Hierarchical Solution Concepts for Flow Control Problems

Michael Hinze, Stefan Turek

Michael Köster

Institute for Applied Mathematics
TU Dortmund

Kloster Banz, 24. – 26.02.2013

Flow Control Model Problem

technische universität dortmund

Distributed Control for the nonstationary Navier-Stokes equation of tracking-type for a given z on $Q = \Omega \times (0, T)$:

$$J(y, u) = \frac{1}{2}||y - z||_Q^2 + \frac{\alpha}{2}||u||_Q^2 \quad \rightarrow \quad \text{min!}$$

subject to

$$\begin{aligned} y_t - \nu \Delta y + (y\nabla)y + \nabla p &= u \quad \text{in } Q \\ -\nabla \cdot y &= 0 \quad \text{in } Q \end{aligned}$$

+ BC, constraints, init. cond.

Project aim: Solve with
$$\begin{cases} \text{costs for simulation} = O(N), \\ \text{costs for optimisation} = O(N), \\ \dfrac{\text{costs for optimisation}}{\text{costs for simulation}} \leq C \approx 10 - 50 \end{cases}$$

Flow Control Model Problem

Distributed Control for the nonstationary Navier-Stokes equation of tracking-type for a given z on $Q = \Omega \times (0, T)$:

$$J(y, u) = \frac{1}{2}||y - z||_Q^2 + \frac{\alpha}{2}||u||_Q^2 \quad \rightarrow \quad \text{min!}$$

subject to

$$\begin{aligned} y_t - \nu \Delta y + (y\nabla)y + \nabla p &= u \quad \text{in } Q \\ -\nabla \cdot y &= 0 \quad \text{in } Q \end{aligned}$$

+ BC, constraints, init. cond.

Project aim: Solve with

$$\begin{cases} \text{costs for simulation} = O(N), \\ \text{costs for optimisation} = O(N), \\ \dfrac{\text{costs for optimisation}}{\text{costs for simulation}} \leq C \approx 10 - 50 \end{cases}$$

Flow Control Model Problem

KKT-System:

$$
\begin{aligned}
y_t - \nu \Delta y + (y\nabla)y + \nabla p &= u \\
-\lambda_t - \nu \Delta \lambda - (y\nabla)\lambda + (\nabla y)^T \lambda + \nabla \xi &= y - z \\
\alpha u + \lambda &= 0
\end{aligned}
$$

+ incompressibility, BC, constraints, ...

1st period:

- Newton + Space-time multigrid in (y, λ)
- $\tilde{Q}_1/Q_0$, Q_2/P_1^{disc}, IE, CN, EOJ-Stabilisation
- Distributed control

2nd period:

- L_2 boundary control
- Control constraints

Flow Control Model Problem

KKT-System:

$$y_t - \nu\Delta y + (y\nabla)y + \nabla p = u$$
$$-\lambda_t - \nu\Delta\lambda - (y\nabla)\lambda + (\nabla y)^T\lambda + \nabla\xi = y - z$$
$$\alpha u + \lambda = 0$$

+ incompressibility, BC, constraints, ...

1st period:

- Newton + Space-time multigrid in (y, λ)
- $\tilde{Q}_1/Q_0$, Q_2/P_1^{disc}, IE, CN, EOJ-Stabilisation
- Distributed control

2nd period:

- L_2 boundary control
- Control constraints

Flow Control Model Problem

KKT-System:

$$\begin{aligned} y_t - \nu\Delta y + (y\nabla)y + \nabla p &= u \\ -\lambda_t - \nu\Delta\lambda - (y\nabla)\lambda + (\nabla y)^T\lambda + \nabla\xi &= y - z \\ \alpha u + \lambda &= 0 \end{aligned}$$

+ incompressibility, BC, constraints, ...

2nd period:

- Newton + Space-time multigrid in u
- Q_2/P_1^{disc}, IE
- Distributed control, L_2, $H^{\frac{1}{2}}$ boundary control
- Control constraints
- Comparison to previous approach

Method 1: Newton approach in u

Algorithm (Newton approach in u)

$$u_{n+1} = u_n + \bar{u}, \qquad F'(u_n)\bar{u} = -F(u_n)$$

Ingredients:

$$F(u) := \alpha u + \lambda \stackrel{!}{=} 0 \qquad\qquad F'(u)\bar{u} = \alpha \bar{u} + \bar{\lambda}$$

$$\left\{ \begin{array}{c} y_t - \nu \Delta y + \ldots = u \\ -\lambda_t - \nu \Delta \lambda + \ldots = y - z \end{array} \right\}, \qquad \left\{ \begin{array}{c} \bar{y}_t - \nu \Delta \bar{y} + \ldots = \bar{u} \\ -\bar{\lambda}_t - \nu \Delta \bar{\lambda} + \ldots = \bar{y} \end{array} \right\}$$

nonlinear simulation — linear simulation

cheap — expensive (simulation) — very expensive (CG)

Method 1: Newton approach in u

Algorithm (Newton approach in u)

$$u_{n+1} = u_n + \bar{u}, \qquad F'(u_n)\bar{u} = -F(u_n)$$

Ingredients:

$$F(u) := \alpha u + \lambda \stackrel{!}{=} 0$$

$$F'(u)\bar{u} = \alpha \bar{u} + \bar{\lambda}$$

$$\left\{ \begin{array}{c} y_t - \nu \Delta y + \ldots = u \\ -\lambda_t - \nu \Delta \lambda + \ldots = y - z \end{array} \right\},$$

nonlinear simulation

$$\left\{ \begin{array}{c} \bar{y}_t - \nu \Delta \bar{y} + \ldots = \bar{u} \\ -\bar{\lambda}_t - \nu \Delta \bar{\lambda} + \ldots = \bar{y} \end{array} \right\}$$

linear simulation

cheap — expensive (simulation) — very expensive (CG)

Method 2: Newton approach in (y, λ, p, ξ)

Algorithm (Newton approach in $x = (y, \lambda, p, \xi)$)

$$x_{n+1} = x_n + \bar{x}, \qquad G'(x_n)\bar{x} = -G(x_n)$$

Ingredients:

$$G(x) := \begin{pmatrix} y_t - \nu\Delta y + \ldots + \frac{1}{\alpha}\lambda \\ -\lambda_t - \nu\Delta\lambda + \ldots - y + z \\ \ldots \end{pmatrix} \stackrel{!}{=} 0$$

$$G'(x)\bar{x} := \begin{pmatrix} \bar{y}_t - \nu\Delta\bar{y} + \ldots + \frac{1}{\alpha}\bar{\lambda} \\ -\bar{\lambda}_t - \nu\Delta\bar{\lambda} + \ldots - \bar{y} \\ \ldots \end{pmatrix}$$

cheap expensive (simulation) very expensive (prec. BiCGStab, GMRES)

Method 2: Newton approach in (y, λ, p, ξ)

Algorithm (Newton approach in $x = (y, \lambda, p, \xi)$)

$$x_{n+1} = x_n + \bar{x}, \qquad G'(x_n)\bar{x} = -G(x_n)$$

Ingredients:

$$G(x) := \begin{pmatrix} y_t - \nu\Delta y + ... + \frac{1}{\alpha}\lambda \\ -\lambda_t - \nu\Delta\lambda + ... - y + z \\ ... \end{pmatrix} \stackrel{!}{=} 0$$

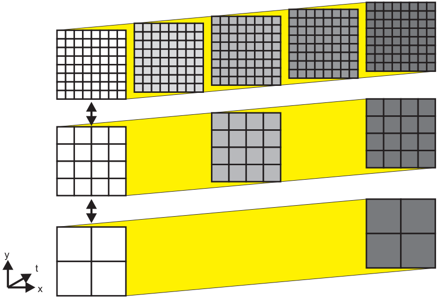
$$G'(x)\bar{x} := \begin{pmatrix} \bar{y}_t - \nu\Delta\bar{y} + ... + \frac{1}{\alpha}\bar{\lambda} \\ -\bar{\lambda}_t - \nu\Delta\bar{\lambda} + ... - \bar{y} \\ ... \end{pmatrix}$$

cheap expensive (simulation) very expensive (prec. BiCGStab, GMRES)

Expensive parts:

- $F'(u_n)\bar{u} = -F(u_n) \quad \Rightarrow$ space-time linear system in u.
- $G'(x_n)\bar{x} = -G(x_n) \quad \Rightarrow$ space-time linear system in x.

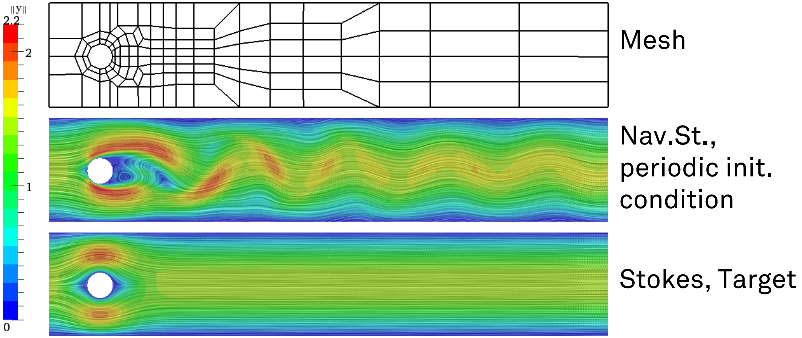
Solve using multigrid on a space-time hierarchy:



On each level: CG, BiCGStab, GMRES,... (+ preconditioner?)

Numerical example

Flow-around-cylinder (based on DFG benchmark BENCH2)



- Problem/Init. Cond.: Navier–Stokes, $Re = 100$, $t \in [0, 0.35]$
- Target flow z: Stationary Stokes flow
- Coarse mesh: Standard DFG benchmark

Numerical example

Discretisation:

- Q_2/P_1^{disc} in space, IE in time
- Coarse mesh: 520 elements, 20 timesteps, $\times 8$ per level

SLv.	#int.	#DOF(u)	#DOF(x)
2	20	87 360	237 120
3	40	682 240	1 863 680
4	80	5 391 360	14 776 320
5	160	42 864 640	117 678 080

Solver configuration (method 1+2):

- Residual reduction Newton 10^{-6}
- Residual reduction space-time MG 10^{-2}
- Stopping crit. forward/backward in space 10^{-14}
- Residual reduction monolithic MG in space 10^{-2}

Test 1: Newton solver in u

Newton-solver in the control space was:

$$u_{n+1} = u_n - F'(u_n)^{-1} F(u_n), \qquad F(u) := \alpha u + \lambda$$

CG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
2	20	20:33	0:40	5	32	31.1
3	40	4:12:29	6:38	5	35	38.1
4	80	36:54:08	52:19	5	43	42.3

MG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
2	20	coarse mesh				
3	40	5:40:00	6:38	4	8	51.3
4	80	46:03:22	52:19	5	9	52.7
5	160	297:26:50	6:13:18	5	8	47.8

Test 1: Newton solver in u

Newton-solver in the control space was:

$$u_{n+1} = u_n - F'(u_n)^{-1} F(u_n), \qquad F(u) := \alpha u + \lambda$$

CG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
2	20	20:33	0:40	5	32	31.1
3	40	4:12:29	6:38	5	35	38.1
4	80	36:54:08	52:19	5	43	42.3

MG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
2	20	coarse mesh				
3	40	5:40:00	6:38	4	8	51.3
4	80	46:03:22	52:19	5	9	52.7
5	160	297:26:50	6:13:18	5	8	47.8

Test 1: Newton solver in u

Newton-solver in the control space was:

$$u_{n+1} = u_n - F'(u_n)^{-1} F(u_n), \qquad F(u) := \alpha u + \lambda$$

CG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
2	20	20:33	0:40	5	32	31.1
3	40	4:12:29	6:38	5	35	38.1
4	80	36:54:08	52:19	5	43	42.3

MG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
2	20	coarse mesh				
3	40	5:40:00	6:38	4	8	51.3
4	80	46:03:22	52:19	5	9	52.7
5	160	297:26:50	6:13:18	5	8	47.8

Test 2: Newton solver in x

Newton-solver in the primal/dual space was:

$$x_{n+1} = x_n - G'(x_n)^{-1} G(x_n), \qquad G(x) := \begin{pmatrix} y_t - \nu \Delta y + \dots \\ -\lambda_t - \nu \Delta y + \dots \end{pmatrix}$$

BiCGStab solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
2	20	9:05	0:40	5	25	13.8
3	40	1:53:48	6:38	6	31	17.2
4	80	12:09:20	52:19	6	34	13.9

MG-solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
2	20	coarse mesh				
3	40	2:22:30	6:38	6	9	21.5
4	80	16:41:27	52:19	6	10	19.1

Test 2: Newton solver in x

Newton-solver in the primal/dual space was:

$$x_{n+1} = x_n - G'(x_n)^{-1} G(x_n), \qquad G(x) := \begin{pmatrix} y_t - \nu \Delta y + \ldots \\ -\lambda_t - \nu \Delta y + \ldots \end{pmatrix}$$

BiCGStab solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
2	20	9:05	0:40	5	25	13.8
3	40	1:53:48	6:38	6	31	17.2
4	80	12:09:20	52:19	6	34	13.9

MG-solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
2	20	coarse mesh				
3	40	2:22:30	6:38	6	9	21.5
4	80	16:41:27	52:19	6	10	19.1

Test 2: Newton solver in x

Newton-solver in the primal/dual space was:

$$x_{n+1} = x_n - G'(x_n)^{-1} G(x_n), \qquad G(x) := \begin{pmatrix} y_t - \nu \Delta y + \dots \\ -\lambda_t - \nu \Delta y + \dots \end{pmatrix}$$

BiCGStab solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
2	20	9:05	0:40	5	25	13.8
3	40	1:53:48	6:38	6	31	17.2
4	80	12:09:20	52:19	6	34	13.9

MG-solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
2	20	coarse mesh				
3	40	2:22:30	6:38	6	9	21.5
4	80	16:41:27	52:19	6	10	19.1

Comparison

- Newton in u: $u_{n+1} = u_n + \bar{u}$, $F'(u_n)\bar{u} = -F(u_n)$
- Newton in x: $x_{n+1} = x_n + \bar{x}$, $G'(x_n)\bar{x} = -G(x_n)$

	Newton in u	Newton in x
alg. complexity	low...medium $\rightarrow$ black-box applicable	high
$-F(u)$, $-G(x)$	simulation (nl.) $\rightarrow$ stopping criteria? robustness?	MatVec
apply $F'(u)$, $G'(x)$	simulation (lin.) $\rightarrow$ stopping criteria? robustness?	MatVec
preconditioner	Ø $\rightarrow$not necessary?	expensive $\rightarrow$ inexact $\surd$ $\rightarrow$ paralleliseable $\surd$
Space-time MG	$\surd$	$\surd$

Conclusion

- Newton-solver in u or (y, λ, p, ξ)?

Focus	Solver type
Black-box	Newton in u
Efficiency	Newton in (y, λ, p, ξ) $\rightarrow$ more freedom w.r.t. stopping criteria / preconditioners

- Space-time multigrid?

type of control	coarse mesh	fine mesh
Distributed	One-level	MG
Boundary	One-level	One-level

Summary

Two solution methods analysed:

- Newton in u + Newton in (y, λ, p, ξ)
- Space-time Multigrid for linear subproblems
- Distributed/boundary control, Control constraints

Main achivements:

- ”Optimal” complexity
- $T_{\text{opt}}/T_{\text{sim}} \approx 20 - 50 \quad \rightarrow$ for ’optimal’ sim.
- Newton in (y, λ, p, ξ) usually more efficient than in u

Outlook

Possible challenges for the future:

- Combination of both solvers in a ”Reduced SQP” approach.
- Detailed analysis concerning stopping criteria
- Higher RE-numbers
- 3D
- Non-isothermal, Non-Newtonian flow
- Fluid-Structure interaction?

Reduced SQP

Disadvantage in Method 1:

$$u_{n+1} = u_n - F'(u_n)^{-1} F(u_n)$$

Defect $F(u_n)$ accurate $\quad \Leftrightarrow \quad$ Accurate nonlinear simulation!

Possible alternative:

$$H(x,u) := \begin{pmatrix} y_t - \nu \Delta y + \ldots \\ -\lambda_t - \nu \Delta \lambda + \ldots \\ \alpha u + \lambda \end{pmatrix}$$

$$(x_{n+1}, u_{n+1}) := (x_n, u_n) - \text{"}F'(u_n)^{-1}\text{"} \; H(x_n, u_n)$$

$\Rightarrow$ Inexact solvers should not destroy the solution
$\Rightarrow$ No nonlinear systems in space
$\Rightarrow$ Black box applicable in subsystems

Efficiency comparison

	Newton in u	Newton in $x = (y, \lambda)$
$T_{\text{opt}}/T_{\text{sim}}$	≈ 50	≈ 20

Reason: Inexact subsolvers.

Method 1:

$$u_{n+1} = u_n - F'(u_n)^{-1} F(u_n), \qquad F(u) := \alpha u + \lambda$$

a) $F(u_n)$:

- Accurate $\Leftrightarrow$ Fw/bw simulation accurate! $\rightarrow$ expensive

b) $F'(u_n)$:

- Accurate $\Leftrightarrow$ Linear fw/bw simulation accurate! $\rightarrow$ expensive

Efficiency comparison

	Newton in u	Newton in $x = (y, \lambda)$
$T_{\text{opt}} / T_{\text{sim}}$	≈ 50	≈ 20

Reason: Inexact subsolvers.

Method 2:

$$x_{n+1} = x_n - G'(x_n)^{-1} G(x_n), \quad G(x) := \begin{pmatrix} y_t - \nu \Delta y + \dots \\ -\lambda_t - \nu \Delta y + \dots \end{pmatrix}$$

a) $G(x_n)$:

- Accurate + cheap by construction (no simulation)

b) $G'(x_n)$:

- Applied for linear residual, cheap, accurate (no simulation)
- Internal solvers inexact $\rightarrow$ less expensive

but: Memory-intensive, no checkpointing, complicated.

Linear system for method 1

The Newton solver in u reads:

$$u_{n+1} = u_n + \bar{u}, \qquad F'(u_n)\bar{u} = \alpha\bar{u} + \bar{\lambda} \stackrel{!}{=} -F(u_n)$$

with

$$\left\{ \begin{array}{r} \bar{y}_t - \nu\Delta\bar{y} + (y\nabla)\bar{y} + (\bar{y}\nabla)y + \nabla\bar{p} = \bar{u} \\ -\bar{\lambda}_t - \nu\Delta\bar{\lambda} - (y\nabla)\bar{\lambda} - (\bar{y}\nabla)\lambda + (\nabla\bar{y})^T\lambda + (\nabla y)^T\bar{\lambda} + \nabla\bar{\xi} = \bar{y} \\ \text{+ incompressibility, BC, constraints, ...} \end{array} \right\}$$

Simple defect correction solver for the linear system, $\omega \in (0, 1]$:

$$\bar{u}^{\text{new}} = \bar{u} + \omega\Big(- F(u_n) - \underbrace{(\alpha\bar{u} + \bar{\lambda})}_{=F'(u_n)\bar{u}}\Big)$$

$\Rightarrow$ One linear fw/bw solve per iteration.

Similar: CG, GMRES,...

Linear system for method 2

The Newton solver in x reads:

$$x_{n+1} = x_n + \bar{x}, \qquad G'(x_n)\bar{x} \stackrel{!}{=} -G(x_n)$$

$$G'(x)\bar{x} = \begin{pmatrix} \bar{y}_t - \nu\Delta\bar{y} + (y\nabla)\bar{y} + (\bar{y}\nabla)y + \nabla\bar{p} + \frac{1}{\alpha}\bar{\lambda} \\ -\bar{\lambda}_t - \nu\Delta\bar{\lambda} - (y\nabla)\bar{\lambda} - (\bar{y}\nabla)\lambda + (\nabla\bar{y})^T\lambda + (\nabla y)^T\bar{\lambda} + \nabla\bar{\xi} - \bar{y} \\ \text{+ incompressibility, BC, constraints, ...} \end{pmatrix}$$

Simple defect correction solver for the linear system:

$$\bar{x}^{\text{new}} = \bar{x} + C^{-1}\big(- G(x_n) - G'(x_n)\bar{x}\big)$$

$\Rightarrow C \approx G'(x_n)$ preconditioner.

Similar: CG, GMRES,...

Method 2: Construction of preconditioners

Algorithm (Defect correction loop)

$$\bar{x}_{new} = \bar{x} + C^{-1}(-G(x_n) - G'(x_n)\bar{x})$$

Discrete counterparts of $G'(x_n)$ and C (e.g., Block Jacobi):

$$G'_h(x_n) = \begin{pmatrix} A_{11} & M_{12} & & \\ M_{22} & A_{22} & M_{23} & \\ & M_{32} & A_{33} & \ddots \\ & & \ddots & \ddots \end{pmatrix}, \qquad C_h = \begin{pmatrix} A_{11} & & & \\ & A_{22} & & \\ & & A_{33} & \\ & & & \ddots \end{pmatrix}$$

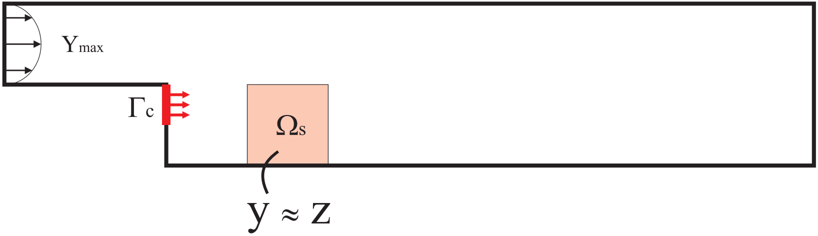
$\Rightarrow \quad C^{-1}$ = solve coupled Nav.St. (A_{ii}^{-1}) in each timestep

Flow Control Model Problem 2

Example 2: **L_2-Dirichlet boundary control + observation area**

- Space-time domain: $Q := \Omega \times (0, T)$
- Observation area: $Q_s := \Omega_s \times (0, T), \qquad \Omega_s \subseteq \Omega$
- Control boundary: $\Gamma_c \subset \partial\Omega$

Example configuration:



Flow Control Model Problem 2

Example 2: L_2**-Dirichlet boundary control + observation area**

- Space-time domain: $Q := \Omega \times (0, T)$
- Observation area: $Q_s := \Omega_s \times (0, T), \qquad \Omega_s \subseteq \Omega$
- Control boundary: $\Gamma_c \subset \partial\Omega$

Mathematical formulation: For a given z, find y with

$$J(y,u) = \frac{1}{2}||y - z||^2_{Q_s} + \frac{\alpha}{2}||u||^2_{\Gamma_c} \quad \rightarrow \quad \text{min!}$$

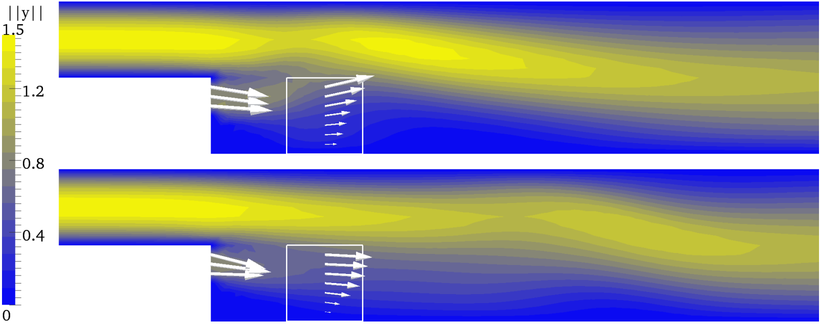
such that

$$\begin{aligned} y_t - \nu\Delta y + (y\nabla)y + \nabla p &= 0 \quad \text{in} \quad Q \\ -\nabla \cdot y &= 0 \quad \text{in} \quad Q \\ \\ y &= u \quad \text{on} \quad \Gamma_c \times (0, T) \end{aligned}$$

Flow Control Model Problem 2

Example 2: L_2-Dirichlet boundary control + observation area

- Time region $[0, T] = [0, 10]$
- Figure below for $t = 1.25$ and $t = 5.0$.



Test 1: Newton solver in u

Newton-solver in the control space was:

$$u_{n+1} = u_n - F'(u_n)^{-1} F(u_n), \qquad F(u) := \alpha u + \lambda$$

CG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
4	20	1:26:38	1:20	5	41	65.0
4	40	2:17:18	2:03	5	44	66.8
4	80	4:51:55	4:03	6	56	72.2
4	160	9:52:07	7:04	6	55	83.7

MG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
4	20	coarse mesh				
4	40	6:04:39	2:02	5	12	179.8
4	80	10:46:57	3:51	6	12	168.3
4	160	20:22:44	7:01	6	11	174.3

Test 1: Newton solver in u

Newton-solver in the control space was:

$$u_{n+1} = u_n - F'(u_n)^{-1} F(u_n), \qquad F(u) := \alpha u + \lambda$$

CG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
4	20	1:26:38	1:20	5	41	65.0
4	40	2:17:18	2:03	5	44	66.8
4	80	4:51:55	4:03	6	56	72.2
4	160	9:52:07	7:04	6	55	83.7

MG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
4	20	coarse mesh				
4	40	6:04:39	2:02	5	12	179.8
4	80	10:46:57	3:51	6	12	168.3
4	160	20:22:44	7:01	6	11	174.3

Test 1: Newton solver in u

Newton-solver in the control space was:

$$u_{n+1} = u_n - F'(u_n)^{-1} F(u_n), \qquad F(u) := \alpha u + \lambda$$

CG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
4	20	1:26:38	1:20	5	41	65.0
4	40	2:17:18	2:03	5	44	66.8
4	80	4:51:55	4:03	6	56	72.2
4	160	9:52:07	7:04	6	55	83.7

MG solver for $F'(u_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{opt}}{T_{sim}}$
4	20	coarse mesh				
4	40	6:04:39	2:02	5	12	179.8
4	80	10:46:57	3:51	6	12	168.3
4	160	20:22:44	7:01	6	11	174.3

Test 2: Newton solver in x

Newton-solver in the primal/dual space was:

$$x_{n+1} = x_n - G'(x_n)^{-1} G(x_n), \qquad G(x) := \begin{pmatrix} y_t - \nu \Delta y + \dots \\ -\lambda_t - \nu \Delta y + \dots \end{pmatrix}$$

BiCGStab solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
4	20	52:49	1:19	10	47	40.3
4	40	1:12:57	2:01	8	44	36.1
4	80	3:54:21	3:52	9	78	60.7
4	160	4:58:56	7:01	8	65	42.6

MG-solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
4	20	coarse mesh				
4	40	6:25:38	2:02	6	28	190.3
4	80	10:57:15	3:50	7	25	171.3
4	160	19:59:51	7:01	7	26	171.0

Test 2: Newton solver in x

Newton-solver in the primal/dual space was:

$$x_{n+1} = x_n - G'(x_n)^{-1} G(x_n), \qquad G(x) := \begin{pmatrix} y_t - \nu \Delta y + \dots \\ -\lambda_t - \nu \Delta y + \dots \end{pmatrix}$$

BiCGStab solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
4	20	52:49	1:19	10	47	40.3
4	40	1:12:57	2:01	8	44	36.1
4	80	3:54:21	3:52	9	78	60.7
4	160	4:58:56	7:01	8	65	42.6

MG-solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
4	20	coarse mesh				
4	40	6:25:38	2:02	6	28	190.3
4	80	10:57:15	3:50	7	25	171.3
4	160	19:59:51	7:01	7	26	171.0

Test 2: Newton solver in x

Newton-solver in the primal/dual space was:

$$x_{n+1} = x_n - G'(x_n)^{-1} G(x_n), \qquad G(x) := \begin{pmatrix} y_t - \nu\Delta y + \dots \\ -\lambda_t - \nu\Delta y + \dots \end{pmatrix}$$

BiCGStab solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
4	20	52:49	1:19	10	47	40.3
4	40	1:12:57	2:01	8	44	36.1
4	80	3:54:21	3:52	9	78	60.7
4	160	4:58:56	7:01	8	65	42.6

MG-solver for $G'(x_n)^{-1}$:

SLv.	#int	T_{opt}	T_{sim}	NL	$\sum$LIN	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
4	20	coarse mesh				
4	40	6:25:38	2:02	6	28	190.3
4	80	10:57:15	3:50	7	25	171.3
4	160	19:59:51	7:01	7	26	171.0

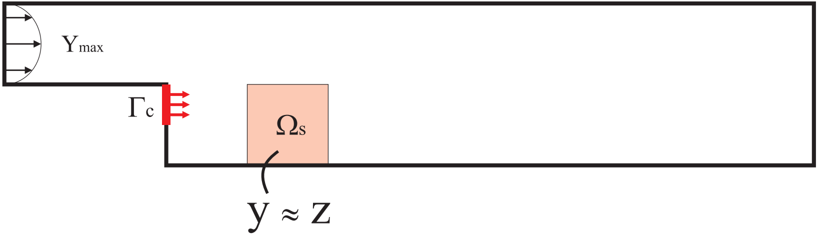
Flow Control Model Problem 3

technische universität dortmund

Example 3: $H^{\frac{1}{2}}$ **Dirichlet boundary control + observation area**

- Space-time domain: $Q := \Omega \times (0, T)$
- Observation area: $Q_s := \Omega_s \times (0, T)$, $\Omega_s \subseteq \Omega$
- Control boundary: $\Gamma_c \subset \partial\Omega$

Example configuration:



Flow Control Model Problem 3

Example 3: $H^{\frac{1}{2}}$ **Dirichlet boundary control + observation area**

- Space-time domain: $Q := \Omega \times (0, T)$
- Observation area: $Q_s := \Omega_s \times (0, T)$, $\Omega_s \subseteq \Omega$
- Control boundary: $\Gamma_c \subset \partial\Omega$

Mathematical formulation: For a given z, find y with

$$J(y, u) = \frac{1}{2}||y - z||^2_{Q_s} + \frac{\alpha}{2}(u, Su) \quad \to \quad \text{min!}$$

such that

$$\begin{aligned} y_t - \nu\Delta y + (y\nabla)y + \nabla p &= 0 \quad \text{in} \quad Q \\ -\nabla \cdot y &= 0 \quad \text{in} \quad Q \\ \\ y &= u \quad \text{on} \quad \Gamma_c \times (0, T) \end{aligned}$$

with S the Steklov-Poincare operator.

Example 3: $H^{\frac{1}{2}}$**-Dirichlet boundary control + observation area**

- Time region $[0, T] = [0, 10]$
- Figure below for $t = 1.25$ and $t = 5.0$.