Efficient distributed matrix-free multigrid methods for a stabilized solver for the incompressible Navier–Stokes equations on locally refined meshes

ENUMATH 2025: MS24 – Adaptivity in Space and Time

Peter Munch[†], Laura Prieto Saavedra[‡], Bruno Blais[‡]

[†]Institute of Mathematics, Technical University of Berlin, Germany
[‡]Polytechnique Montréal, Université de Montréal, Canada

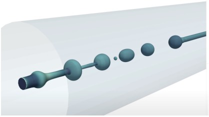
3. September, 2025

Part 1:

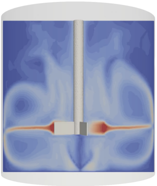
Motivation

Motivation: chemical and process engineering

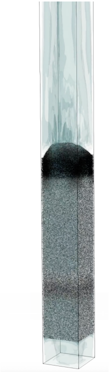
Multiphase



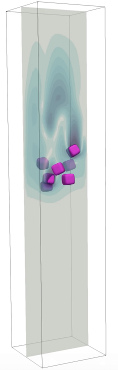
Mixing



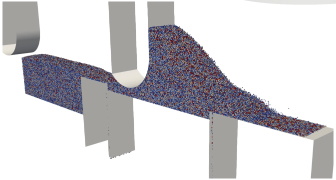
Fluidized beds



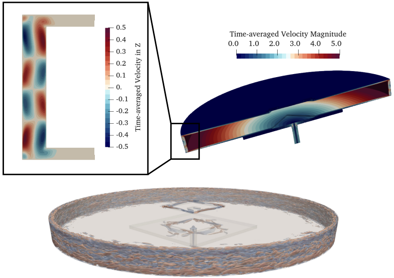
PR DNS



Granular flows



spinning disc reactor



Tools: CFD (single- and multiphase), transport equations, DEM, CFD/DEM, ...

Stabilized incompressible Navier–Stokes solver

Stabilized Navier–Stokes equations: find $\mathbf{u}$, p s.t. $\quad\rightarrow$ allow equal order elements ($Q_p Q_p$)

$$
\begin{aligned}
(\partial_t \mathbf{u},\ \mathbf{v}) + (\mathbf{u}\cdot\nabla\mathbf{u},\ \mathbf{v}) - (p,\ \nabla\cdot\mathbf{v}) + (\nu\varepsilon(\mathbf{u}),\ \varepsilon(\mathbf{v}))& \\
+ \sum_k \underbrace{\delta_1 \left(\partial_t\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \nu\Delta\mathbf{u},\ \mathbf{u}\cdot\nabla\mathbf{v}\right)_{\Omega_k}}_{\text{SUPG}} + \sum_k \underbrace{\delta_2 \left(\nabla\cdot\mathbf{u},\ \nabla\cdot\mathbf{v}\right)_{\Omega_k}}_{\text{GD}} &= 0, \\
(\nabla\cdot\mathbf{u},\ q) + \sum_k \underbrace{\delta_1 \left(\partial_t\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \nu\Delta\mathbf{u},\ \nabla q\right)_{\Omega_k}}_{\text{PSPG}} &= 0 \quad \forall \mathbf{v}, q
\end{aligned}
$$

- monolithic approach with BDF2
- Newton's method

Stabilized incompressible Navier–Stokes solver

Stabilized Navier–Stokes equations: find $\mathbf{u}$, p s.t. $\rightarrow$ allow equal order elements ($Q_p Q_p$)

$$
\begin{aligned}
&(\partial_t \mathbf{u},\ \mathbf{v}) + (\mathbf{u}\cdot\nabla\mathbf{u},\ \mathbf{v}) - (p,\ \nabla\cdot\mathbf{v}) + (\nu\varepsilon(\mathbf{u}),\ \varepsilon(\mathbf{v}))\\
&\quad + \sum_k \underbrace{\delta_1 \left(\partial_t \mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \nu\Delta\mathbf{u},\ \mathbf{u}\cdot\nabla\mathbf{v}\right)_{\Omega_k}}_{\text{SUPG}} + \sum_k \underbrace{\delta_2 \left(\nabla\cdot\mathbf{u},\ \nabla\cdot\mathbf{v}\right)_{\Omega_k}}_{\text{GD}} = 0,\\
&(\nabla\cdot\mathbf{u},\ q) + \sum_k \underbrace{\delta_1 \left(\partial_t \mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \nu\Delta\mathbf{u},\ \nabla q\right)_{\Omega_k}}_{\text{PSPG}} = 0 \quad \forall \mathbf{v}, q
\end{aligned}
$$

- monolithic approach with BDF2
- Newton's method
- software: Lethe based on deal.II

A. Alphonius, L. Barbeau, B. Blais, O. Gaboriault, O. Guévremont, J. Lamouche, P. Laurentin, O. Marquis, PM, V. Oliveira Ferreira, Papillon-Laroche, H., P.A. Patience, L. Prieto Saavedra, and M. Vaillant, 2025. Lethe 1.0: An Open-Source High-Performance and High-Order Computational Fluid Dynamics Software for Single and Multiphase Flows. SSRN.

Solution of linearized system

Resulting block structure of Jacobian:

- unstabilized:

$$J = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix}$$

- stabilized:

$$J' = \begin{bmatrix} A' & B' \\ C' & D \end{bmatrix}$$

Possible solution approaches (for unstabilized NS):

- block preconditioner based on block factorization, e.g., ... with $S = D - CA^{-1}B$

$$J \approx \begin{bmatrix} A & B \\ & S \end{bmatrix}$$

- monolithic preconditioner: ILU, AMG, (geometric) multigrid with Vanka smoothers, ...

E.C. Cyr, J.N. Shadid, R.S. Tuminaro ['12]:
- comparison of monolithic AMG and block preconditioner for stabilized FEM
- observation: AMG is scalable, block prec. based on block factorization not easy to construct

Solution of linearized system

Resulting block structure of Jacobian:

- unstabilized:
- stabilized:

$$J' = \begin{bmatrix} A' & B' \\ C' & D \end{bmatrix}$$

Scope of this presentation:

1. stabilized NS
2. monolithic geometric multigrid
3. locally refined meshes
4. using deal.II (and matrix-free evaluation)

Possibl

- bl , e.g., ... with $S = D - CA^{-1}B$
- monolithic preconditioner: ILU, AMG, (geometric) multigrid with Vanka smoothers, ...

E.C. Cyr, J.N. Shadid, R.S. Tuminaro ['12]:
- comparison of monolithic AMG and block preconditioner for stabilized FEM
- observation: AMG is scalable, block prec. based on block factorization not easy to construct

Part 2:

Multigrid in deal.II

Multigrid types in deal.II

Solve the system of linear equations $\mathcal{A}(\boldsymbol{x}) = \boldsymbol{b}$:

- presmoothing:

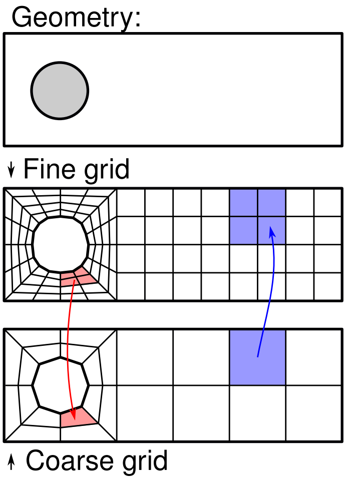
$$\boldsymbol{x} \leftarrow \mathcal{S}(\boldsymbol{x})$$

- recursive coarse-grid correction:

$$\mathcal{A}_c(\boldsymbol{v}) = \mathcal{R}(\boldsymbol{b} - \mathcal{A}(\boldsymbol{x})) \quad \text{and} \quad \boldsymbol{x} \leftarrow \boldsymbol{x} + \mathcal{P}(\boldsymbol{v})$$

- postsmoothing:

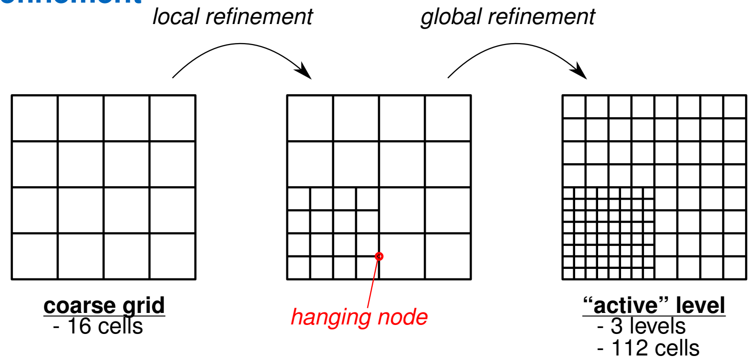
$$\boldsymbol{x} \leftarrow \mathcal{S}(\boldsymbol{x})$$



Definition of the levels gives the multigrid type.

Multigrid in deal.II: AMG via PETSc/Trilinos, $\underbrace{\text{geometric, polynomial, non-nested multigrid}}_{\text{native}}$.

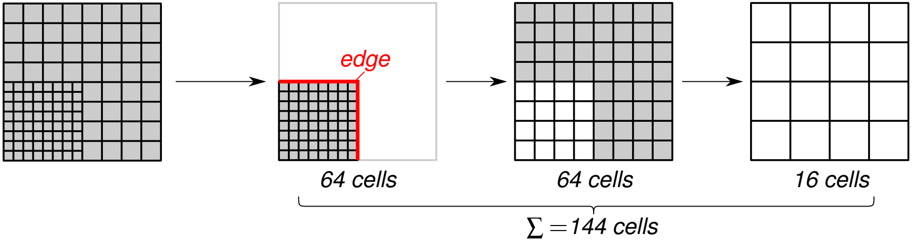
Local mesh refinement



- at hanging nodes: maintain H^1 regularity of the tentative solution (force solution representation of refined side to be matching polynomial representation of coarse side)
- apply (hanging-node) constraints via $x_i = \sum_j c_{ij} x_j + b_j$ (constraint matrix)
- hanging nodes are “motivation” for development of different geometric multigrid variants
- other variants: *p*- and *hp*-adaptivity (▷ Marc Fehling, Th 12:15—12:40)

Shephard, M.S., 1984. Linear multipoint constraints applied via transformation as part of a direct stiffness assembly process. IJNME.

Local mesh refinement: local smoothing (LS)



- internal interface/“edge”: (in)homogeneous DBC during pre-/postsmoothing
- uses refinement levels + first-child policy $\rightarrow$ memory-efficient, efficient transfer
- smoothers designed for uniform meshes, e.g., patch smoothers, are applicable

Brandt, A. 1977. Multi-level adaptive solutions to boundary-value problems. Mathematics of Computation.

Local mesh refinement: local smoothing (LS, cont.)

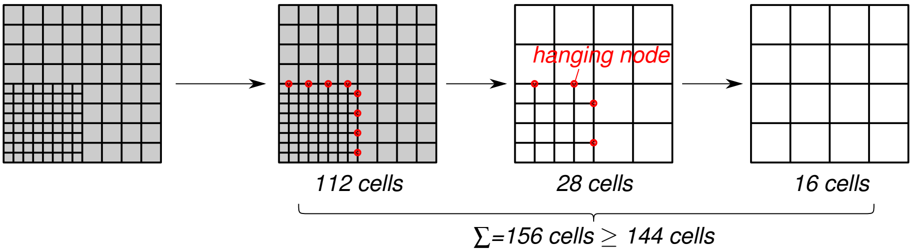
Janssen, B. and Kanschat, G., 2011. Adaptive multilevel methods with local smoothing for H^1-and H^{curl}-conforming high order finite element methods. SIAM JSC.

Kronbichler, M. and Wall, W.A., 2018. A performance comparison of continuous and discontinuous Galerkin methods with fast multigrid solvers. SISC.

Kronbichler, M. and Ljungkvist, K., 2019. Multigrid for matrix-free high-order finite element computations on graphics processors. ACM TOPC.

Clevenger, T.C. et. al., 2021. A flexible, parallel, adaptive geometric multigrid method for FEM. ACM TOMS.

Local mesh refinement: global coarsening (GC)



- repartitioning of levels is common $\rightarrow$ good load balance but potentially expensive transfer
- smoothing: hanging-node constraints need to be considered; more cells

Becker, R. and Braack, M., 2000. Multigrid techniques for finite elements on locally refined meshes. Numerical linear algebra with applications.

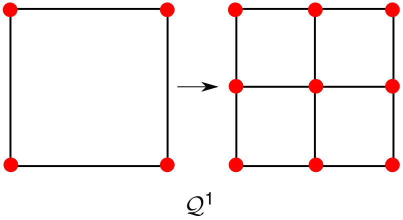
Becker, R., Braack, M. and Richter, T., 2007. Parallel multigrid on locally refined meshes. RFDT.

Local mesh refinement (cont.)

PM, T. Heister, L. Prieto Saavedra, and M. Kronbichler, “Efficient distributed matrix-free multigrid methods on locally refined meshes for FEM computations”, ACM TOPC, 2022.

- presents a unified multigrid framework with focus on (matrix-free) transfer operator

$$\boldsymbol{x}^{(f)} = \mathcal{P}^{(f,c)} \circ \boldsymbol{x}^{(c)} \quad \leftrightarrow \quad \boxed{\boldsymbol{x}^{(f)} = \sum_{e \in \{cells\}} \mathcal{S}_e^{(f)} \circ \mathcal{W}_e^{(f)} \circ \mathcal{P}_e^{(f,c)} \circ \mathcal{C}_e^{(c)} \circ \mathcal{G}_e^{(c)} \circ \boldsymbol{x}^{(c)}}$$



with:

$\mathcal{C}_e^{(c)} \circ \mathcal{G}_e^{(c)}$: gather values and apply constraints

$\mathcal{P}_e^{(f,c)}$: prolongate on coarse cell (see figure)

$\mathcal{W}_e^{(f)}$: consider valence

$\mathcal{S}_e^{(f)}$: add into $\boldsymbol{x}^{(f)}$ (coarse-side identification $\rightarrow$ communication)

PM, K. Ljungkvist, and M. Kronbichler, M., 2022. Efficient application of hanging-node constraints for matrix-free high-order FEM computations on CPU and GPU. ISC High Performance.

Local mesh refinement (cont.)

PM, T. Heister, L. Prieto Saavedra, and M. Kronbichler, "Efficient distributed matrix-free multigrid methods on locally refined meshes for FEM computations", ACM TOPC, 2022.

- presents a unified multigrid framework with focus on (matrix-free) transfer operator

Research question:

1. Which is faster: GS or LS?
3. How many iterations do GS/LS need?
3. What is the bottleneck: smoothing vs. transfer?
4. Serial vs. parallel!?

$\boldsymbol{x}^{(} \quad {}^{(c)}_{e} \circ \mathcal{G}_e^{(c)} \circ \boldsymbol{x}^{(c)}$

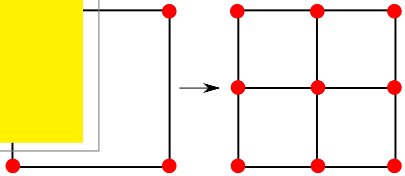
with:

$\mathcal{C}_e^{(c)} \circ$

$\mathcal{P}_e^{(f,c)}$: prolongate on coarse cell (see figure)

$\mathcal{W}_e^{(f)}$: consider valence

$\mathcal{S}_e^{(f)}$: add into $\boldsymbol{x}^{(f)}$ (coarse-side identification $\rightarrow$ communication)

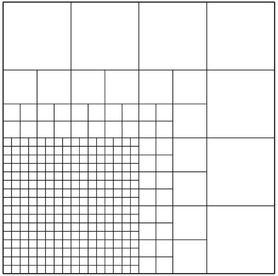


$\mathcal{Q}^1$

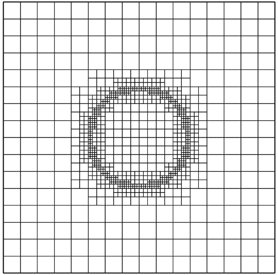
PM, K. Ljungkvist, and M. Kronbichler, M., 2022. Efficient application of hanging-node constraints for matrix-free high-order FEM computations on CPU and GPU. ISC High Performance.

Local mesh refinement (cont.)

Solve 3D Poisson problem with constant right-hand-side function and DBC on:



octant $L = 5$



sphere $L = 7$

configuration:

- PCG with 1 V-cycle GMG
- relative tolerance: 10^{-4}
- $p = 1$ and $p = 4$
- mixed precision (double, MG: float)
- coarse-grid solver: AMG via ML
- weight $2\times$ of cell with hanging nodes
- smoother: Chebyshev iteration (degree 3) around a point-Jacobi method

Local mesh refinement (cont.)

introduce geometric metrics (serial workload W_s, parallel workload W_p, parallel workload efficiency η_w, vertical communication efficiency η_v)

	1 process		192 processes					
	LS	GC	LS			GC		
	W_s	W_s	W_p	η_w	η_v	W_p	η_w	η_v
octant ($L = 9$)	1.9e+7	1.9e+7	1.6e+5	64%	99%	1.0e+5	98%	38%
sphere ($L = 9$)	2.5e+6	2.5e+6	3.5e+4	36%	99%	1.5e+4	93%	84%

and relate these to performance differences (Poisson problem; point Jacobi)

	1 process				192 processes			
	LS		GC		LS		GC	
	$\#i$	$t[s]$	$\#i$	$t[s]$	$\#i$	$t[s]$	$\#i$	$t[s]$
octant ($L = 9$, $p = 1$)	4	2.2e+1	3	1.8e+1	4	2.3e-1	3	1.3e-1
sphere ($L = 9$, $p = 1$)	5	3.7e+0	4	4.3e+0	5	6.3e-2	4	3.5e-2

Local mesh refinement (cont.)

introduce geometric metrics (serial workload W_s, parallel workload W_p, parallel workload efficiency η_w, vertical communication efficiency η_v)

	1 process		192 processes					
	LS	GC	LS			GC		
	W_s	W_s	W_p	η_w	η_v	W_p	η_w	η_v
octant ($L = 9$)	1.9e+7	1.9e+7	1.6e+5	64%	99%	1.0e+5	98%	38%
sphere ($L = 9$)	2.5e+6	2.5e+6	3.5e+4	36%	99%	1.5e+4	93%	84%

and relate these to performance differences (Poisson problem; point Jacobi)

	1 process				192 processes			
	LS		GC		LS		GC	
	$\#i$	$t[s]$	$\#i$	$t[s]$	$\#i$	$t[s]$	$\#i$	$t[s]$
octant ($L = 9$, $p = 1$)	4	2.2e+1	3	1.8e+1	4	2.3e-1	3	1.3e-1
sphere ($L = 9$, $p = 1$)	5	3.7e+0	4	4.3e+0	5	6.3e-2	4	3.5e-2

Local mesh refinement (cont.)

introduce geometric metrics (serial workload W_s, parallel workload W_p, parallel workload efficiency η_w, vertical communication efficiency η_v)

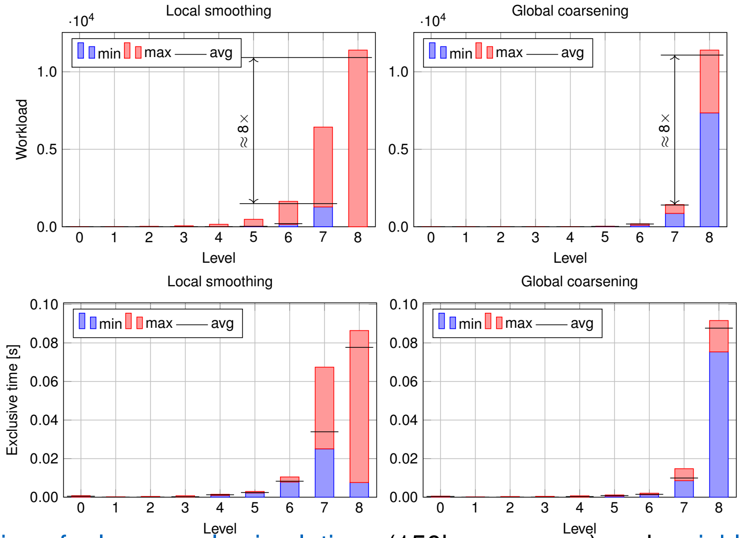
	1 process		192 processes					
	LS	GC	LS			GC		
	W_s	W_s	W_p	η_w	η_v	W_p	η_w	η_v
octant ($L = 9$)	1.9e+7	1.9e+7	1.6e+5	64%	99%	1.0e+5	98%	38%
sphere ($L = 9$)	2.5e+6	2.5e+6	3.5e+4	36%	99%	1.5e+4	93%	84%

and relate these to performance differences (Poisson problem; point Jacobi)

	1 process				192 processes			
	LS		GC		LS		GC	
	$\#i$	$t[s]$	$\#i$	$t[s]$	$\#i$	$t[s]$	$\#i$	$t[s]$
octant ($L = 9$, $p = 1$)	4	2.2e+1	3	1.8e+1	4	2.3e-1	3	1.3e-1
sphere ($L = 9$, $p = 1$)	5	3.7e+0	4	4.3e+0	5	6.3e-2	4	3.5e-2

Local mesh refinement (cont.)

Workload and execution time per level, e.g., for octant ($L = 8$):



Similar observations for large-scale simulations (150k processes) and variable viscosity problems (block preconditioner) $\Rightarrow$ What about stabilized NS?

Part 3:

Multigrid for the stabilized Navier–Stokes equations

Solution strategy

Solve stabilized Navier–Stokes equations with a monolithic approach and the Jacobian of the form

$$J = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

Monolithic *h*-multigrid as preconditioner for GMRES:

- Smoother:
  relaxation + inverse diagonal ($\times$ 5) or additive Schwarz method (element-centric, $\times$ 2)
- coarse-grid solver:
  - AMG, ILU, direct solver + GMRES (optional)
  - *p*-multigrid for higher-order elements
- locally refined meshes: local smoothing, global coarsening

L. Prieto Saavedra, PM, B. Blais, 2025, "A Matrix-Free Stabilized Solver for the Incompressible Navier-Stokes Equations", JCP.

Solution strategy

Solve stabilized Navier–Stokes equations with a monolithic approach and the Jacobian of the form

$$J = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

Estimates of comp. costs:

1. stabilized NS operator $\approx 2\times$ more expensive than vector Laplace operator
2. ASM $>100\times$ more expensive than point diagonal

$\Rightarrow$ smoother dominates costs of a V-cycle

Monolithic *h*-multigrid as precon

- Smoother:
  relaxation + inverse diagon
- coarse-grid solver:
  - AMG, ILU, direct solver + GMRES (optional)
  - *p*-multigrid for higher-order elements
- locally refined meshes: local smoothing, global coarsening

L. Prieto Saavedra, PM, B. Blais, 2025, "A Matrix-Free Stabilized Solver for the Incompressible Navier-Stokes Equations", JCP.

Solution strategy

Solve stabilized Navier–Stokes equations with a monolithic approach and the Jacobian of the form

$$J = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

Monolithic *h*-multigrid as preconditioner for GMRES:

Benchmarks:

1. MMS (globally refined meshes, steady state)
2. Taylor–Couette flow (locally refined meshes, transient)
3. flow past a sphere (locally refined meshes, steady state)

Hardware:

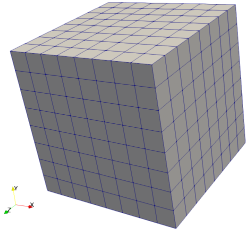
- Niagara distributed memory cluster of the Digital Research Alliance of Canada; CPU: 2 sockets with 20 Intel Skylake cores (2.4GHz, AVX512)
- workstation: 24 cores, Intel(R) Core(TM) i9-14900

Globally refined meshes: results for steady-state simulations

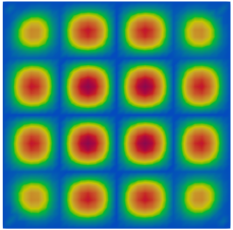
MMS: We define $\vec{u}$ and p as follows:

$$\vec{u} = \begin{pmatrix} \sin^2(\pi x)\cos(\pi y)\sin(\pi y)\cos(\pi z)\sin(\pi z) \\ \cos(\pi x)\sin(\pi x)\sin^2(\pi y)\cos(\pi z)\sin(\pi z) \\ -2\cos(\pi x)\sin(\pi x)\cos(\pi y)\sin(\pi y)\sin^2(\pi z) \end{pmatrix}$$

$$p = \sin(\pi x)\sin(\pi y)\sin(\pi z)$$



insert them into the Navier-Stokes equations and find the appropriate source term $\vec{f}$.



- steady-state
- domain: $\Omega = (-1,1)^3$
- zero Dirichlet boundary conditions
- Q_pQ_p elements with $p = 1,2,3$
- abs (n)/abs(l)/rel(l) tolerance: $10^{-8/-10/-4}$

Globally refined meshes: results for steady-state simulations (cont.)

Number of iterations: $-$: not converged; $*$: out of memory

	Q_1Q_1				Q_2Q_2				Q_3Q_3			
	ID		**ASM**		**ID**		**ASM**		**ID**		**ASM**	
ℓ	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$
4	3	7.0	3	4.3	3	7.0	3	4.7	3	7.7	3	5.7
5	–	–	3	5.0	3	7.7	3	5.0	3	8.3	3	6.0
6	–	–	3	5.3	3	7.7	3	5.3	3	9.0	3	6.7
7	–	–	3	6.0	3	8.7	3	5.3	3	9.3	3	6.7
8	–	–	3	6.0	2	10.5	3	5.7	2	12.0	$\star$	$\star$
9	–	–	2	7.5								

Reference (AMG):

Q_1Q_1			Q_2Q_2			Q_3Q_3		
ℓ	N_N	$\overline{N}_L$	ℓ	N_N	$\overline{N}_L$	ℓ	N_N	$\overline{N}_L$
6	3	13.8 ± 5.2	7	3	12.8 ± 0.2	8	3	13.5 ± 1.2

Globally refined meshes: results for steady-state simulations (cont.)

Number of iterations: $-$: not converged; $*$: out of memory

	$Q_1 Q_1$				$Q_2 Q_2$				$Q_3 Q_3$			
	ID		**ASM**		**ID**		**ASM**		**ID**		**ASM**	
ℓ	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$
4	3	7.0	3	4.3	3	7.0	3	4.7	3	7.7	3	5.7
5	–	–	3	5.0	3	7.7	3	5.0	3	8.3	3	6.0
6	–	–	3	5.3	3	7.7	3	5.3	3	9.0	3	6.7
7	–	–	3	6.0	3	8.7	3	5.3	3	9.3	3	6.7
8	–	–	3	6.0	2	10.5	3	5.7	2	12.0	$\star$	$\star$
9	–	–	2	7.5								

Reference (AMG):

$Q_1 Q_1$			$Q_2 Q_2$			$Q_3 Q_3$		
ℓ	N_N	$\overline{N}_L$	ℓ	N_N	$\overline{N}_L$	ℓ	N_N	$\overline{N}_L$
6	3	13.8 ± 5.2	7	3	12.8 ± 0.2	8	3	13.5 ± 1.2

Globally refined meshes: results for steady-state simulations (cont.)

Number of iterations: $-$: not converged; $*$: out of memory

	$Q_1 Q_1$				$Q_2 Q_2$				$Q_3 Q_3$			
	ID		**ASM**		**ID**		**ASM**		**ID**		**ASM**	
ℓ	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$
4	3	7.0	3	4.3	3	7.0	3	4.7	3	7.7	3	5.7
5	–	–	3	5.0	3	7.7	3	5.0	3	8.3	3	6.0
6	–	–	3	5.3	3	7.7	3	5.3	3	9.0	3	6.7
7	–	–	3	6.0	3	8.7	3	5.3	3	9.3	3	6.7
8	–	–	3	6.0	2	10.5	3	5.7	2	12.0	$\star$	$\star$
9	–	–	2	7.5								

Reference (AMG):

$Q_1 Q_1$			$Q_2 Q_2$			$Q_3 Q_3$		
ℓ	N_N	$\overline{N}_L$	ℓ	N_N	$\overline{N}_L$	ℓ	N_N	$\overline{N}_L$
6	3	13.8 ± 5.2	7	3	12.8 ± 0.2	8	3	13.5 ± 1.2

Globally refined meshes: results for steady-state simulations (cont.)

Number of iterations: $-$: not converged; $*$: out of memory

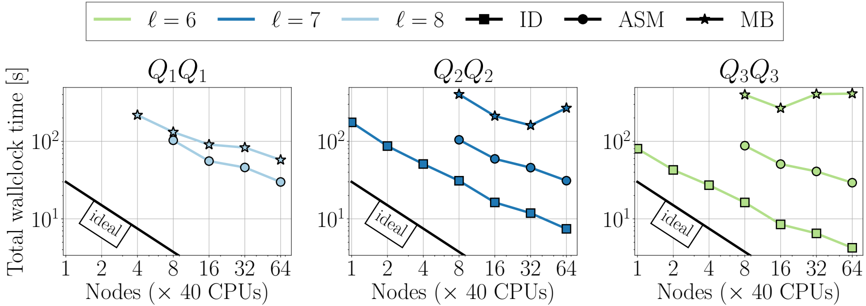
	$Q_1 Q_1$				$Q_2 Q_2$				$Q_3 Q_3$			
	ID		**ASM**		**ID**		**ASM**		**ID**		**ASM**	
ℓ	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$	N_N	$\overline{N}_L$
4	3	7.0	3	4.3	3	7.0	3	4.7	3	7.7	3	5.7
5	–	–	3	5.0	3	7.7	3	5.0	3	8.3	3	6.0
6	–	–	3	5.3	3	7.7	3	5.3	3	9.0	3	6.7
7	–	–	3	6.0	3	8.7	3	5.3	3	9.3	3	6.7
8	–	–	3	6.0	2	10.5	3	5.7	2	12.0	$\star$	$\star$
9	–	–	2	7.5								

Reference (AMG):

$Q_1 Q_1$			$Q_2 Q_2$			$Q_3 Q_3$		
ℓ	N_N	$\overline{N}_L$	ℓ	N_N	$\overline{N}_L$	ℓ	N_N	$\overline{N}_L$
6	3	13.8 ± 5.2	7	3	12.8 ± 0.2	8	3	13.5 ± 1.2

Globally refined meshes: results for steady-state simulations (cont.)

Strong-scalability study on Niagara:



Globally refined meshes: results for steady-state simulations (cont.)

Observations:

- geometric multigrid solver is faster than matrix-based AMG
- inverse diagonal works in many cases but not for all (e.g., $Q_1 Q_1$)
- additive Schwarz method is more robust
- element-centric patches are enough & vertex-star patches are more expensive (reason: non-zero D block?)

Comments on static **flow past a sphere** (1024 coarse-grid cells):

- similar observations
- for inverse diagonal, the quality of the coarse-grid solver is important

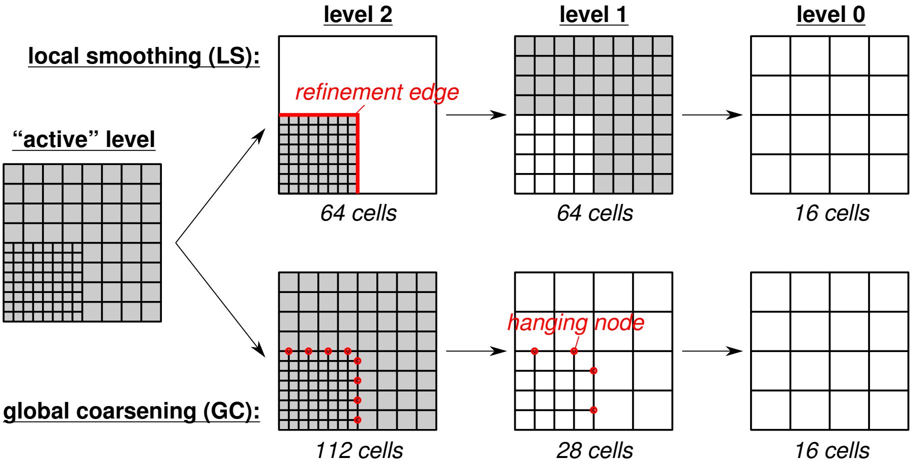
Comments on **transient simulations**, e.g., Taylor–Green vortex:

- inverse diagonal is enough

More details:

L. Prieto Saavedra, PM, B. Blais, 2025, “A Matrix-Free Stabilized Solver for the Incompressible Navier-Stokes Equations”, JCP.

Locally refined meshes



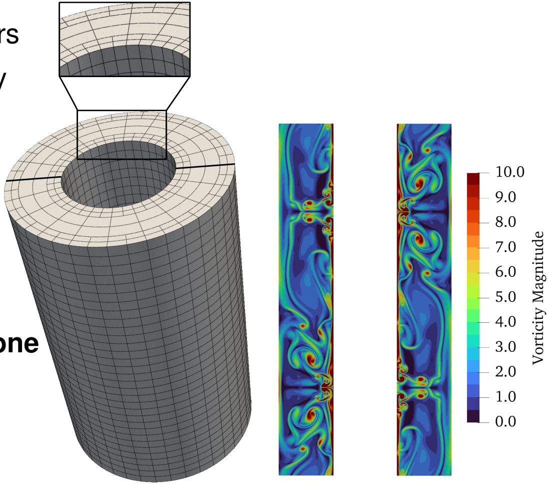
Local smoothing:

- Dirichlet BC both for $\boldsymbol{u}$ and p at refinement edges!?

Locally refined meshes: Taylor–Couette flow

Complex turbulent flow problem:

- annular flow between two coaxial cylinders
- inner cylinder with a fixed angular velocity
- outer cylinder is static
- curved walls
- transient: BDF2, fixed CFL=1
- $\mathrm{Re} = 4000$
- $Q_p Q_p$ elements with $p = 1, 2$
- **static: global mesh refinement ℓ with one additional refinement** next to the walls
- simulation time: 60s
- abs (n)/abs(l)/rel(l) tolerance: $10^{-5/-7/-4}$



L. Prieto Saavedra, J. Archambault, PM, B. Blais, 2025, “An implicit large-eddy simulation study of the turbulent Taylor-Couette flow with an inner rotating cylinder”, Journal of Turbulence.

Locally refined meshes: Taylor–Couette flow (cont.)

Number of iterations and time T (with 12 cores on workstation):

	$Q_1 Q_1$					
			GC		LS	
l	N_T	$\overline{N}_N$	$\overline{N}_L$	T	$\overline{N}_L$	T
2	72	2	4.71	6.47	7.02	7.97
3	152	2	4.01	84.5	5.03	84.5

	$Q_2 Q_2$					
			GC		LS	
l	N_T	$\overline{N}_N$	$\overline{N}_L$	T	$\overline{N}_L$	T
2	149	2	4.49	79.8	5.36	88.2
3	308	1.95	3.73	745	3.87	753

Locally refined meshes: Taylor–Couette flow (cont.)

Number of iterations and time T (with 12 cores on workstation):

	$Q_1 Q_1$						$Q_2 Q_2$					
			GC		LS				GC		LS	
l	N_T	$\overline{N}_N$	$\overline{N}_L$	T	$\overline{N}_L$	T	l	N_T	$\overline{N}_N$	$\overline{N}_L$	T	$\overline{N}_L$	T
2	72	2	4.71	6.47	7.02	7.97	2	149	2	4.49	79.8	5.36	88.2
3	152	2	4.01	84.5	5.03	84.5	3	308	1.95	3.73	745	3.87	753

Locally refined meshes: Taylor–Couette flow (cont.)

Number of iterations and time T (with 12 cores on workstation):

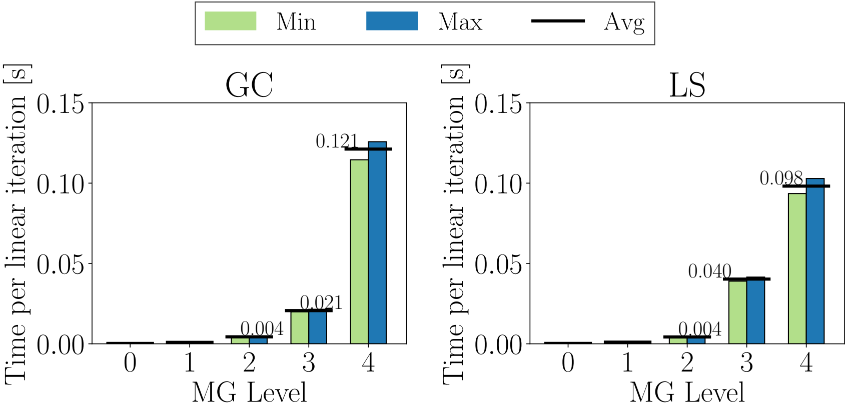
	$Q_1 Q_1$						
			GC		LS		
l	N_T	$\overline{N}_N$	$\overline{N}_L$	T	$\overline{N}_L$	T
2	72	2	4.71	6.47	7.02	7.97
3	152	2	4.01	84.5	5.03	84.5

	$Q_2 Q_2$					
			GC		LS	
l	N_T	$\overline{N}_N$	$\overline{N}_L$	T	$\overline{N}_L$	T
2	149	2	4.49	79.8	5.36	88.2
3	308	1.95	3.73	745	3.87	753

Locally refined meshes: Taylor–Couette flow (cont.)

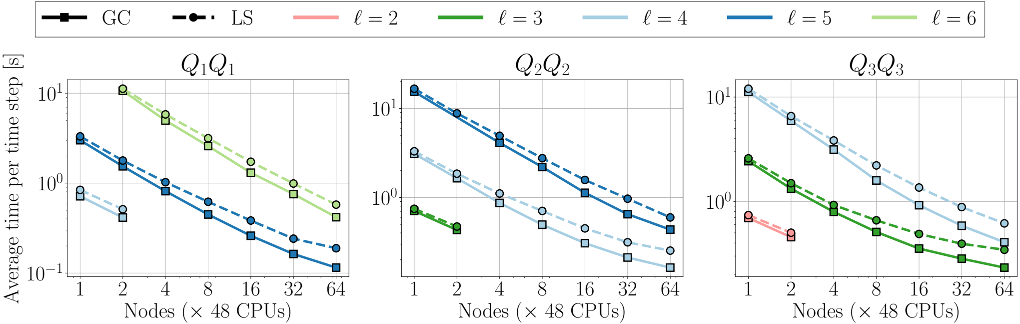
	GC	LS
η_w	99.7%	97.9%
η_v	98.9%	99.7%

Time of each multigrid level ($l = 3$, $Q_2 Q_2$):



Locally refined meshes: Taylor–Couette flow (cont.)

Strong-scaling study on Niagara:

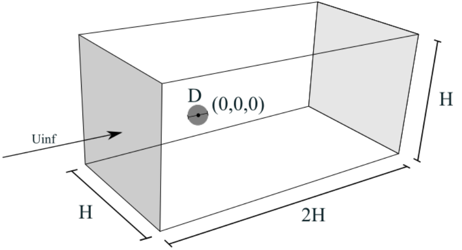


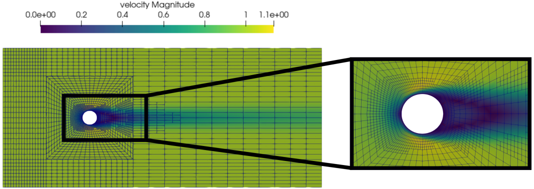
Observation: with increasing number of processes, the workload imbalance increases.

Locally refined meshes: flow past a cylinder

Challenging external flow problem:

- fixed entrance velocity $U_\infty = 1$
- steady-state, $\mathrm{Re} = 150$. Initial condition: ramp up Re starting with $\mathrm{Re} = 10$
- no-slip boundary conditions around the sphere and slip boundary conditions on the wall
- $Q_p Q_p$ elements with $p = 1, 2$
- initial global refinement ℓ
- **dynamic mesh refinement:** using Kelly error estimator on the pressure
- abs (n)/abs(l)/rel(l) tolerance: $10^{-5/-7/-4}$





Locally refined meshes: flow past a cylinder (cont.)

Number of iterations and time T (with 12 cores on workstation; ramp up ignored):

	$Q_1 Q_1$						$Q_2 Q_2$				
		GC		LS				GC		LS	
l	N_N	$\overline{N}_L$	T	$\overline{N}_L$	T	l	N_N	$\overline{N}_L$	T	$\overline{N}_L$	T
1	5	14.6	7.36	15.2	7.14	1	4	14.75	28.2	14.75	34.1
2	4	23.25	37.4	23	43.7	2	2	23	162	23.5	206

Locally refined meshes: flow past a cylinder (cont.)

Number of iterations and time T (with 12 cores on workstation; ramp up ignored):

	$Q_1 Q_1$						$Q_2 Q_2$				
		GC		LS				GC		LS	
l	N_N	$\overline{N}_L$	T	$\overline{N}_L$	T	l	N_N	$\overline{N}_L$	T	$\overline{N}_L$	T
1	5	14.6	7.36	15.2	7.14	1	4	14.75	28.2	14.75	34.1
2	4	23.25	37.4	23	43.7	2	2	23	162	23.5	206

Locally refined meshes: flow past a cylinder (cont.)

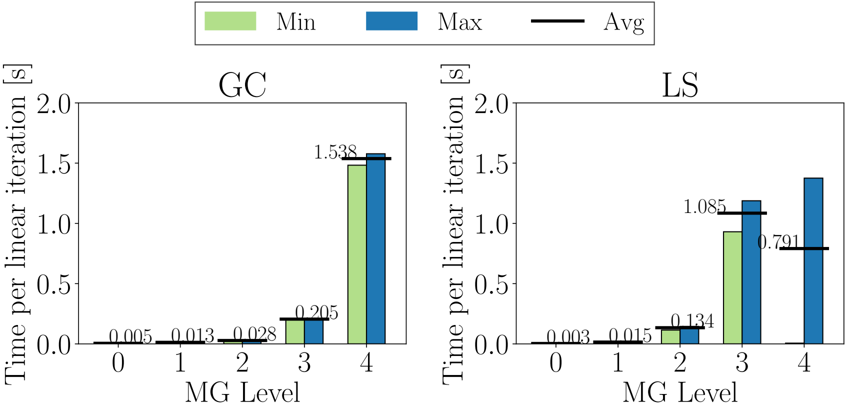
Number of iterations and time T (with 12 cores on workstation; ramp up ignored):

	$Q_1 Q_1$						$Q_2 Q_2$				
		GC		LS				GC		LS	
l	N_N	$\overline{N}_L$	T	$\overline{N}_L$	T	l	N_N	$\overline{N}_L$	T	$\overline{N}_L$	T
1	5	14.6	7.36	15.2	7.14	1	4	14.75	28.2	14.75	34.1
2	4	23.25	37.4	23	43.7	2	2	23	162	23.5	206

Locally refined meshes: flow past a cylinder (cont.)

	GC	LS
η_w	99.9%	53.8%
η_v	87.1%	99.9%

Time of each multigrid level ($l = 2$, $Q_2 Q_2$):



Part 4:

Conclusions & Outlook

Conclusions & Outlook

Conclusions:

- matrix-free stabilized Navier–Stokes solver
- freely available: deal.II + Lethe CFD
- monolithic solution approach $\rightarrow$ monolithic geometric multigrid
- transient simulations only require the diagonal for preconditioning
- steady state needs stronger smoothers, e.g., additive Schwarz smoother
- on locally refined meshes: both global coarsening and local smoothing is working

Outlook:

- improve efficiency of additive Schwarz smoother
- investigate influence of BC in local smoothing

Conclusions & Outlook (cont.)

Observation: pure Dirichlet BC during local smoothing seems not to have a negative influence.

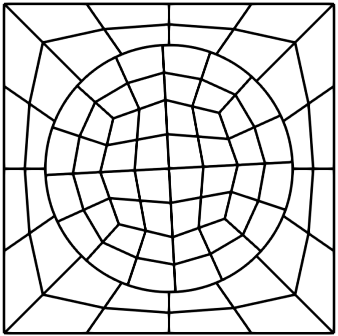
Other fields:

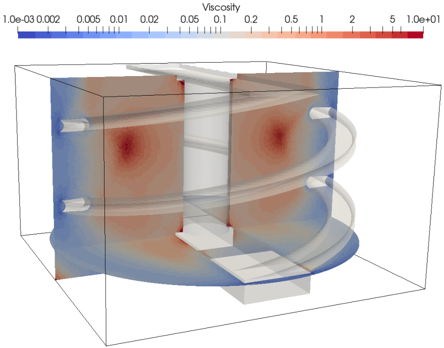
- model refinement methods, e.g., Tominec, I., Ahlkrona, J. and Braack, M., 2025. Well-posedness of the Stokes problem under modified pressure Dirichlet boundary conditions. BIT Numerical Mathematics.
- domain decomposition, e.g., Cai, M. and Pavarino, L.F., 2016. Hybrid and multiplicative overlapping Schwarz algorithms with standard coarse spaces for mixed linear elasticity and Stokes problems. Communications in Computational Physics.

Conclusions & Outlook (cont.)

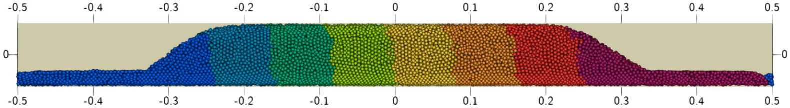
Goal: to develop a fast and efficient multi-physics solver for process engineering

- mortar FEM





- volume-averaged Navier—Stokes ⇒ unresolved CFD-DEM



- non-Newtonian fluid
- reactive flows