A Numerical Study of Hierarchical Solution Concepts for Flow Control Problems

with M. Hinze, M. Köster, M. Razzaq (SPP1253)

Stefan Turek

Institute for Applied Mathematics
TU Dortmund

TU Dortmund, September 2014

Flow control model problem

Distributed Control for the nonstationary Navier-Stokes equations 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.

<u>Aim:</u> Solve with

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

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, ...

Ingredients:

- Newton + Space-time multigrid solvers
- Q_2/P_1^{disc}, IE + CN
- Distributed control, L_2+ $H^{\frac{1}{2}}$ boundary control
- Control constraints

Design of solution methods

General: Eliminate variables, apply Newton method

$$y_t - \nu\Delta y + \ldots = u$$
$$-\lambda_t - \nu\Delta\lambda + \ldots = y - z$$
$$\alpha u + \lambda = 0$$

Method 1: With $\lambda = \lambda(y(u))$, apply Newton solver to

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

Method 2: With $x = (y, \lambda, p, \xi)$, apply Newton solver to

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

Method 1: Newton approach in u

technische universität dortmund

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

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:

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

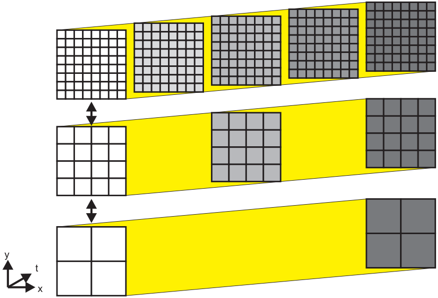
$$G'(x)\bar{x} := \begin{pmatrix} \bar{y}_t - \nu \Delta \bar{y} + ... + \frac{1}{\alpha}\bar{\lambda} \\ -\bar{\lambda}_t - \nu \Delta \bar{\lambda} + ... - \bar{y} \\ ... \end{pmatrix}$$

Multigrid solver methodology

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?)

KKT-system

Idea 1: Apply optimal control methods

- Lagrange approach on $Q = (0, T) \times \Omega \quad \Rightarrow \quad$ KKT-system

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

Idea 2: Exploit the ellipticity with MG methods

- Analysis $\Rightarrow$ elliptic charakter in space and time
  Problem equivalent to:

$$-y_{tt} + \Delta^2 y + \ldots = \ldots$$

- Monolithic Newton/MG in space + time

KKT-system

Idea 1: Apply optimal control methods

- Lagrange approach on $Q = (0, T) \times \Omega \quad \Rightarrow \quad$ KKT-system

$$\begin{aligned} y_t + C(y)y \;\; + \nabla p &= -\frac{1}{\alpha}\lambda && \text{in } Q, && y(0) = y_0 \\ -\lambda_t + N^*(y)\lambda + \nabla\xi &= (y - z) && \text{in } Q, && \lambda(T) = 0 \end{aligned}$$

Idea 2: Exploit the ellipticity with MG methods

- Analysis $\Rightarrow$ elliptic charakter in space and time
  Problem equivalent to:

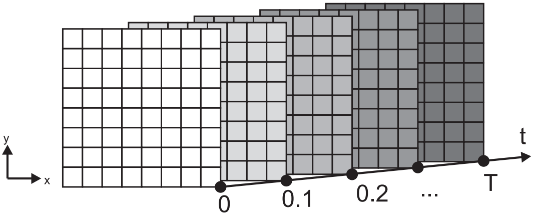
$$-y_{tt} + \Delta^2 y + \ldots = \ldots$$

- Monolithic Newton/MG in space + time

Space-time discretisation

Step 1: Space-time discretisation

- Unstructured mesh in space, $N \in \mathbb{N}$ timesteps



- Time discretisation: IE, CN, ..., timestep $k = 1/N$

$$
\begin{aligned}
(y_n - y_{n-1})/k + C(y_n)y_n \quad + \nabla p_n &= -\frac{1}{\alpha}\lambda_n \qquad \text{in } Q \\
(\lambda_n - \lambda_{n+1})/k + N^*(y_n)\lambda_n + \nabla \xi_n &= (y_n - z_n) \quad \text{in } Q
\end{aligned}
$$

- Space discretisation: FEM $(\tilde{Q}_1/Q_0,\ Q_2/P_1^{\text{disc}}, \ldots)$

Space-time system

Resultat: Nonlinear system in space and time

$$G(w)w = f$$

$$\left(\begin{array}{c|c|c|c|c} G_0 & \hat{M} & & & \\ \hline \check{M} & G_1 & \hat{M} & & \\ \hline & \check{M} & G_2 & \hat{M} & \\ \hline & & \ddots & \ddots & \ddots \\ \hline & & & \check{M} & G_N \end{array}\right) \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}$$

$\Rightarrow$ sparse block tridiagonal system

Space-time system

Resultat: Nonlinear system in space and time

$$G(w)w = f$$

$$\begin{pmatrix} G_0 & \hat{M} & & & \\ \check{M} & G_1 & \hat{M} & & \\ & \check{M} & G_2 & \hat{M} & \\ & & \ddots & \ddots & \ddots \\ & & & \check{M} & G_N \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}$$

$$w_0 \to \begin{pmatrix} y_0 \\ \lambda_0 \\ p_0 \\ \xi_0 \\ \hline y_1 \\ \lambda_1 \\ p_1 \\ \xi_1 \\ \hline \vdots \end{pmatrix}$$

$\Rightarrow$ sparse block tridiagonal system

Space-time system

Resultat: Nonlinear system in space and time

$$G(w)w = f$$

$$\begin{pmatrix} G_0 & \hat{M} & & & \\ \check{M} & G_1 & \hat{M} & & \\ & \check{M} & G_2 & \hat{M} & \\ & & \ddots & \ddots & \ddots \\ & & & \check{M} & G_N \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}$$

$$\check{M} = \begin{pmatrix} -\frac{I}{k} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$G_2 = \begin{pmatrix} \frac{I}{k} + C(y_2) & \frac{I}{\alpha} & \nabla & 0 \\ -I & \frac{I}{k} + N^*(y_2) & 0 & \nabla \\ \nabla\cdot & 0 & 0 & 0 \\ 0 & \nabla\cdot & 0 & 0 \end{pmatrix}$$

Space-time hierarchy and solver

Step 2: Space-time hierarchy

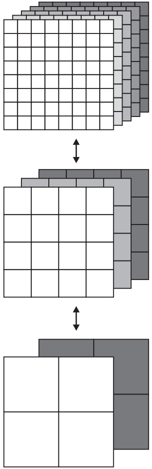
- Coarsening in space + time

Step 3: Space-time Newton-MG solver

- Newton solver on the fine mesh

$$w^{k+1} = w^k + G'(w^k)^{-1}(f - G(w^k)w^k)$$

- Space-time multigrid
  - To apply $G'(w^k)^{-1}$
  - Exploitation of the hierarchy
  - Needs smoother and prol./rest.



Space-time hierarchy and solver

Step 2: Space-time hierarchy

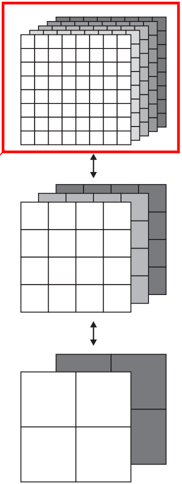
- Coarsening in space + time

Step 3: Space-time Newton-MG solver

- Newton solver on the fine mesh

$$w^{k+1} = w^k + G'(w^k)^{-1}(f - G(w^k)w^k)$$

- Space-time multigrid
  - To apply $G'(w^k)^{-1}$
  - Exploitation of the hierarchy
  - Needs smoother and prol./rest.



Space-time hierarchy and solver

Step 2: Space-time hierarchy

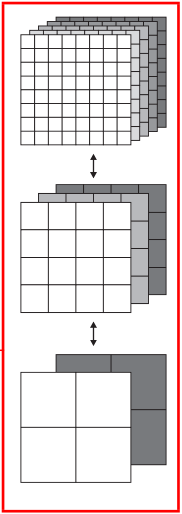
- Coarsening in space + time

Step 3: Space-time Newton-MG solver

- Newton solver on the fine mesh

$$w^{k+1} = w^k + G'(w^k)^{-1}(f - G(w^k)w^k)$$

- Space-time multigrid
  - To apply $G'(w^k)^{-1}$
  - Exploitation of the hierarchy
  - Needs smoother and prol./rest.



Construction of a smoother

- On every level: Linear subproblem

$$\begin{aligned} G'(w^k)x &= b \\ \Leftrightarrow \qquad Ax &= b \qquad \text{mit} \qquad A := G'(w^k) \end{aligned}$$

- Iterative smoother: Defect correction

$$x_{n+1} = x_n + P^{-1}(b - Ax_n)$$

- Typical preconditioner: Block methods

$$A = \left(\begin{array}{c|c|c|c|c} A_0 & \hat{M} & & & \\ \hline \check{M} & A_1 & \hat{M} & & \\ \hline & \check{M} & A_2 & \hat{M} & \\ \hline & & \ddots & \ddots & \ddots \\ \hline & & & \check{M} & A_N \end{array}\right)$$

Construction of a smoother

- On every level: Linear subproblem

$$G'(w^k)x = b$$
$$\Leftrightarrow \quad Ax = b \quad \text{mit} \quad A := G'(w^k)$$

- Iterative smoother: Defect correction

$$x_{n+1} = x_n + P^{-1}(b - Ax_n)$$

- Typical preconditioner: Block methods

$$P_0 = \left(\begin{array}{c|c|c|c|c} A_0 & & & & \\ \hline & A_1 & & & \\ \hline & & A_2 & & \\ \hline & & & \ddots & \\ \hline & & & & A_N \end{array}\right) \quad \Rightarrow \quad \text{Block-Jacobi}$$

Construction of a smoother

- On every level: Linear subproblem

$$G'(w^k)x = b$$

$$\Leftrightarrow \quad Ax = b \quad \text{mit} \quad A := G'(w^k)$$

- Iterative smoother: Defect correction

$$x_{n+1} = x_n + P^{-1}(b - Ax_n)$$

- Typical preconditioner: Block methods

$$P_1 = \begin{pmatrix} A_0 & & & & \\ \check{M} & A_1 & & & \\ & \check{M} & A_2 & & \\ & & \ddots & \ddots & \\ & & & \check{M} & A_N \end{pmatrix} \quad \Rightarrow \quad \text{Block-GS forward}$$

Construction of a smoother

- On every level: Linear subproblem

$$G'(w^k)x = b$$
$$\Leftrightarrow \qquad Ax = b \qquad \text{mit} \qquad A := G'(w^k)$$

- Iterative smoother: Defect correction

$$x_{n+1} = x_n + P^{-1}(b - Ax_n)$$

- Typical preconditioner: Block methods

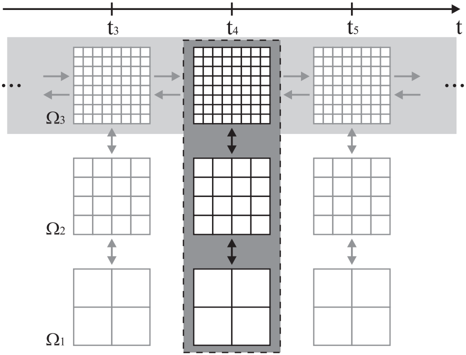
$$P_2 = \left(\begin{array}{c|c|c|c|c} A_0 & \hat{M} & & & \\ \hline & A_1 & \hat{M} & & \\ \hline & & A_2 & \hat{M} & \\ \hline & & & \ddots & \ddots \\ \hline & & & & A_N \end{array}\right) \quad \Rightarrow \quad \text{Block-GS backward}$$

Forward-Backward block smoother

- robust+efficient: Forward-/Backward strategy
- Diagonal blocks A_i: Equation in space, Oseen type
- every "timestep": A_i^{-1} = monolithic MG, LPSC smoother

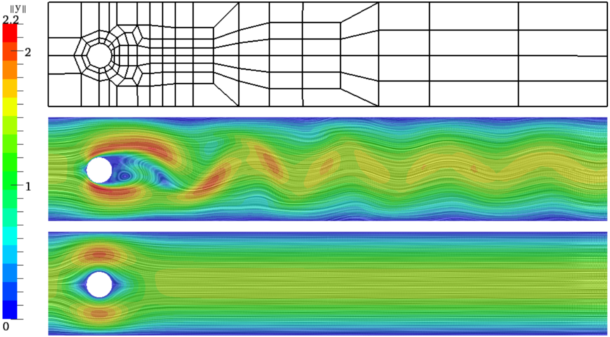
$$\begin{pmatrix} A_0 & & & \\ \check{M} & A_1 & & \\ & \check{M} & A_2 & \\ & & \ddots & \ddots \end{pmatrix}^{-1}$$

$$\begin{pmatrix} A_0 & \hat{M} & & \\ & A_1 & \hat{M} & \\ & & A_2 & \ddots \\ & & & \ddots \end{pmatrix}^{-1}$$



Numerical example

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



Mesh

Nav.St., periodic init. condition

Stokes, Target

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

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	$\frac{T_{\mathrm{opt}}}{T_{\mathrm{sim}}}$
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	$\frac{T_{\mathrm{opt}}}{T_{\mathrm{sim}}}$
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	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
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 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	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
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

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

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	$\frac{T_{\mathrm{opt}}}{T_{\mathrm{sim}}}$
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	$\frac{T_{\mathrm{opt}}}{T_{\mathrm{sim}}}$
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 → black-box applicable	high
$-F(u)$, $-G(x)$	simulation (nl.) → stopping criteria? robustness?	MatVec
apply $F'(u)$, $G'(x)$	simulation (lin.) → stopping criteria? robustness?	MatVec
preconditioner	∅ →not necessary?	expensive → inexact √ → paralleliseable √
Space-time MG	√	√

General results

- Newton-solver in (y, λ, p, ξ) usually more efficient than in u
  $\rightarrow$ more freedom w.r.t. stopping criteria
  $\rightarrow$ more freedom w.r.t. preconditioners

- 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

$\Rightarrow$ use SQP-type solvers if possible

Disadvantage in Method 1:

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

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

Possible alternative:(Analogous to CFD solvers)

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

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

$\Rightarrow$ 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 $x = (y, \lambda)$
$T_{\mathrm{opt}} / T_{\mathrm{sim}}$	≈ 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, but no preconditioning.

Similar: CG, GMRES,...

Linear system for method 2

The Newton solver in x reads:

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

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

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

Summary

Two solution methods analysed:

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

Main achivements:

- "Optimal" complexity
- $T_{\text{opt}} / T_{\text{sim}} \approx 20 - 50 \quad \rightarrow \quad$ for 'optimal' sim.
- Newton in (y, λ, p, ξ) usually more efficient than in $u \rightarrow$ due to inexact inner solvers + strong preconditioning

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?