Efficiently Solving Optimal Control Problems in CFD using Space-Time Multigrid Techniques

Michael Hinze, Stefan Turek

Part of the SPP1253: Optimization with PDE's

Michael Köster

Institute for Applied Mathematics
TU Dortmund

Annual Meeting, 21. – 23.09.2008

Distributed Control of nonstationary flow

Distributed Control for the nonstationary Navier-Stokes equation with tracking-type functional for a given z:

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

on $Q = \Omega \times [0, T]$ such that

$$\left.\begin{array}{rcll} y_t - \nu\Delta y + (y\nabla)y + \nabla p & = & u & \text{in } Q \\ -\nabla \cdot y & = & 0 & \text{in } Q \end{array}\right. \quad + \text{BC}$$

Design goals for a solver

Moderate performance measure; for C not too large:

$$\frac{\text{effort for optimization}}{\text{effort for simulation}} \leq C$$

By modern CFD-techniques:

$$\begin{aligned} & \text{effort for simulation} = O(N) \\ \overset{!}{\Rightarrow} \quad & \text{effort for optimization} = O(N) \end{aligned}$$

Idea:

- One-shot approach $\rightarrow$ Newton + Multigrid in space–time.
- Efficient + accurate CFD $\rightarrow$ Multigrid + FEM in space

Design goals for a solver

Moderate performance measure; for C not too large:

$$\frac{\text{effort for optimization}}{\text{effort for simulation}} \leq C$$

By modern CFD-techniques:

$$\begin{aligned} & \text{effort for simulation} = O(N) \\ \overset{!}{\Rightarrow} \quad & \text{effort for optimization} = O(N) \end{aligned}$$

Idea:

- One-shot approach $\rightarrow$ Newton + Multigrid in space–time.
- Efficient + accurate CFD $\rightarrow$ Multigrid + FEM in space

Distributed Control of nonstationary flow

Corresponding KKT-System:

$$
\begin{aligned}
y_t - \nu \Delta y + (y\nabla)y + \nabla p &= u && \text{in } Q \\
-\nabla \cdot y &= 0 && \text{in } Q \\[1em]
-\lambda_t - \nu \Delta \lambda - (y\nabla)\lambda + (\nabla y)\lambda + \nabla \xi &= (y - z) && \text{in } Q \\
-\nabla \cdot \lambda &= 0 && \text{in } Q \\[1em]
u &= -\frac{1}{\alpha}\lambda && \text{in } Q
\end{aligned}
$$

+ boundary conditions
+ initial condition
+ terminal condition $\lambda(T) = \gamma(y(T) - z(T))$ in Ω

Distributed Control of nonstationary flow

Corresponding KKT-System:

$$
\begin{array}{rcll}
y_t + N(y)y + \nabla p + \frac{1}{\alpha}\lambda & = & 0 & \text{in } Q \\
-\nabla \cdot y & = & 0 & \text{in } Q
\end{array}
$$

$$
\begin{array}{rcll}
-\lambda_t + N^*(y)\lambda + \nabla \xi - y & = & -z & \text{in } Q \\
-\nabla \cdot \lambda & = & 0 & \text{in } Q
\end{array}
$$

Distributed Control of nonstationary flow

Corresponding KKT-System:

$$\begin{pmatrix} y_t \\ -\lambda_t \\ 0 \\ 0 \end{pmatrix} + \underbrace{\begin{pmatrix} N(y) & \frac{1}{\alpha} & \nabla & \\ -I & N^*(y) & & \nabla \\ -\nabla\cdot & & 0 & \\ & -\nabla\cdot & & 0 \end{pmatrix}}_{\begin{pmatrix} \tilde{N}(y) & \nabla \\ -\nabla\cdot & 0 \end{pmatrix}} \begin{pmatrix} y \\ \lambda \\ p \\ \xi \end{pmatrix} = \begin{pmatrix} 0 \\ -z \\ 0 \\ 0 \end{pmatrix}$$

$\Rightarrow$ saddle point structure

Space–time discretisation

- Time discretisation:
  Backward Euler (later: 2nd order Crank-Nicolson, FS-θ)

$$\begin{aligned} \frac{y_i - y_{i-1}}{\Delta t} \; + \; N(y_i)y_i \; + \; \nabla p_i \; + \; \frac{1}{\alpha}\lambda_i \; &= \; \dots \\ \frac{\lambda_i - \lambda_{i+1}}{\Delta t} \; + \; N^*(y_i)\lambda_i \; + \; \nabla \xi_i \; - \; y_i \; &= \; \dots \end{aligned}$$

- Space discretisation:
  LBB-stable Finite Elements ($\tilde{Q}_1/Q_0$, later Q_2/P_1)
  on general 2D meshes (later: 3D)

Space–time discretisation

⇒ Space–time system

$$A(x)x = b$$

Here (for n timesteps, $x_k = x(t_k)$):

$$x = (\underbrace{y_0, \lambda_0, p_0, \xi_0}_{x_0 = x(t_0)}, \;\ldots,\; \underbrace{y_n, \lambda_n, p_n, \xi_n}_{x_n = x(t_n)})$$

Space–time discretisation

$A(x)$ has a very special structure! E.g. for 2 timesteps:

$$A(x)x =$$

$$\left(\begin{array}{c|c|c}
\begin{array}{cc|cc} NST & & -B & \\ -M & NST^* & & -B \\ \hline -B^T & & 0 & \\ & -B^T & & 0 \end{array} & -\frac{M}{\Delta t} & \\
\hline
-\frac{M}{\Delta t} & \begin{array}{cc|cc} NST & \frac{M}{\alpha} & -B & \\ -M & NST^* & & -B \\ \hline -B^T & & 0 & \\ & -B^T & & 0 \end{array} & -\frac{M}{\Delta t} \\
\hline
 & -\frac{M}{\Delta t} & \begin{array}{cc|cc} NST & \frac{M}{\alpha} & -B & \\ -c(\gamma,\Delta t)M & NST^* & & -B \\ \hline -B^T & & 0 & \\ & -B^T & & 0 \end{array}
\end{array}\right)
\left(\begin{array}{c} y_0 \\ \lambda_0 \\ p_0 \\ \xi_0 \\ \hline y_1 \\ \lambda_1 \\ p_1 \\ \xi_1 \\ \hline y_2 \\ \lambda_2 \\ p_2 \\ \xi_2 \end{array}\right)$$

$\rightarrow$ Sparse, (block) tridiagonal system

Design of a One-Shot solver

- Nonlinearity: Newton method for quadratic convergence

$$x^{i+1} = x^i + A'^{-1}(x^i)(b - A(x^i)x^i)$$

- Linear subproblems: space-time Multigrid solver

  $\rightarrow$ Convergence rates independent of refinement level of the space-time mesh

Space–time multigrid ingredients

Essential multigrid components:

- Mesh hierarchy for a space-time cylinder $Q = \Omega \times [0, T]$:
  Choose arbitrary space-time coarse mesh and refine!

- Prolongation/Restriction in space + time.
  Combination of FE in space & FD in time.

- An efficient smoother! E.g. with $\tilde{A} := A'(x^i)$:

$$v^{j+1} = v^j + \omega P^{-1}(b - \tilde{A} v^j), \qquad j = 1, ..., \mathrm{NSM}$$

What to use as preconditioner P?

Space–time multigrid ingredients

Essential multigrid components:

- Mesh hierarchy for a space-time cylinder $Q = \Omega \times [0, T]$:
  Choose arbitrary space-time coarse mesh and refine!
- Prolongation/Restriction in space + time.
  Combination of FE in space & FD in time.
- An efficient smoother! E.g. with $\tilde{A} := A'(x^i)$:

$$v^{j+1} = v^j + \omega P^{-1}(b - \tilde{A}v^j), \qquad j = 1, ..., \mathrm{NSM}$$

What to use as preconditioner P?

Space–time discretisation and preconditioner

$\tilde{A}$ in compressed form (omitting B and B^T here):

$$
\left(\begin{array}{cc|cc|cc|cc}
NST & & & & & & & \\
-M & NST^* & & -\frac{M}{\Delta t} & & & & \\
\hline
-\frac{M}{\Delta t} & & NST & \frac{1}{\alpha}M & & & & \\
 & & -M & NST^* & & -\frac{M}{\Delta t} & & \\
\hline
 & & -\frac{M}{\Delta t} & & NST & \frac{1}{\alpha}M & & \\
 & & & & -M & NST^* & & -\frac{M}{\Delta t} \\
\hline
 & & & & & \dots & & \dots
\end{array}\right)
$$

$\Rightarrow$ Block-Jacobi preconditioner:

$$
P := P_{Jac} := \left(\begin{array}{cc|cc|cc|cc}
NST & & & & & & & \\
-M & NST^* & & & & & & \\
\hline
 & & NST & \frac{1}{\alpha}M & & & & \\
 & & -M & NST^* & & & & \\
\hline
 & & & & NST & \frac{1}{\alpha}M & & \\
 & & & & -M & NST^* & & \\
\hline
 & & & & & & & \dots
\end{array}\right)
$$

Space–time discretisation and preconditioner

$\tilde{A}$ in compressed form (omitting B and B^T here):

$$
\left(\begin{array}{cc|cc|cc|c}
NST & & & & & & \\
-M & NST^* & & -\frac{M}{\Delta t} & & & \\
\hline
-\frac{M}{\Delta t} & & NST & \frac{1}{\alpha}M & & & \\
 & & -M & NST^* & & -\frac{M}{\Delta t} & \\
\hline
 & & -\frac{M}{\Delta t} & & NST & \frac{1}{\alpha}M & \\
 & & & & -M & NST^* & -\frac{M}{\Delta t} \\
\hline
 & & & & & \dots & \dots
\end{array}\right)
$$

$\Rightarrow$ Block-Jacobi preconditioner:

$$
P := P_{Jac} := \left(\begin{array}{cc|cc|cc|c}
NST & & & & & & \\
-M & NST^* & & & & & \\
\hline
 & & NST & \frac{1}{\alpha}M & & & \\
 & & -M & NST^* & & & \\
\hline
 & & & & NST & \frac{1}{\alpha}M & \\
 & & & & -M & NST^* & \\
\hline
 & & & & & & \dots
\end{array}\right)
$$

Space–time discretisation and preconditioner

$\tilde{A}$ in compressed form (omitting B and B^T here):

$$
\left(\begin{array}{cc|cc|cc|cc}
NST & & & & & & & \\
-M & NST^* & & -\frac{M}{\Delta t} & & & & \\
\hline
-\frac{M}{\Delta t} & & NST & \frac{1}{\alpha}M & & & & \\
& & -M & NST^* & & -\frac{M}{\Delta t} & & \\
\hline
& & -\frac{M}{\Delta t} & & NST & \frac{1}{\alpha}M & & \\
& & & & -M & NST^* & & -\frac{M}{\Delta t} \\
\hline
& & & & & \dots & & \dots
\end{array}\right)
$$

$\Rightarrow$ Symmetric Gauß-Seidel-like preconditioner $P := P_{SGS}$.

In every timestep:

$$
\begin{pmatrix} -\frac{M}{\Delta t} & 0 \\ 0 & 0 \end{pmatrix} v_{i-1} + \color{red}{\begin{pmatrix} NST & \frac{1}{\alpha}M \\ -M & NST^* \end{pmatrix}} \color{red}{v_i} + \begin{pmatrix} 0 & 0 \\ 0 & -\frac{M}{\Delta t} \end{pmatrix} v_{i+1} = \dots
$$

$\rightarrow v_i$

Space–time preconditioner

Essential: To use $P^{-1}_{Jac/SGS}$, apply for every timestep:

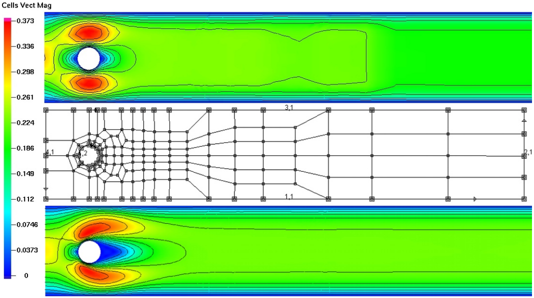
$$\left(\begin{array}{cc|cc} NST & \frac{1}{\alpha}M & B & \\ -M & NST^* & & B \\ \hline B^T & & 0 & \\ & B^T & & 0 \end{array}\right)^{-1}$$

saddle point structure $\Rightarrow$ use efficient CFD techniques!

$$\left.\begin{array}{l} \bullet\ \text{Multigrid in space} \\ \bullet\ \text{Smoother: mod. LPSC (VANKA-like)} \\ \quad \rightarrow \text{Primal and dual coupled!} \end{array}\right\} \textbf{FeatFlow!}$$

Numerical example: Flow-around-cylinder

- Target flow z: Stokes flow, $t \in [0,1]$, starting from rest



Stokes at $t = 1$

Mesh

uncontrolled Nav.St.

- Optimal control problem: Navier–Stokes, $Re = 20$
- Coarse mesh: 1404 DOF's in space, 5 timesteps, $\Delta t := 0.2$
  $\Rightarrow$ 8 424 DOF's, $\times 8$ per level

Numerical example: Flow-around-cylinder

Convergence of the Newton solver

		Simulation				Optimisation	
Δt	Space-Lv.	#NL	#MG	⌀#NL	⌀#MG	#NL	#MG
1/20	3	63	312	3	16	4	24
1/40	4	123	709	3	18	4	14
1/80	5	246	1589	3	20	4	13

- Nonlinear solver gained 5 digits
- Space-time MG gained 2 digits per step
- Space-preconditioner gained 2 digits per step

Numerical example: Flow-around-cylinder

Convergence of the Newton solver

Δt	Space-Lv.	Time sim.	Time opt.	Time opt / Time sim.
1/20	3	27	917	34.0
1/40	4	209	4346	20.7
1/80	5	2227	55174	24.8

Optimisation is $C \approx 20 - 30\times$ more expensive than simulation
$\rightarrow$ independent of the refinement level

$\rightarrow$ Aim: Factor 10-15 due to ”code optimisation”!

Numerical example: Flow-around-cylinder

Convergence of the Newton solver

Δt	Space-Lv.	Time sim.	Time opt.	Time opt / Time sim.
1/20	3	27	917	34.0
1/40	4	209	4346	20.7
1/80	5	2227	55174	24.8

Optimisation is $C \approx 20 - 30\times$ more expensive than simulation
$\rightarrow$ independent of the refinement level

$\rightarrow$ Aim: Factor 10-15 due to ”code optimisation”!

Numerical example: Flow-around-cylinder

Convergence of the solver for different α and γ

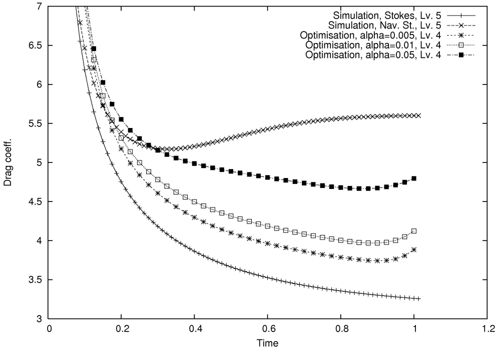
α	0.05		0.01		0.005	
γ	#NL	#MG	#NL	#MG	#NL	#MG
0	5	15	5	15	5	16
0.1	5	15	5	15	5	17
0.3	5	15	5	20	5	23
0.5	5	16	6	29	5	79

$\Delta t = 1/40$, Space-level 4

$\Rightarrow$ Smaller α + larger γ $\rightarrow$ worse convergence rates

Numerical example: Flow-around-cylinder

Solution quality: Analysis of the drag

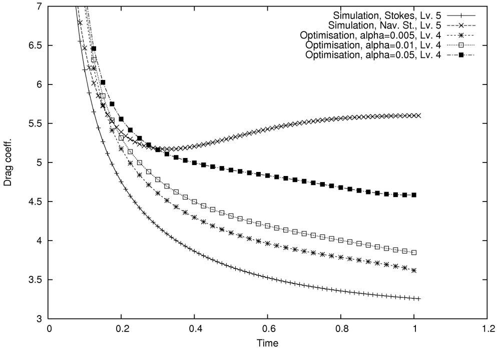


$\alpha = 0.05,\ 0.01,\ 0.005,$
$\gamma = 0.0$

$$C_D = \frac{2}{(\frac{2}{3}U_{\max})^2 \cdot 0.1} \int_\Gamma (\sigma n) n_x \, d\Gamma$$

Numerical example: Flow-around-cylinder

Solution quality: Analysis of the drag

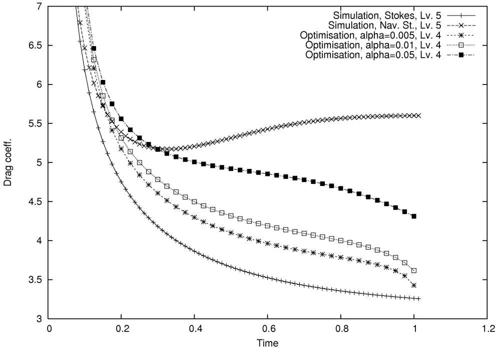


$\alpha = 0.05, 0.01, 0.005,$
$\gamma = 0.1$

$$C_D = \frac{2}{(\frac{2}{3}U_{max})^2 \cdot 0.1} \int_\Gamma (\sigma n) n_x d\Gamma$$

Numerical example: Flow-around-cylinder

Solution quality: Analysis of the drag



$\alpha = 0.05, 0.01, 0.005,$
$\gamma = 0.3$

$$C_D = \frac{2}{(\frac{2}{3}U_{\max})^2 \cdot 0.1} \int_\Gamma (\sigma n) n_x d\Gamma$$

Outlook

Further steps (till summer 2009):

- 2nd-order Crank-Nicolson scheme
- FEM-stabilisation (EO-FEM/interior penalty)
  $\rightarrow$ higher RE numbers
- improved smoothers for the space-time problem
- incorporate constraints

Outlook

Future plans for new period:

- Treatment of adapted meshes in space and time
- FEM & solver improvements
  $\rightarrow$ solver behaviour for α/γ, Q_2/P_1, Stabil., inexact Newton
- HPC / improved algorithms $\rightarrow$ necessary for 3D!
- boundary control
- complex CFD (non-Newtonian, non-isothermal fluids, FSI)

Space–time discretisation

Structure of $A(x)$:

$$
\left(\begin{array}{c|c|c}
\begin{array}{cc|cc}
\frac{M}{\Delta t}+N & & -B & \\
-M & \frac{M}{\Delta t}+N^* & & -B \\ \hline
-B^T & & 0 & \\
 & -B^T & & 0
\end{array} & -\frac{M}{\Delta t} & \\ \hline
-\frac{M}{\Delta t} &
\begin{array}{cc|cc}
\frac{M}{\Delta t}+N & \frac{1}{\alpha}M & -B & \\
-M & \frac{M}{\Delta t}+N^* & & -B \\ \hline
-B^T & & 0 & \\
 & -B^T & & 0
\end{array} & -\frac{M}{\Delta t} \\ \hline
 & -\frac{M}{\Delta t} &
\begin{array}{cc|cc}
\frac{M}{\Delta t}+N & \frac{1}{\alpha}M & -B & \\
-\frac{\gamma}{\Delta t}M & \frac{M}{\Delta t}+N^* & & -B \\ \hline
-B^T & & 0 & \\
 & -B^T & & 0
\end{array}
\end{array}\right)
$$

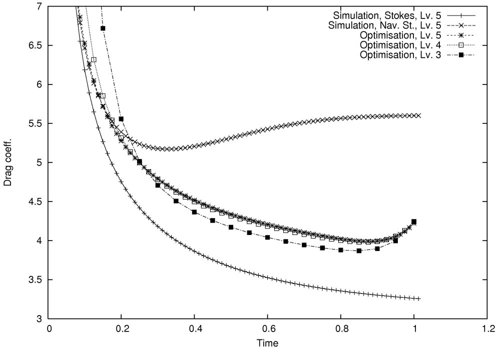
Space–time discretisation

Structure of $A'(x)$:

$$
\left[\begin{array}{c|c|c}
\begin{array}{cc|cc}
\frac{M}{\Delta t} + N^{**} & & -B & \\
-M + R_0 & \frac{M}{\Delta t} + N^{*} & & -B \\
\hline
-B^T & & 0 & \\
& -B^T & & 0
\end{array}
& -\frac{M}{\Delta t} & \\
\hline
-\frac{M}{\Delta t} &
\begin{array}{cc|cc}
\frac{M}{\Delta t} + N^{**} & \frac{1}{\alpha} M & -B & \\
-M + R_1 & \frac{M}{\Delta t} + N^{*} & & -B \\
\hline
-B^T & & 0 & \\
& -B^T & & 0
\end{array}
& -\frac{M}{\Delta t} \\
\hline
& -\frac{M}{\Delta t} &
\begin{array}{cc|cc}
\frac{M}{\Delta t} + N^{**} & \frac{1}{\alpha} M & -B & \\
-\frac{\gamma}{\Delta t} M + R_2 & \frac{M}{\Delta t} + N^{*} & & -B \\
\hline
-B^T & & 0 & \\
& -B^T & & 0
\end{array}
\end{array}\right]
$$

Numerical example: Flow-around-cylinder

Solution quality: Analysis of the drag

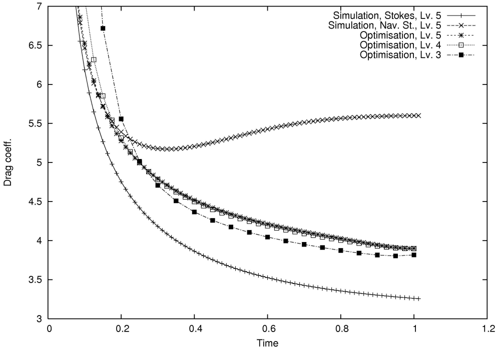


$\alpha = 0.01,$
$\gamma = 0.00$

$$C_D = \frac{2}{(\frac{2}{3}U_{\max})^2 \cdot 0.1} \int_\Gamma (\sigma n) n_x \, d\Gamma$$

Numerical example: Flow-around-cylinder

Solution quality: Analysis of the drag

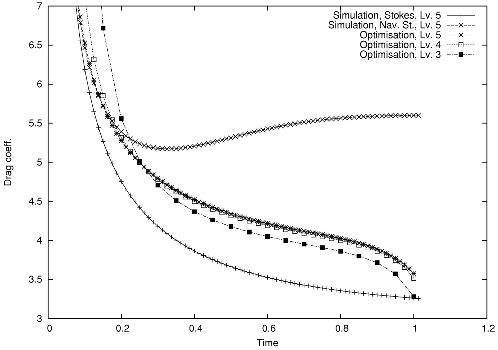


$\alpha = 0.01,$
$\gamma = 0.1$

$$C_D = \frac{2}{(\frac{2}{3}U_{\max})^2 \cdot 0.1} \int_\Gamma (\sigma n) n_x \, d\Gamma$$

Numerical example: Flow-around-cylinder

Solution quality: Analysis of the drag



$\alpha = 0.01$,
$\gamma = 0.5$

$$C_D = \frac{2}{(\frac{2}{3}U_{\max})^2 \cdot 0.1} \int_\Gamma (\sigma n) n_x d\Gamma$$

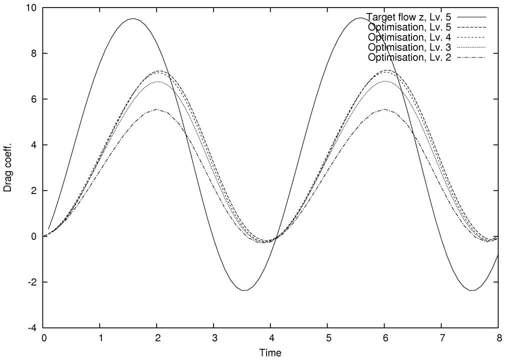
Numerical example:

Flow-around-cylinder with pulsating BC

- Domain: Flow-around-cylinder (like above), $T = 8$.
- Coarse mesh:
  1404 DOF's in space, 20 timesteps with $\Delta t = 0.4$.
- Target flow z: Parabolic inflow with max. velocity

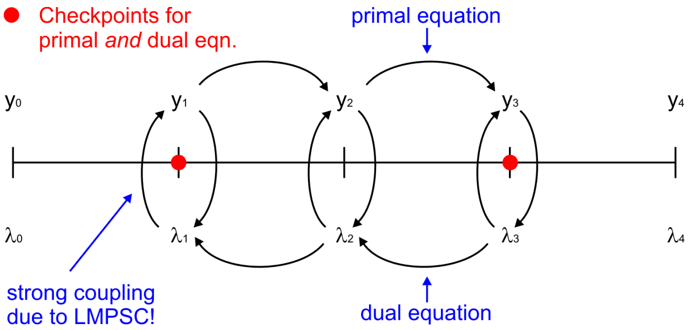
$$U_{\max}(t) := 0.15 \cdot \left(1 + \sin(\frac{t+3}{2}\pi)\right)/2$$

- Optimisation: Natural boundary conditions at inflow



$$C_D = \frac{2}{(\frac{2}{3}0.15)^2 \cdot 0.1} \int_\Gamma (\sigma n) n_x d\Gamma$$

Checkpointing in the One-shot approach



- Checkpoints $\rightarrow$ nonlinear subproblems of the same kind.
- High computational costs necessary for recomputation $\rightarrow$ due to strong coupling by LPSC!