

- Use iterative methods because sizes become quite large
- For symmetric Stokes case: MINRES with preconditioner

\[
P = \begin{bmatrix}
-P_F & 0 \\
0 & P_{SC}
\end{bmatrix},
\text{ with } P_F \approx F := A^T + p_i M^T,
\]

\[
P_{SC} \approx G^T F^{-1} G \text{ (Schur complement).}
\]

- Primary focus: Handle the non-symmetric Navier-Stokes case
Solving Large-Scale Saddle Point Systems

Preconditioned Iterative Solvers [Elman/Silvester/Wathen ’05]

- Use iterative methods because sizes become quite large
- For symmetric Stokes case: MINRES with preconditioner
  \[
  P = \begin{bmatrix}
  -P_F & 0 \\
  0 & P_{SC}
  \end{bmatrix},
  \text{ with } P_F \approx F := A^T + p_i M^T, \\
  P_{SC} \approx G^T F^{-1} G (Schur complement).
  \]

- Primary focus: Handle the non-symmetric Navier-Stokes case
- Non-symmetric iterative solver: GMRES with preconditioner
  \[
  P = \begin{bmatrix}
  P_F & 0 \\
  G^T & -P_{SC}
  \end{bmatrix},
  \text{ with } P_F \approx F := A^T + p_i M^T, F \neq F^T, \\
  P_{SC} \approx G^T F^{-1} G (Schur complement).
  \]
Solving Large-Scale Saddle Point Systems

<table>
<thead>
<tr>
<th>Preconditioned Iterative Solvers</th>
<th>[Elman/Silvester/Wathen ’05]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use iterative methods because sizes become quite large</td>
<td></td>
</tr>
<tr>
<td>For symmetric Stokes case: MINRES with preconditioner</td>
<td></td>
</tr>
<tr>
<td>$P = \begin{bmatrix} -P_F &amp; 0 \ 0 &amp; P_{SC} \end{bmatrix}$, with $P_F \approx F := A^T + p_i M^T$,</td>
<td></td>
</tr>
<tr>
<td>$P_{SC} \approx G^T F^{-1} G$ (Schur complement).</td>
<td></td>
</tr>
<tr>
<td>Primary focus: Handle the non-symmetric Navier-Stokes case</td>
<td></td>
</tr>
<tr>
<td>Non-symmetric iterative solver: GMRES with preconditioner</td>
<td></td>
</tr>
<tr>
<td>$P = \begin{bmatrix} P_F &amp; 0 \ G^T &amp; -P_{SC} \end{bmatrix}$, with $P_F \approx F := A^T + p_i M^T$, $F \neq F^T$,</td>
<td></td>
</tr>
<tr>
<td>$P_{SC} \approx G^T F^{-1} G$ (Schur complement).</td>
<td></td>
</tr>
<tr>
<td>Additional cost, but only a few GMRES steps</td>
<td></td>
</tr>
</tbody>
</table>
Solving Large-Scale Saddle Point Systems

Preconditioned Iterative Solvers

\[
F = \begin{bmatrix} F & G \\ G^T & 0 \end{bmatrix}, \quad P = \begin{bmatrix} P_F & 0 \\ G^T & -P_{SC} \end{bmatrix} \Rightarrow P^{-1} = \begin{bmatrix} P_F^{-1} & 0 \\ P_{SC}^{-1}G^TP_F^{-1} & -P_{SC}^{-1} \end{bmatrix}
\]

[Elman/Silvester/Wathen ’05]
Solving Large-Scale Saddle Point Systems

Preconditioned Iterative Solvers

\[ \mathbf{F} = \begin{bmatrix} F & G \\ G^T & 0 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} P_F & 0 \\ G^T & -P_{SC} \end{bmatrix} \Rightarrow \mathbf{P}^{-1} = \begin{bmatrix} P_F^{-1} & 0 \\ P_{SC}G^TP_F^{-1} & -P_{SC}^{-1} \end{bmatrix} \]

Applying \( \mathbf{P}^{-1} \) from the left to \( \mathbf{F} \):
Solving Large-Scale Saddle Point Systems

Preconditioned Iterative Solvers

\[ \mathbf{F} = \begin{bmatrix} F & G \\ G^T & 0 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} P_F & 0 \\ G^T & -P_{SC} \end{bmatrix} \Rightarrow \mathbf{P}^{-1} = \begin{bmatrix} P_F^{-1} & 0 \\ P_{SC} G^T P_F^{-1} & -P_{SC}^{-1} \end{bmatrix} \]

Applying \( \mathbf{P}^{-1} \) from the left to \( \mathbf{F} \):

\[
\mathbf{P}^{-1} \mathbf{F} = \begin{bmatrix} P_F^{-1} F \\ P_{SC} G^T P_F^{-1} F - P_{SC}^{-1} G^T \ast P_{SC} G^T P_F^{-1} G \end{bmatrix}
\]
Solving Large-Scale Saddle Point Systems

Preconditioned Iterative Solvers

\[
F = \begin{bmatrix} F & G \\ G^T & 0 \end{bmatrix}, \quad P = \begin{bmatrix} P_F & 0 \\ G^T & -P_{SC} \end{bmatrix} \Rightarrow P^{-1} = \begin{bmatrix} \frac{1}{P_F} & 0 \\ -P_{SC}G^TP_F^{-1} & -P_{SC} \end{bmatrix}
\]

Applying \( P^{-1} \) from the left to \( F \):

\[
P^{-1}F = \begin{bmatrix} \frac{1}{P_F}F \\ -P_{SC}G^TP_F^{-1}F + P^{-1}_{SC}G^T \end{bmatrix}
\]

For now, assume \( P_F = F \) to be the best preconditioner for \( F \).
Preconditioned Iterative Solvers

\[
F = \begin{bmatrix} F & G \\ G^T & 0 \end{bmatrix}, \quad P = \begin{bmatrix} P_F & 0 \\ G^T & -P_{SC} \end{bmatrix} \Rightarrow P^{-1} = \begin{bmatrix} P_F^{-1} & 0 \\ P_{SC}^{-1}G^TP_F^{-1}P_F & -P_{SC}^{-1} \end{bmatrix}
\]

Applying \( P^{-1} \) from the left to \( F \):

\[
P^{-1}F = \begin{bmatrix} I & * \\ P_{SC}^{-1}G^TP_F^{-1}F - P_{SC}^{-1}G^T & P_{SC}^{-1}G^TP_F^{-1}G \end{bmatrix}
\]

For now, assume \( P_F = F \) to be the best preconditioner for \( F \).
Solving Large-Scale Saddle Point Systems

Preconditioned Iterative Solvers

\[
F = \begin{bmatrix} F & G \\ G^T & 0 \end{bmatrix}, \quad P = \begin{bmatrix} P_F & 0 \\ G^T & -P_{SC} \end{bmatrix} \Rightarrow P^{-1} = \begin{bmatrix} P_F^{-1} & 0 \\ P_{SC}^T P_F^{-1} G & -P_{SC}^{-1} \end{bmatrix}
\]

Applying \( P^{-1} \) from the left to \( F \):

\[
P^{-1}F = \begin{bmatrix} I & * \\ 0 & P_{SC}^{-1} G^T P_F^{-1} G \end{bmatrix}
\]

For now, assume \( P_F = F \) to be the best preconditioner for \( F \).
Solving Large-Scale Saddle Point Systems

Preconditioned Iterative Solvers

\[
\mathbf{F} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{G}^T & 0 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_F & 0 \\ \mathbf{G}^T & -\mathbf{P}_{SC} \end{bmatrix} \Rightarrow \mathbf{P}^{-1} = \begin{bmatrix} \mathbf{P}_F^{-1} & 0 \\ \mathbf{P}_{SC}^{-1} \mathbf{G}^T \mathbf{P}_F^{-1} & -\mathbf{P}_{SC}^{-1} \end{bmatrix}
\]

Applying \( \mathbf{P}^{-1} \) from the left to \( \mathbf{F} \):

\[
\mathbf{P}^{-1} \mathbf{F} = \begin{bmatrix} I & * \\ 0 & \mathbf{P}_{SC}^{-1} \mathbf{G}^T \mathbf{P}_F^{-1} \end{bmatrix}
\]

For now, assume \( \mathbf{P}_F = \mathbf{F} \) to be the best preconditioner for \( \mathbf{F} \).
Solving Large-Scale Saddle Point Systems

Preconditioned Iterative Solvers

\[ F = \begin{bmatrix} F & G \\ G^T & 0 \end{bmatrix}, \quad P = \begin{bmatrix} P_F & 0 \\ G^T & -P_{SC} \end{bmatrix} \Rightarrow P^{-1} = \begin{bmatrix} P_F^{-1} & 0 \\ -P_{SC}G^TP_F^{-1} & -P_{SC} \end{bmatrix} \]

Applying \( P^{-1} \) from the left to \( F \):

\[ P^{-1}F = \begin{bmatrix} I & * \\ 0 & P_{SC}G^TP_F^{-1}G \end{bmatrix} \]
Preconditioned Iterative Solvers

\[ F = \begin{bmatrix} F & G \\ G^T & 0 \end{bmatrix}, \quad P = \begin{bmatrix} P_F & 0 \\ G^T & -P_{SC} \end{bmatrix} \Rightarrow P^{-1} = \begin{bmatrix} P_F^{-1} & 0 \\ -P_{SC}G^TP_F^{-1} & -P_{SC}^{-1} \end{bmatrix} \]

Applying \( P^{-1} \) from the left to \( F \):

\[ P^{-1}F = \begin{bmatrix} I & * \\ 0 & P_{SC}^{-1}G^TF^{-1}G \end{bmatrix} \]

Schur Complement

- Choose Schur complement for lower right block: \( P_{SC} = G^TF^{-1}G \)
- Can not build this matrix (dense, high dimensional).
- Have to find a good approximation for \( P_{SC} \).
Solving Large-Scale Saddle Point Systems

**Schur Complement Approximation**

Approximation is derived from a least-squares commutator approach.

\[ P_{SC} \approx S_p F_p^{-1} M_p \quad \Rightarrow \quad P_{SC}^{-1} \approx M_p^{-1} F_p S_p^{-1}, \]

with \( S_p \) discretized Laplacian on pressure space, \( F_p \) system matrix on pressure space and \( M_p \) mass matrix on pressure space.
Approximation is derived from a least-squares commutator approach.

\[ P_{SC} \approx S_p F_p^{-1} M_p \quad \Rightarrow \quad P_{SC}^{-1} \approx M_p^{-1} F_p S_p^{-1}, \]

with \( S_p \) discretized Laplacian on pressure space, \( F_p \) system matrix on pressure space and \( M_p \) mass matrix on pressure space.

For Stokes case \( F_p = A_p + p_i M_p = -\nu S_p + p_i M_p \):

\[ P_{SC} \approx S_p (-\nu S_p + p_i M_p)^{-1} M_p, \]

\[ \Rightarrow P_{SC}^{-1} \approx M_p^{-1} (-\nu S_p + p_i M_p) S_p^{-1} = -\nu M_p^{-1} + p_i S_p^{-1}. \]
Solving Large-Scale Saddle Point Systems

Schur Complement Approximation

Approximation is derived from a *least-squares commutator* approach.

\[ P_{SC} \approx S_p F_p^{-1} M_p \quad \Rightarrow \quad P_{SC}^{-1} \approx M_p^{-1} F_p S_p^{-1}, \]

with \( S_p \) discretized Laplacian on pressure space, \( F_p \) system matrix on pressure space and \( M_p \) mass matrix on pressure space.

For Stokes case \( F_p = A_p + p_i M_p = -\nu S_p + p_i M_p \):

\[ P_{SC} \approx S_p (-\nu S_p + p_i M_p)^{-1} M_p, \]

\[ \Rightarrow P_{SC}^{-1} \approx M_p^{-1} (-\nu S_p + p_i M_p) S_p^{-1} = -\nu M_p^{-1} + p_i S_p^{-1}. \]

Approximation of \( P_F \)

- *Multigrid* approximation of \( F \)
- Proof of concepts: sparse direct solver in MATLAB
Problem Setting: Kármán vortex street

- Create matrices with FEM software NAVIER.
- Discretization of the domain with conformal *Taylor-Hood elements*.
- P2-P1 elements fulfill the LBB condition.
- *Bänsch-refinement* (every second level corresponds to one level of global uniform refinement):

<table>
<thead>
<tr>
<th>Level</th>
<th>( n_v )</th>
<th>( n_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3 452</td>
<td>453</td>
</tr>
<tr>
<td>2</td>
<td>8 726</td>
<td>1 123</td>
</tr>
<tr>
<td>3</td>
<td>20 512</td>
<td>2 615</td>
</tr>
<tr>
<td>4</td>
<td>45 718</td>
<td>5 783</td>
</tr>
<tr>
<td>5</td>
<td>99 652</td>
<td>12 566</td>
</tr>
<tr>
<td>6</td>
<td>211 452</td>
<td>26 572</td>
</tr>
</tbody>
</table>
Numerical Examples
Solving the Saddle Point System

Preconditioned residuals of GMRES ($p_i = -1, \text{Re} = 10$)

![Graph showing the convergence of the preconditioned residuals for different levels.](image-url)
Numerical Examples
Solving the Saddle Point System

Number of iterations for different Reynolds numbers ($\rho_i = -1$, Level 1)
Numerical Examples

Solving the Saddle Point System

Number of iterations for different ADI-shifts (Re = 10, Level 1)

number of iterations

$p_i$

-10^{-4} -10^{-2} -1 -10^{-2} -10^{-4}

-10^4 -10^2 -1 10^{-2} 10^{-4}
## Numerical Examples

### Solving Nested Iteration

<table>
<thead>
<tr>
<th>GMRES tol</th>
<th>(\varnothing) GMRES steps</th>
<th># ADI steps</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>(10^{-6})</td>
<td>13</td>
<td>&gt; 500</td>
<td>&gt; 2800 sec.</td>
</tr>
<tr>
<td>(10^{-7})</td>
<td>15</td>
<td>82</td>
<td>375 sec.</td>
</tr>
<tr>
<td>(10^{-8})</td>
<td>16</td>
<td>71</td>
<td>348 sec.</td>
</tr>
<tr>
<td>(10^{-9})</td>
<td>18</td>
<td>65</td>
<td>330 sec.</td>
</tr>
<tr>
<td>(10^{-10})</td>
<td>19</td>
<td>55</td>
<td>308 sec.</td>
</tr>
<tr>
<td>(10^{-11})</td>
<td>20</td>
<td>55</td>
<td>335 sec.</td>
</tr>
<tr>
<td>(10^{-12})</td>
<td>21</td>
<td>55</td>
<td>341 sec.</td>
</tr>
<tr>
<td>(10^{-13})</td>
<td>23</td>
<td>55</td>
<td>386 sec.</td>
</tr>
<tr>
<td>(10^{-14})</td>
<td>23</td>
<td>55</td>
<td>458 sec.</td>
</tr>
<tr>
<td>(10^{-15})</td>
<td>24</td>
<td>55</td>
<td>494 sec.</td>
</tr>
<tr>
<td>(10^{-16})</td>
<td>25</td>
<td>55</td>
<td>491 sec.</td>
</tr>
</tbody>
</table>

"direct solver":

- - | 55 | 7 sec.
Conclusions

Review

- Idea of index reduction for balanced truncation model order reduction used for the Riccati-based feedback approach
- Applied to a Stokes flow problem
- Properties of the arising saddle point systems
- Investigated preconditioners for iterative methods
- Illustrated numerical issues for the different parameters
Conclusions

Review

- Idea of index reduction for balanced truncation model order reduction used for the Riccati-based feedback approach
- Applied to a Stokes flow problem
- Properties of the arising saddle point systems
- Investigated preconditioners for iterative methods
- Illustrated numerical issues for the different parameters

Outlook

- Extend to non-symmetric Navier-Stokes equations
- Investigate the choice of ADI shifts in detail
- Use recycling techniques for iterative solvers to deal with multiple right hand sides
Conclusions

Review

- Idea of index reduction for balanced truncation model order reduction used for the Riccati-based feedback approach
- Applied to a Stokes flow problem
- Properties of the arising saddle point systems

Many thanks for your attention!

Outlook

- Extend to non-symmetric Navier-Stokes equations
- Investigate the choice of ADI shifts in detail
- Use recycling techniques for iterative solvers to deal with multiple right hand sides
### Literature


