Parameter-Robust Preconditioning for Oseen Iteration Applied to Navier–Stokes Control[1]

Santolo Leveque

University of Edinburgh

February 14, 2022

[1] Joint work with John W. Pearson, University of Edinburgh

Overview

Navier–Stokes Control Problems

PDE-Constrained Optimization Problems

A general PDE-Constrained Optimization Problem can be formulated as

$$\min_{v,u} \ \frac{1}{2} \|v - v_d\|^2_{L^2(\mathcal{Q})} + \frac{\beta}{2} \|u\|^2_{L^2(\mathcal{Q})}$$

subject to

$$\mathcal{D}v = u \qquad + \mathrm{BCs}$$

where $\mathcal{D}$ is a differential operator [Hinze, Pinnau, Ulbrich, and Ulbrich, 2010].

PDE-Constrained Optimization Problems

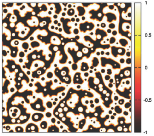
A general PDE-Constrained Optimization Problem can be formulated as

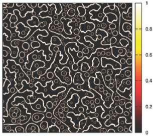
$$\min_{v,u} \ \frac{1}{2}\|v - v_d\|^2_{L^2(\mathcal{Q})} + \frac{\beta}{2}\|u\|^2_{L^2(\mathcal{Q})}$$

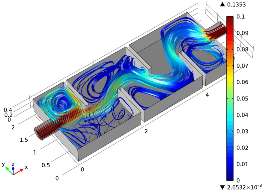
subject to

$$\mathcal{D}v = u \qquad + \mathrm{BCs}$$

where $\mathcal{D}$ is a differential operator [Hinze, Pinnau, Ulbrich, and Ulbrich, 2010].





Navier–Stokes Control Problem

Given $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, $\nu > 0$, and $t_f > 0$, we consider the following time-dependent Navier–Stokes Control Problem

$$\min_{\vec{v},\vec{u}} \mathcal{F}(\vec{v},\vec{u}) = \min_{\vec{v},\vec{u}} \frac{1}{2} \|\vec{v} - \vec{v}_d\|^2_{L^2(\mathcal{Q})} + \frac{\beta}{2} \|\vec{u}\|^2_{L^2(\mathcal{Q})}$$

subject to

$$\begin{cases} \frac{\partial \vec{v}}{\partial t} - \nu \nabla^2 \vec{v} + \vec{v} \cdot \nabla \vec{v} + \nabla p = \vec{u} + \vec{f} & \text{in } \Omega \times (0, t_f) \\ -\nabla \cdot \vec{v} = 0 & \text{in } \Omega \times (0, t_f) \\ \vec{v}(x,t) = \vec{g}(x,t) & \text{on } \partial\Omega \times (0, t_f) \\ \vec{v}(x,0) = \vec{v}_0(x) & \text{in } \Omega \end{cases}$$

where $\mathcal{Q} = \Omega \times (0, t_f)$. We suppose $\vec{v}_0(x)$ solenoidal, i.e. $\nabla \cdot \vec{v}_0(x) = 0$.

Optimize-then-Discretize Strategy

KKT-conditions

In order to obtain the Optimality Conditions, we find the stationary points of the Lagrangian (using Fréchet derivative)

$$\mathcal{L}(\vec{v}, p, \vec{u}, \vec{\zeta}, \mu) = \mathcal{F}(\vec{v}, \vec{u}) + \int_0^{t_f}\int_\Omega \left(\frac{\partial \vec{v}}{\partial t} - \nu \nabla^2 \vec{v} + \vec{v} \cdot \nabla \vec{v} + \nabla p - \vec{u} - \vec{f} \right) \cdot \vec{\zeta} + \mu \nabla \cdot \vec{v} \, dxdt.$$

This leads to the gradient equation $\beta \vec{u} - \vec{\zeta} = 0$, and

$$\left\{ \begin{array}{lll} \left. \begin{array}{ll} \frac{\partial \vec{v}}{\partial t} - \nu \nabla^2 \vec{v} + \vec{v} \cdot \nabla \vec{v} + \nabla p = \frac{1}{\beta}\vec{\zeta} + \vec{f} & \text{in } \Omega \times (0, t_f) \\ -\nabla \cdot \vec{v} = 0 & \text{in } \Omega \times (0, t_f) \\ \vec{v}(x,t) = \vec{g}(x,t) & \text{on } \partial\Omega \times (0, t_f) \\ \vec{v}(x,0) = \vec{v}_0(x) & \text{in } \Omega \end{array} \right\} & \text{state equation} \\ \left. \begin{array}{ll} -\frac{\partial \vec{\zeta}}{\partial t} - \nu \nabla^2 \vec{\zeta} - \vec{v} \cdot \nabla \vec{\zeta} + (\nabla \vec{v})^\top \vec{\zeta} & \\ \qquad + \nabla \mu = \vec{v}_d - \vec{v} & \text{in } \Omega \times (0, t_f) \\ -\nabla \cdot \vec{\zeta} = 0 & \text{in } \Omega \times (0, t_f) \\ \vec{\zeta}(x,t) = 0 & \text{on } \partial\Omega \times (0, t_f) \\ \vec{\zeta}(x,t_f) = 0 & \text{in } \Omega \end{array} \right\} & \text{adjoint equation} \end{array} \right.$$

Oseen Iteration for Navier–Stokes Control

Given $\vec{v}^{(k)}, p^{(k)}, \vec{\zeta}^{(k)}, \mu^{(k)}$ approximation of $\vec{v}$, p, $\vec{\zeta}$, μ, we consider the Oseen iterate

$$\vec{v}^{(k+1)} = \vec{v}^{(k)} + \vec{\delta v}^{(k)}, \qquad p^{(k+1)} = p^{(k)} + \delta p^{(k)},$$
$$\vec{\zeta}^{(k+1)} = \vec{\zeta}^{(k)} + \vec{\delta\zeta}^{(k)}, \qquad \mu^{(k+1)} = \mu^{(k)} + \delta\mu^{(k)},$$

with

$$\begin{cases} \frac{\partial}{\partial t}\vec{\delta v}^{(k)} - \nu\nabla^2\vec{\delta v}^{(k)} + \vec{v}^{(k)}\cdot\nabla\vec{\delta v}^{(k)} + \nabla\delta p^{(k)} - \frac{1}{\beta}\vec{\delta\zeta}^{(k)} = \vec{R}_1^{(k)} \\ -\nabla\cdot\vec{\delta v}^{(k)} = r_1^{(k)} \\ -\frac{\partial}{\partial t}\vec{\delta\zeta}^{(k)} - \nu\nabla^2\vec{\delta\zeta}^{(k)} - \vec{v}^{(k)}\cdot\nabla\vec{\delta\zeta}^{(k)} + \nabla\delta\mu^{(k)} + \vec{\delta v}^{(k)} = \vec{R}_2^{(k)} \\ -\nabla\cdot\vec{\delta\zeta}^{(k)} = r_2^{(k)} \end{cases}$$

with homogeneous initial, final, and boundary conditions.

Numerical Discretization

We employ Crank–Nicolson in time, with a staggered grid for the pressures (discretization error $\mathcal{O}(\tau^2) + \mathcal{O}(h^2)$ for Stokes control, with τ the time-step and h the mesh-size in space).

Numerical Discretization

We employ Crank–Nicolson in time, with a staggered grid for the pressures (discretization error $\mathcal{O}(\tau^2) + \mathcal{O}(h^2)$ for Stokes control, with τ the time-step and h the mesh-size in space).

In the following, we employ Taylor–Hood $\mathbf{Q}_2$–$\mathbf{Q}_1$ finite elements, with:

- M (resp., $\mathbf{M}$) is the $\mathbf{Q}_1$ (resp., $\mathbf{Q}_2$) mass matrix ($M, \mathbf{M} \succ 0$);
- K (resp., $\mathbf{K}$) is a $\mathbf{Q}_1$ (resp., $\mathbf{Q}_2$) FE discretization of $-\nabla^2$ ($K, \mathbf{K} \succ 0$);
- $N_n^{(k)}$ (resp., $\mathbf{N}_n^{(k)}$) is the $\mathbf{Q}_1$ (resp., $\mathbf{Q}_2$) FE discretization of $\vec{v}_n^{(k)} \cdot \nabla$ ($N_n^{(k)}, \mathbf{N}_n^{(k)}$ skew-symmetric if $\nabla \cdot \vec{v}_n^{(k)} = 0$);
- $W_n^{(k)}$ (resp., $\mathbf{W}_n^{(k)}$) is the $\mathbf{Q}_1$ (resp., $\mathbf{Q}_2$) possible stabilization related to $\vec{v}_n^{(k)}$ (in our tests, we used Local Projection Stabilization (LPS) [Becker and Vexler, 2007]; $W_n^{(k)}, \mathbf{W}_n^{(k)} \succeq 0$);
- B is the (negative) divergence matrix.

Preconditioning for Saddle Point Systems

Given the invertible system

$$\mathcal{A}\left[\begin{array}{c} y_1 \\ y_2 \end{array}\right] = \left[\begin{array}{c} b_1 \\ b_2 \end{array}\right], \qquad \mathcal{A} = \left[\begin{array}{cc} \Phi & \Psi_1 \\ \Psi_2 & -\Theta \end{array}\right],$$

if we precondition it with an invertible preconditioner of the form:

$$\mathcal{P} = \left[\begin{array}{cc} \Phi & 0 \\ \Psi_2 & -S \end{array}\right],$$

where $S = \Theta + \Psi_2 \Phi^{-1} \Phi_1$, the eigenvalues of the preconditioned matrix will be [Ipsen, 2001], [Murphy, Golub, and Wathen, 1999]

$$\lambda(\mathcal{P}^{-1}\mathcal{A}) = \{1\}.$$

In practice, we replace Φ and S with cheap approximation $\widetilde{\Phi}$ and $\widetilde{S}$.

Commutator Argument for Block Matrices

Let be $I \in \mathbb{R}^{m \times m}$ the identity matrix, for some $m \in \mathbb{N}$, and $\nabla_m = I \otimes \nabla$. Consider the vectorial differential operators

$$\mathcal{D} = \begin{bmatrix} \mathcal{D}^{1,1} & \dots & \mathcal{D}^{1,m} \\ \vdots & \ddots & \vdots \\ \mathcal{D}^{m,1} & \dots & \mathcal{D}^{m,m} \end{bmatrix}, \qquad \mathcal{D}_p = \begin{bmatrix} \mathcal{D}_p^{1,1} & \dots & \mathcal{D}_p^{1,m} \\ \vdots & \ddots & \vdots \\ \mathcal{D}_p^{m,1} & \dots & \mathcal{D}_p^{m,m} \end{bmatrix},$$

where $\mathcal{D}^{i,j}$ is a differential operator on the velocity space, with $D_p^{i,j}$ the corresponding differential operator on the pressure space (suppose each $D_p^{i,j}$ is well defined).

Commutator Argument for Block Matrices

Suppose the commutator

$$\mathcal{E}_m = \mathcal{D}\nabla_m - \nabla_m \mathcal{D}_p \approx 0$$

in some sense.

Commutator Argument for Block Matrices

Suppose the commutator

$$\mathcal{E}_m = \mathcal{D}\nabla_m - \nabla_m \mathcal{D}_p \approx 0$$

in some sense. After discretizing, we can rewrite

$$\mathcal{K}_p D_p^{-1} \mathcal{M}_p \approx \vec{B}\mathbf{D}^{-1}\vec{B}^\top,$$

where $\mathcal{K}_p = I \otimes K$, $\mathcal{M}_p = I \otimes M$, $\vec{B} = I \otimes B$, and

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}^{1,1} & \dots & \mathbf{D}^{1,m} \\ \vdots & \ddots & \vdots \\ \mathbf{D}^{m,1} & \dots & \mathbf{D}^{m,m} \end{bmatrix}, \qquad D_p = \begin{bmatrix} D_p^{1,1} & \dots & D_p^{1,m} \\ \vdots & \ddots & \vdots \\ D_p^{m,1} & \dots & D_p^{m,m} \end{bmatrix},$$

with $\mathbf{D}^{i,j}$ and $D_p^{i,j}$ the discretizations of $\mathcal{D}^{i,j}$ and $\mathcal{D}_p^{i,j}$, respectively.

Commutator Argument for Block Matrices

Suppose the commutator

$$\mathcal{E}_m = \mathcal{D}\nabla_m - \nabla_m \mathcal{D}_p \approx 0$$

in some sense. After discretizing, we can rewrite

$$\mathcal{K}_p D_p^{-1} \mathcal{M}_p \approx \vec{B} \mathbf{D}^{-1} \vec{B}^\top,$$

where $\mathcal{K}_p = I \otimes K$, $\mathcal{M}_p = I \otimes M$, $\vec{B} = I \otimes B$, and

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}^{1,1} & \dots & \mathbf{D}^{1,m} \\ \vdots & \ddots & \vdots \\ \mathbf{D}^{m,1} & \dots & \mathbf{D}^{m,m} \end{bmatrix}, \qquad D_p = \begin{bmatrix} D_p^{1,1} & \dots & D_p^{1,m} \\ \vdots & \ddots & \vdots \\ D_p^{m,1} & \dots & D_p^{m,m} \end{bmatrix},$$

with $\mathbf{D}^{i,j}$ and $D_p^{i,j}$ the discretizations of $\mathcal{D}^{i,j}$ and $\mathcal{D}_p^{i,j}$, respectively. This is a generalization of the argument presented in [Kay, Loghin, and Wathen, 2002].

Preconditioning Navier–Stokes Control with Crank–Nicolson in Time

Crank–Nicolson Scheme for Navier–Stokes Control

Applying Crank–Nicolson (eliminating IC and FC and multiplying the incompresibility constraints by τ), the matrix form is

$$\underbrace{\left[\begin{array}{c|c} \boldsymbol{\mathcal{A}}_1^{\mathrm{CDC},(k)} & \begin{matrix} \bar{\mathcal{B}}^\top & \\ & \bar{\mathcal{B}}^\top \end{matrix} \\ \hline \begin{matrix} \bar{\mathcal{B}} & \\ & \bar{\mathcal{B}} \end{matrix} & 0 \end{array}\right]}_{=:\boldsymbol{\mathcal{A}}_1^{\mathrm{NSC},(k)}} \begin{bmatrix} \delta v^{(k)} \\ \delta \zeta^{(k)} \\ \delta \mu^{(k)} \\ \delta p^{(k)} \end{bmatrix} = \begin{bmatrix} b_1^{(k)} \\ b_2^{(k)} \\ b_3^{(k)} \\ b_4^{(k)} \end{bmatrix}.$$

where

$$\bar{\mathcal{B}} = \tau \begin{bmatrix} B & & \\ & \ddots & \\ & & B \end{bmatrix}.$$

Crank–Nicolson Scheme for Navier–Stokes Control

The matrix $\mathcal{A}_1^{\mathrm{CDC},(k)}$ is given by

$$
\mathcal{A}_1^{\mathrm{CDC},(k)} = \left[\begin{array}{cccc|cccc}
\bar{\boldsymbol{M}} & & & & \boldsymbol{T}_0^{+,(k)} & \boldsymbol{T}_1^{-,(k)} & & \\
\bar{\boldsymbol{M}} & \ddots & & & & \ddots & \ddots & \\
 & \ddots & \bar{\boldsymbol{M}} & & & & \boldsymbol{T}_{n_t-2}^{+,(k)} & \boldsymbol{T}_{n_t-1}^{-,(k)} \\
 & & \bar{\boldsymbol{M}} & \bar{\boldsymbol{M}} & & & & \boldsymbol{T}_{n_t-1}^{+,(k)} \\
\hline
\boldsymbol{L}_1^{+,(k)} & & & & -\bar{\boldsymbol{M}}_\beta & -\bar{\boldsymbol{M}}_\beta & & \\
\boldsymbol{L}_1^{-,(k)} & \boldsymbol{L}_2^{+,(k)} & & & & \ddots & \ddots & \\
 & \ddots & \ddots & & & & -\bar{\boldsymbol{M}}_\beta & -\bar{\boldsymbol{M}}_\beta \\
 & & \boldsymbol{L}_{n_t-1}^{-,(k)} & \boldsymbol{L}_{n_t}^{+,(k)} & & & & -\bar{\boldsymbol{M}}_\beta
\end{array}\right],
$$

where $\bar{\boldsymbol{M}} = \frac{\tau}{2}\mathbf{M}$, $\bar{\boldsymbol{M}}_\beta = \frac{\tau}{2\beta}\mathbf{M}$, and

$$
\mathbf{L}_n^{+,(k)} = \tfrac{\tau}{2}(\nu\mathbf{K} + \mathbf{N}_n^{(k)} + \mathbf{W}_n^{(k)}) + \mathbf{M}, \quad \mathbf{T}_n^{+,(k)} = \tfrac{\tau}{2}(\nu\mathbf{K} - \mathbf{N}_n^{(k)} + \mathbf{W}_n^{(k)}) + \mathbf{M},
$$
$$
\mathbf{L}_n^{-,(k)} = \tfrac{\tau}{2}(\nu\mathbf{K} + \mathbf{N}_n^{(k)} + \mathbf{W}_n^{(k)}) - \mathbf{M}, \quad \mathbf{T}_n^{-,(k)} = \tfrac{\tau}{2}(\nu\mathbf{K} - \mathbf{N}_n^{(k)} + \mathbf{W}_n^{(k)}) - \mathbf{M},
$$

"Symmetrization"

Let consider the following linear transformations

$$T = \begin{bmatrix} T_1 & & & \\ & T_2 & & \\ & & T_3 & \\ & & & T_4 \end{bmatrix}$$

where

$$T_1 = \begin{bmatrix} I_{\vec{v}} & I_{\vec{v}} & & \\ & \ddots & \ddots & \\ & & I_{\vec{v}} & I_{\vec{v}} \\ & & & I_{\vec{v}} \end{bmatrix}, \quad T_3 = \begin{bmatrix} I_p & & & \\ I_p & I_p & & \\ & \ddots & \ddots & \\ & & I_p & I_p \end{bmatrix},$$

with $T_2 = T_1^\top$, $T_4 = T_3^\top$, being $I_{\vec{v}}$ and I_p the identity matrices on velocity and pressure spaces respectively.

Applying T

We consider the matrix

$$\underbrace{\left[\begin{array}{cc|cc} \multicolumn{2}{c|}{\boldsymbol{\mathcal{A}}^{\mathrm{CDC},(k)}} & \mathcal{B}_1^\top & \\ & & & \mathcal{B}_2^\top \\ \hline \mathcal{B}_1 & & \multicolumn{2}{c}{\multirow{2}{*}{0}} \\ & \mathcal{B}_2 & & \end{array}\right]}_{=T\boldsymbol{\mathcal{A}}_1^{\mathrm{NSC},(k)} =: \boldsymbol{\mathcal{A}}^{\mathrm{NSC},(k)}},$$

where

$$\mathcal{B}_1 = \tau \begin{bmatrix} B & & & \\ B & \ddots & & \\ & \ddots & \ddots & \\ & & B & B \end{bmatrix}, \qquad \mathcal{B}_2 = \tau \begin{bmatrix} B & B & & \\ & \ddots & \ddots & \\ & & \ddots & B \\ & & & B \end{bmatrix}.$$

Applying T

We consider the matrix

$$\underbrace{\left[\begin{array}{cc|cc} \multicolumn{2}{c|}{\boldsymbol{\mathcal{A}}^{\mathrm{CDC},(k)}} & \mathcal{B}_1^\top & \\ & & & \mathcal{B}_2^\top \\ \hline \mathcal{B}_1 & & \multicolumn{2}{c}{\multirow{2}{*}{0}} \\ & \mathcal{B}_2 & & \end{array}\right]}_{=T\boldsymbol{\mathcal{A}}_1^{\mathrm{NSC},(k)}=:\boldsymbol{\mathcal{A}}^{\mathrm{NSC},(k)}},$$

where

$$\mathcal{B}_1 = \tau \begin{bmatrix} B & & & \\ B & \ddots & & \\ & \ddots & \ddots & \\ & & B & B \end{bmatrix}, \qquad \mathcal{B}_2 = \tau \begin{bmatrix} B & B & & \\ & \ddots & \ddots & \\ & & \ddots & B \\ & & & B \end{bmatrix}.$$

Note that $\boldsymbol{\mathcal{A}}^{\mathrm{CDC},(k)}$ is not symmetric (unless solving instationary Stokes control with Crank–Nicolson in time).

Preconditioning $\mathcal{A}^{\mathrm{NSC},(k)}$

We employ as a preconditioner [Leveque and Pearson, 2021b]

$$\widetilde{\mathcal{P}}^{\mathrm{NSC},(k)} = \left[\begin{array}{c|c} \widetilde{\boldsymbol{\mathcal{A}}}^{\mathrm{CDC},(k)} & 0 \\ \hline \begin{matrix} \mathcal{B}_1 & \\ & \mathcal{B}_2 \end{matrix} & -\widetilde{S}^{\mathrm{NSC},(k)} \end{array}\right],$$

with $\widetilde{\boldsymbol{\mathcal{A}}}^{\mathrm{CDC},(k)}$ an approximation of the inverse of $\boldsymbol{\mathcal{A}}^{\mathrm{CDC},(k)}$, and $\widetilde{S}^{\mathrm{NSC},(k)}$ an approximation of

$$S^{\mathrm{NSC},(k)} = \tau^2 \left[\begin{array}{cc} T_3 & 0 \\ 0 & T_4 \end{array}\right] \left[\begin{array}{cc} \vec{\mathcal{B}} & \\ & \vec{\mathcal{B}} \end{array}\right] \left(\boldsymbol{\mathcal{A}}_1^{\mathrm{CDC},(k)}\right)^{-1} \left[\begin{array}{cc} \vec{\mathcal{B}}^\top & \\ & \vec{\mathcal{B}}^\top \end{array}\right].$$

Approximating $\mathcal{A}^{\mathrm{CDC},(k)}$

Following [Leveque and Pearson, 2021a], we employ as a preconditioner

$$\widetilde{\mathcal{P}}^{\mathrm{CDC},(k)} = \begin{bmatrix} \widetilde{\mathbf{\Phi}}^{\mathrm{CDC}} & 0 \\ \mathbf{\Psi}_2^{\mathrm{CDC},(k)} & -\widetilde{S}^{\mathrm{CDC},(k)} \end{bmatrix},$$

with

$$\widetilde{\mathbf{\Phi}}^{\mathrm{CDC}} = \tfrac{\tau}{2} T_1 \begin{bmatrix} \widetilde{\boldsymbol{M}} & & \\ & \ddots & \\ & & \widetilde{\boldsymbol{M}} \end{bmatrix} T_1^\top,$$

$$\widetilde{S}^{\mathrm{CDC},(k)} = T_2(\mathbf{\Psi}_2^{\mathrm{CDC},(k)} + \widehat{\boldsymbol{M}}) T_2^{-1} (\mathbf{\Phi}_D^{\mathrm{CDC}})^{-1} (\mathbf{\Psi}_1^{\mathrm{CDC},(k)} + \widehat{\boldsymbol{M}}^\top),$$

where $\widetilde{\boldsymbol{M}}$ is an approximation of $\boldsymbol{M}$, and

$$\widehat{\boldsymbol{M}} = \frac{\tau}{2\sqrt{\beta}} \begin{bmatrix} \boldsymbol{M} & & & \\ \boldsymbol{M} & \boldsymbol{M} & & \\ & \ddots & \ddots & \\ & & \boldsymbol{M} & \boldsymbol{M} \end{bmatrix}, \quad \mathbf{\Phi}_D^{\mathrm{CDC}} = \begin{bmatrix} \bar{\boldsymbol{M}} & & \\ & \ddots & \\ & & \bar{\boldsymbol{M}} \end{bmatrix}.$$

Approximating $S^{\mathrm{NSC},(k)}$

We apply a commutator argument on a differential operator $\mathcal{D}$ that "mimics" the blocks of $\boldsymbol{\mathcal{A}}_1^{\mathrm{CDC},(k)}$, and employ as an approximation of $S^{\mathrm{NSC},(k)}$ [Leveque and Pearson, 2021b]

$$\widetilde{S}^{\mathrm{NSC},(k)} = \tau^2 \begin{bmatrix} T_3 & 0 \\ 0 & T_4 \end{bmatrix} \begin{bmatrix} K & & \\ & \ddots & \\ & & K \end{bmatrix} \left(\mathcal{A}_1^{\mathrm{CDC},(k)}\right)^{-1} \begin{bmatrix} M & & \\ & \ddots & \\ & & M \end{bmatrix}.$$

How to Apply the Preconditioner

- The $(1,1)$-block is (approximately) inverted applying (a fixed number of steps of) GMRES, accelerated by the $\widetilde{\mathcal{P}}^{\mathrm{CDC},(k)}$ preconditioner for Convection–Diffusion Control.
- The matrix $\widetilde{S}^{\mathrm{CDC},(k)}$ is (approximately) inverted applying Block-Forward and Block-Backward Substitution on $(\boldsymbol{\Psi}_2^{\mathrm{CDC},(k)} + \widehat{\boldsymbol{M}})$ and $(\boldsymbol{\Psi}_1^{\mathrm{CDC},(k)} + \widehat{\boldsymbol{M}}^\top)$.
- Any mass matrix is (approximately) inverted applying (a fixed number of steps of) Chebyshev semi-iteration on $\mathbf{M}$ (resp., M).
- All the other blocks are (approximately) inverted with the action of a multigrid method (e.g.: fixed number of V-cycles).

How to Apply the Preconditioner

- The $(1,1)$-block is (approximately) inverted applying (a fixed number of steps of) GMRES, accelerated by the $\widetilde{\mathcal{P}}^{\mathrm{CDC},(k)}$ preconditioner for Convection–Diffusion Control.
- The matrix $\widetilde{S}^{\mathrm{CDC},(k)}$ is (approximately) inverted applying Block-Forward and Block-Backward Substitution on $(\boldsymbol{\Psi}_2^{\mathrm{CDC},(k)} + \widehat{\boldsymbol{M}})$ and $(\boldsymbol{\Psi}_1^{\mathrm{CDC},(k)} + \widehat{\boldsymbol{M}}^\top)$.
- Any mass matrix is (approximately) inverted applying (a fixed number of steps of) Chebyshev semi-iteration on $\mathbf{M}$ (resp., M).
- All the other blocks are (approximately) inverted with the action of a multigrid method (e.g.: fixed number of V-cycles).

Note: since the preconditioner is non-symmetric and requires an inner solve for the $(1,1)$-block $\boldsymbol{\mathcal{A}}^{\mathrm{CDC},(k)}$, for the outer solver we apply flexible GMRES (FGMRES).

Numerical Experiments (NS Control with CN)

Table: Level of grid refinement, average FGMRES iterations and CPU times, with $\nu = \frac{1}{100}$, for a range of l, β.

l	$\beta = 10^0$		$\beta = 10^{-1}$		$\beta = 10^{-2}$		$\beta = 10^{-3}$		$\beta = 10^{-4}$		$\beta = 10^{-5}$		$\beta = 10^{-6}$	
	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU
2	16	0.74	13	0.64	11	0.51	10	0.45	9	0.40	9	0.38	8	0.38
3	21	3.80	19	3.72	13	2.80	10	2.20	10	2.20	9	1.77	9	1.87
4	23	22.6	22	22.2	18	18.5	12	15.4	11	13.6	10	13.3	10	12.2
5	22	187	21	184	19	166	16	135	12	103	11	87.7	11	89.7
6	24	2141	24	2087	22	1922	18	1507	15	1272	12	973	11	913

[2] † means that the outer (Oseen) iteration did not converge in 20 iterations.

Numerical Experiments (NS Control with CN)

Table: Level of grid refinement, average FGMRES iterations and CPU times, with $\nu = \frac{1}{100}$, for a range of l, β.

l	$\beta = 10^0$		$\beta = 10^{-1}$		$\beta = 10^{-2}$		$\beta = 10^{-3}$		$\beta = 10^{-4}$		$\beta = 10^{-5}$		$\beta = 10^{-6}$	
	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU
2	16	0.74	13	0.64	11	0.51	10	0.45	9	0.40	9	0.38	8	0.38
3	21	3.80	19	3.72	13	2.80	10	2.20	10	2.20	9	1.77	9	1.87
4	23	22.6	22	22.2	18	18.5	12	15.4	11	13.6	10	13.3	10	12.2
5	22	187	21	184	19	166	16	135	12	103	11	87.7	11	89.7
6	24	2141	24	2087	22	1922	18	1507	15	1272	12	973	11	913

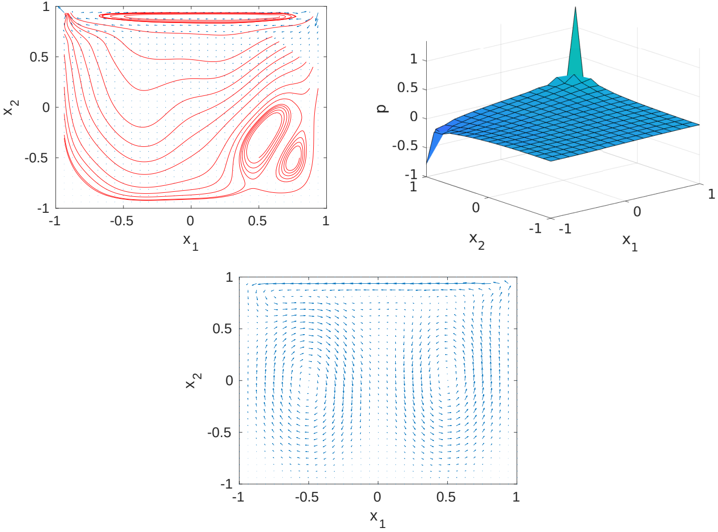
Table: Level of grid refinement, average FGMRES iterations and CPU times, with $\nu = \frac{1}{500}$, for a range of l, β.

l	$\beta = 10^0$		$\beta = 10^{-1}$		$\beta = 10^{-2}$		$\beta = 10^{-3}$		$\beta = 10^{-4}$		$\beta = 10^{-5}$		$\beta = 10^{-6}$	
	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU
2	16†[2]	0.75†	14†	0.64†	10†	0.46†	10	0.43	9	0.40	8	0.38	8	0.37
3	26†	6.49†	20†	5.70†	13	3.78	10	2.90	9	2.60	9	2.19	9	2.19
4	47	64.6	33	53.5	17	28.2	11	20.3	10	17.7	10	15.8	9	14.6
5	54	476	44	417	28	291	15	148	11	119	11	109	10	98.2
6	44	3728	37	3140	29	2472	20	1701	14	1157	11	983	11	968

[2] † means that the outer (Oseen) iteration did not converge in 20 iterations.

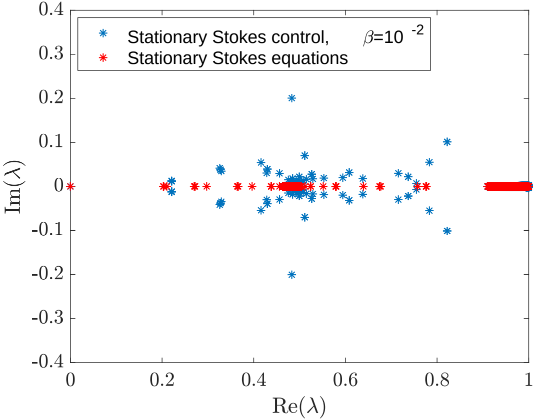
Numerical Experiments (NS Control with CN)

Figure: Solution plots for the instationary Navier–Stokes control problem, for $\nu = \frac{1}{100}$, $\beta = 10^{-1}$, and $l = 4$.



Commutator Approximation for Stationary Stokes Control

Figure: Commutator approximation for stationary Stokes control and stationary Stokes equations. In blue, eigenvalues of $\widetilde{S}_{\mathcal{A},\mathrm{S}}^{-1} S_{\mathcal{A},\mathrm{S}}$, for $\beta = 10^{-2}$, and $l = 5$. In red, eigenvalues of $M_p^{-1}(B\mathbf{K}^{-1}B^\top)$, for $l = 5$.



Conclusions

Conclusions

- Can efficiently solve Optimal Control of time-dependent Navier–Stokes equations with Crank–Nicolson in time (discretization error $\mathcal{O}(\tau^2) + \mathcal{O}(h^2)$ for Stokes control);

- preconditioner based on potent and flexible commutator argument (applied also to stationary Navier–Stokes control and time-dependent Navier–Stokes control with backward Euler in time);

- ongoing work & challenges:
  - add algebraic constraints on state/control variables;
  - employ second order methods based on Newton iterate (IPMs);
  - solve more complex PDEs;
  - solve boundary control problems;
  - consider different cost functionals;
  - apply schemes of higher-order/with better stability properties.

References

R. Becker and B. Vexler. Optimal Control of the Convection–Diffusion Equation using Stabilized Finite Element Methods. *Numer. Math.*, 106:349–367, 2007.

M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Mathematical Modelling: Theory and Applications. Springer Netherlands, 2010.

I. C. F. Ipsen. A Note on Preconditioning Nonsymmetric Matrices. *SIAM J. Sci. Comput.*, 23:1050–1051, 2001.

D. Kay, D. Loghin, and A. J. Wathen. A Preconditioner for the Steady-State Navier–Stokes Equations. *SIAM J. Sci. Comput.*, 24:237–256, 2002.

S. Leveque and J. W. Pearson. Fast Iterative Solver for the Optimal Control of Time-Dependent PDEs with Crank-Nicolson Discretization in Time. *Numer. Linear Algebra Appl.*, 2021a.

S. Leveque and J. W. Pearson. Parameter-Robust Preconditioning for Oseen Iteration Applied to Stationary and Instationary Navier–Stokes Control. *to appear in SIAM J. Sci. Comput.*, 2021b.

M. F. Murphy, G. H. Golub, and A. J. Wathen. A Note on Preconditioning for Indefinite Linear Systems. *SIAM J. Sci. Comput.*, 21:1969–1972, 1999.

Thank you

S. L. and J. W. Pearson: *Parameter-Robust Preconditioning for Oseen Iteration Applied to Stationary and Instationary Navier–Stokes Control*, to appear in *SIAM J. Sci. Comput.*, arXiv:2108.00282

Non-linear Residuals

The non-linear residuals are given by

$$\begin{cases} \vec{R}_1^{(k)} = \vec{f} - \frac{\partial}{\partial t}\vec{v}^{(k)} + \nu\nabla^2\vec{v}^{(k)} - \vec{v}^{(k)}\cdot\nabla\vec{v}^{(k)} - \nabla p^{(k)} + \dfrac{1}{\beta}\vec{\zeta}^{(k)}, \\ r_1^{(k)} = \nabla\cdot\vec{v}^{(k)}, \\ \vec{R}_2^{(k)} = \vec{v}_d - \vec{v}^{(k)} + \frac{\partial}{\partial t}\vec{\zeta}^{(k)} + \nu\nabla^2\vec{\zeta}^{(k)} + \vec{v}^{(k)}\cdot\nabla\vec{\zeta}^{(k)} \\ \qquad -(\nabla\vec{v}^{(k)})^\top\vec{\zeta}^{(k)} - \nabla\mu^{(k)}, \\ r_2^{(k)} = \nabla\cdot\vec{\zeta}^{(k)}. \end{cases}$$

Approximating $S^{\mathrm{NSC},(k)}$

We apply a commutator argument on a differential operator $\mathcal{D}$ that "mimics" the blocks of $\boldsymbol{\mathcal{A}}_1^{\mathrm{CDC},(k)}$.

$$\mathcal{D} = \left[\begin{array}{cccc|cccc} \frac{\tau}{2}\mathrm{Id} & & & & \mathcal{D}_{0,\mathrm{adj}}^{+} & \mathcal{D}_{1,\mathrm{adj}}^{-} & & \\ \frac{\tau}{2}\mathrm{Id} & \ddots & & & & \ddots & \ddots & \\ & \ddots & \frac{\tau}{2}\mathrm{Id} & & & & \mathcal{D}_{n_t-2,\mathrm{adj}}^{+} & \mathcal{D}_{n_t-1,\mathrm{adj}}^{-} \\ & & \frac{\tau}{2}\mathrm{Id} & \frac{\tau}{2}\mathrm{Id} & & & & \mathcal{D}_{n_t-1,\mathrm{adj}}^{+} \\ \hline \mathcal{D}_1^{+} & & & & -\frac{\tau}{2\beta}\mathrm{Id} & -\frac{\tau}{2\beta}\mathrm{Id} & & \\ \mathcal{D}_1^{-} & \ddots & & & & \ddots & \ddots & \\ & \ddots & \mathcal{D}_{n_t-1}^{+} & & & & -\frac{\tau}{2\beta}\mathrm{Id} & -\frac{\tau}{2\beta}\mathrm{Id} \\ & & \mathcal{D}_{n_t-1}^{-} & \mathcal{D}_{n_t}^{+} & & & & -\frac{\tau}{2\beta}\mathrm{Id} \end{array}\right],$$

where

$$\mathcal{D}_i^{+} = \tfrac{\tau}{2}\left(-\nu\nabla^2 + \vec{v}_i^{(k)}\cdot\nabla\right) + \mathrm{Id}, \quad \mathcal{D}^{-} = \tfrac{\tau}{2}\left(-\nu\nabla^2 + \vec{v}_i^{(k)}\cdot\nabla\right) - \mathrm{Id},$$
$$\mathcal{D}_{i,\mathrm{adj}}^{+} = \tfrac{\tau}{2}\left(-\nu\nabla^2 - \vec{v}_i^{(k)}\cdot\nabla\right) + \mathrm{Id}, \quad \mathcal{D}_{i,\mathrm{adj}}^{-} = \tfrac{\tau}{2}\left(-\nu\nabla^2 - \vec{v}_i^{(k)}\cdot\nabla\right) - \mathrm{Id},$$

with Id the identity operator.

Numerical Experiments (NS Control with CN)

Table: Degrees of freedom (DoF) and number of Oseen iterations. In each cell are the Oseen iterations for the given l, ν, and $\beta = 10^{-j}$, $j = 0, 1, ..., 6$.

l	DoF	$\nu = \frac{1}{100}$							$\nu = \frac{1}{500}$						
2	984	12	7	7	7	5	4	4	†	†	†	10	7	5	4
3	8496	8	8	6	7	6	4	4	†	†	7	7	7	5	4
4	70,752	6	6	5	5	4	4	3	18	14	6	5	6	5	4
5	577,728	5	4	4	4	4	4	3	9	8	6	5	4	4	3
6	4,669,824	4	4	4	3	3	3	3	5	5	5	4	4	3	3

Numerical Experiments (Time-Dependent Stokes Control with CN)

Table: Level of grid refinement, FGMRES iterations, CPU times, and errors, for a range of β (problem with exact solution).

	$\beta = 1$				$\beta = 10^{-2}$				$\beta = 10^{-4}$			
l	it	CPU	$\vec{v}_{\text{err}}$	$\vec{\zeta}_{\text{err}}$	it	CPU	$\vec{v}_{\text{err}}$	$\vec{\zeta}_{\text{err}}$	it	CPU	$\vec{v}_{\text{err}}$	$\vec{\zeta}_{\text{err}}$
2	22	0.85	4.76e-1	2.49e-1	22	0.79	5.66e-1	1.16e-1	16	0.73	8.63e0	5.45e-2
3	22	4.28	3.34e-2	5.68e-2	22	4.24	7.07e-2	3.42e-2	19	3.39	2.47e0	2.67e-2
4	23	23.9	2.25e-3	1.15e-2	24	24.1	7.35e-3	7.79e-3	20	23.2	3.73e-1	7.30e-3
5	23	200	1.74e-4	2.15e-3	27	232	6.70e-4	1.59e-3	20	162	3.84e-2	1.55e-3
6	26	2082	2.16e-5	4.00e-4	37	2960	5.97e-5	3.02e-4	23	1830	3.38e-3	3.00e-4

Numerical Experiments (NS Control with BE)

Table: Level of grid refinement, average FGMRES iterations and CPU times, with $\nu = \frac{1}{100}$, for $\tau = 0.05$ and a range of l, β.

l	$\beta = 10^0$		$\beta = 10^{-1}$		$\beta = 10^{-2}$		$\beta = 10^{-3}$		$\beta = 10^{-4}$		$\beta = 10^{-5}$		$\beta = 10^{-6}$	
	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU
2	17	6.96	14	5.70	11	4.47	11	4.21	10	4.08	12	4.93	21	8.54
3	22	18.8	19	17.7	14	13.2	11	10.9	11	10.6	13	11.8	21	19.7
4	23	45.3	22	45.4	18	39.0	14	37.5	13	35.0	15	41.2	23	61.4
5	22	190	22	196	19	169	17	148	15	126	16	136	25	220
6	25	1153	24	1099	21	979	18	809	17	729	17	685	25	1080

Table: Level of grid refinement, average FGMRES iterations and CPU times, with $\nu = \frac{1}{500}$, for $\tau = 0.05$ and a range of l, β.

l	$\beta = 10^0$		$\beta = 10^{-1}$		$\beta = 10^{-2}$		$\beta = 10^{-3}$		$\beta = 10^{-4}$		$\beta = 10^{-5}$		$\beta = 10^{-6}$	
	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU
2	16†	6.66†	14†	5.55†	11†	4.43†	10†	3.92†	10	3.84	12	4.91	22	8.87
3	26†	28.9†	20†	25.4†	14	17.2	11	13.6	11	13.3	13	15.1	22	27.9
4	43†	130†	34	120	17	64.5	13	50.6	12	47.8	14	53.6	24	82.7
5	54	467	43	417	28	294	16	160	15	164	16	187	24	263
6	39	1723	35	1552	27	1209	20	842	17	742	18	830	26	1165

Numerical Experiments (NS Control with BE)

Table: Degrees of freedom (DoF) and number of Oseen iterations, with $\tau = 0.05$. In each cell are the Oseen iterations for the given l, ν, and $\beta = 10^{-j}$, $j = 0, 1, ..., 6$.

l	DoF	$\nu = \frac{1}{100}$							$\nu = \frac{1}{500}$						
2	10,086	15	8	6	5	5	5	5	†	†	†	†	10	8	8
3	43,542	9	8	6	5	5	5	5	†	†	7	6	6	6	6
4	181,302	6	6	6	5	5	5	5	†	14	8	6	6	6	6
5	740,214	5	5	5	4	4	4	4	8	8	7	5	5	5	5
6	2,991,606	4	4	4	4	4	3	3	6	5	5	5	4	4	4

Numerical Experiments (Stationary NS Control)

Table: Level of grid refinement, average FGMRES iterations and CPU times, with $\nu = \frac{1}{100}$, for a range of l, β.

l	$\beta = 10^0$		$\beta = 10^{-1}$		$\beta = 10^{-2}$		$\beta = 10^{-3}$		$\beta = 10^{-4}$		$\beta = 10^{-5}$		$\beta = 10^{-6}$	
	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU
4	31	1.84	24	1.28	18	0.96	12	0.43	11	0.61	11	0.59	10	0.50
5	29	5.47	23	4.23	20	3.23	16	2.32	12	1.60	11	1.58	11	1.64
6	31	28.0	27	23.6	22	18.5	18	14.6	15	11.1	12	8.05	11	8.14
7	32	116	27	93.6	24	83.9	20	69.6	17	58.7	15	46.7	13	35.9
8	38	627	32	528	27	437	22	351	19	298	17	276	15	236

Table: Level of grid refinement, average FGMRES iterations and CPU times, with $\nu = \frac{1}{500}$, for a range of l, β.

l	$\beta = 10^0$		$\beta = 10^{-1}$		$\beta = 10^{-2}$		$\beta = 10^{-3}$		$\beta = 10^{-4}$		$\beta = 10^{-5}$		$\beta = 10^{-6}$	
	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU	it	CPU
4	86†	4.18†	48†	2.02†	18	1.01	12	0.68	11	0.67	10	0.57	10	0.55
5	74	15.4	49	8.02	28	4.13	14	2.08	12	1.87	11	1.73	10	1.50
6	57	55.1	39	34.2	27	22.0	20	14.6	13	8.93	11	8.86	11	8.63
7	54	192	32	111	27	90.6	21	67.3	17	49.4	12	32.5	12	37.2
8	53	878	34	561	29	472	23	369	19	292	16	232	13	172

Numerical Experiments (Stationary NS Control)

Table: Degrees of freedom (DoF) and number of Oseen iterations. In each cell are the Oseen iterations for the given l, ν, and $\beta = 10^{-j}$, $j = 0, 1, \dots, 6$.

l	DoF	$\nu = \frac{1}{100}$							$\nu = \frac{1}{500}$						
4	4422	8	6	6	5	4	4	4	†	†	9	5	4	4	4
5	18,054	7	5	5	4	4	4	3	16	10	8	6	4	4	3
6	72,966	6	4	4	4	4	4	3	11	6	5	5	4	4	3
7	293,382	5	4	3	3	3	3	3	8	4	4	4	4	4	3
8	1,176,582	4	3	3	3	3	3	3	5	3	3	3	3	3	3