Speeding up an unsteady flow simulation by adaptive BDDC and Krylov subspace recycling

Jakub Šístek[1]
joint work with
Martin Hanek[1,2] and Jan Papež[1]

[1]Institute of Mathematics of the Czech Academy of Sciences, Prague
[2]Czech Technical University in Prague

HPCSE 2026
High Performance Computing in Science and Engineering

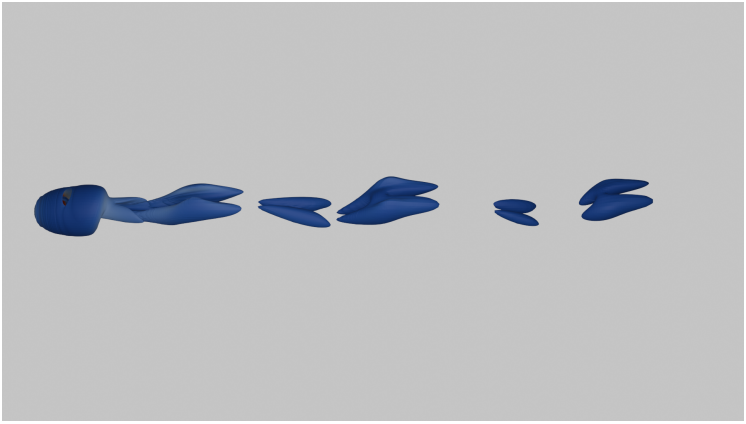
Institute of Mathematics
Czech Academy of Sciences

CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

High Performance Computing in Science and Engineering (HPCSE) 2026
Hotel Soláň, Czech Republic
May 18–21, 2026

Aim of the work

- Speeding up solving large scale sequences of linear systems from time dependent simulations of incompressible flow.
- Combining a powerful preconditioner (adaptive BDDC) and Krylov subspace recycling.



Flow around the unit sphere at Re = 300

Mathematical model

Unsteady Navier–Stokes equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}$$

$$\nabla \cdot \boldsymbol{u} = 0$$

Initial and boundary conditions

$$\boldsymbol{u} = \boldsymbol{u}_0 \quad \text{for } t = 0$$

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \Gamma_D$$

$$-\nu(\nabla \boldsymbol{u})\boldsymbol{n} + p\,\boldsymbol{n} = 0 \quad \text{on } \Gamma_N$$

- $\boldsymbol{u}$ . . . velocity
- t . . . time
- ν . . . kinematic viscosity
- p . . . pressure normalised by the (constant) density
- $\boldsymbol{f} = \boldsymbol{0}$ . . . external body force

Time semi-discretization

Backward Euler time discretization

$$\frac{\partial \boldsymbol{u}}{\partial t} \approx \frac{\boldsymbol{u}_{n+1} - \boldsymbol{u}_n}{\Delta t}$$

Semi-implicit approximation of the convective term

$$(\boldsymbol{u}_{n+1} \cdot \nabla)\boldsymbol{u}_{n+1} \approx (\boldsymbol{u}_n \cdot \nabla)\boldsymbol{u}_{n+1}$$

Time-discrete semi-implicit Navier–Stokes equations

$$\frac{1}{\Delta t}\boldsymbol{u}_{n+1} + (\boldsymbol{u}_n \cdot \nabla)\boldsymbol{u}_{n+1} - \nu\Delta\boldsymbol{u}_{n+1} + \nabla p_n = \frac{1}{\Delta t}\boldsymbol{u}_n + \boldsymbol{f}$$
$$\nabla \cdot \boldsymbol{u}_{n+1} = 0$$

- Δt . . . time-step size
- subscript $_n$ denotes quantities at time layer t_n

Time semi-discretization

Backward Euler time discretization

$$\frac{\partial \boldsymbol{u}}{\partial t} \approx \frac{\boldsymbol{u}_{n+1} - \boldsymbol{u}_n}{\Delta t}$$

Semi-implicit approximation of the convective term

$$(\boldsymbol{u}_{n+1} \cdot \nabla)\boldsymbol{u}_{n+1} \approx (\boldsymbol{u}_n \cdot \nabla)\boldsymbol{u}_{n+1}$$

Time-discrete semi-implicit Navier–Stokes equations

$$\frac{1}{\Delta t}\boldsymbol{u}_{n+1} + (\boldsymbol{u}_n \cdot \nabla)\boldsymbol{u}_{n+1} - \nu\Delta\boldsymbol{u}_{n+1} + \nabla p_n = \frac{1}{\Delta t}\boldsymbol{u}_n + \boldsymbol{f}$$

$$\nabla \cdot \boldsymbol{u}_{n+1} = 0$$

- Δt . . . time-step size
- subscript $_n$ denotes quantities at time layer t_n

Time semi-discretization

Backward Euler time discretization

$$\frac{\partial \boldsymbol{u}}{\partial t} \approx \frac{\boldsymbol{u}_{n+1} - \boldsymbol{u}_n}{\Delta t}$$

Semi-implicit approximation of the convective term

$$(\boldsymbol{u}_{n+1} \cdot \nabla)\boldsymbol{u}_{n+1} \approx (\boldsymbol{u}_n \cdot \nabla)\boldsymbol{u}_{n+1}$$

Time-discrete semi-implicit Navier–Stokes equations

$$\frac{1}{\Delta t}\boldsymbol{u}_{n+1} + (\boldsymbol{u}_n \cdot \nabla)\boldsymbol{u}_{n+1} - \nu\Delta\boldsymbol{u}_{n+1} + \nabla p_n = \frac{1}{\Delta t}\boldsymbol{u}_n + \boldsymbol{f}$$
$$\nabla \cdot \boldsymbol{u}_{n+1} = 0$$

- Δt . . . time-step size
- subscript $_n$ denotes quantities at time layer t_n

Incremental pressure-correction method

- Operator splitting studied already in [Chorin (1968)], [Temam (1969)], recent overview in [Guermond, Minev, Shen (2006)], large-scale experiments in [Šístek, Cirak (2015)]

1. A modified momentum equation is solved for the new velocity $\boldsymbol{u}_n$ as an independent (scalar) *convection-diffusion problem* for each component of velocity

$$\frac{1}{\Delta t}\boldsymbol{u}_{n+1} + (\boldsymbol{u}_n \cdot \nabla)\boldsymbol{u}_{n+1} - \nu\Delta\boldsymbol{u}_{n+1} = \frac{1}{\Delta t}\boldsymbol{u}_n - \nabla(p_n + \psi_n) + \boldsymbol{f}$$

2. The pressure corrector ψ is obtained by solving the *Poisson problem* (sometimes called *projection step*)

$$-\Delta\psi_{n+1} = -\frac{1}{\Delta t}\nabla \cdot \boldsymbol{u}_{n+1} \quad \text{for } \boldsymbol{x} \in \Omega$$

3. The new pressure p_n is obtained by a simple update

$$p_{n+1} = p_n + \psi_{n+1} - \nu\nabla \cdot \boldsymbol{u}_{n+1}$$

Incremental pressure-correction method

- Operator splitting studied already in [Chorin (1968)], [Temam (1969)], recent overview in [Guermond, Minev, Shen (2006)], large-scale experiments in [Šístek, Cirak (2015)]

1. A modified momentum equation is solved for the new velocity $\boldsymbol{u}_n$ as an independent (scalar) *convection-diffusion problem* for each component of velocity

$$\frac{1}{\Delta t}\boldsymbol{u}_{n+1} + (\boldsymbol{u}_n \cdot \nabla)\boldsymbol{u}_{n+1} - \nu\Delta\boldsymbol{u}_{n+1} = \frac{1}{\Delta t}\boldsymbol{u}_n - \nabla(p_n + \psi_n) + \boldsymbol{f}$$

2. The pressure corrector ψ is obtained by solving the *Poisson problem* (sometimes called *projection step*)

$$-\Delta\psi_{n+1} = -\frac{1}{\Delta t}\nabla \cdot \boldsymbol{u}_{n+1} \quad \text{for } \boldsymbol{x} \in \Omega$$

3. The new pressure p_n is obtained by a simple update

$$p_{n+1} = p_n + \psi_{n+1} - \nu\nabla \cdot \boldsymbol{u}_{n+1}$$

Incremental pressure-correction method

- Operator splitting studied already in [Chorin (1968)], [Temam (1969)], recent overview in [Guermond, Minev, Shen (2006)], large-scale experiments in [Šístek, Cirak (2015)]

1. A modified momentum equation is solved for the new velocity $\boldsymbol{u}_n$ as an independent (scalar) *convection-diffusion problem* for each component of velocity

$$\frac{1}{\Delta t}\boldsymbol{u}_{n+1} + (\boldsymbol{u}_n \cdot \nabla)\boldsymbol{u}_{n+1} - \nu\Delta\boldsymbol{u}_{n+1} = \frac{1}{\Delta t}\boldsymbol{u}_n - \nabla(p_n + \psi_n) + \boldsymbol{f}$$

2. The pressure corrector ψ is obtained by solving the *Poisson problem* (sometimes called *projection step*)

$$-\Delta\psi_{n+1} = -\frac{1}{\Delta t}\nabla \cdot \boldsymbol{u}_{n+1} \quad \text{for } \boldsymbol{x} \in \Omega$$

3. The new pressure p_n is obtained by a simple update

$$p_{n+1} = p_n + \psi_{n+1} - \nu\nabla \cdot \boldsymbol{u}_{n+1}$$

Incremental pressure-correction method

- Operator splitting studied already in [Chorin (1968)], [Temam (1969)], recent overview in [Guermond, Minev, Shen (2006)], large-scale experiments in [Šístek, Cirak (2015)]

1. A modified momentum equation is solved for the new velocity $\boldsymbol{u}_n$ as an independent (scalar) *convection-diffusion problem* for each component of velocity

$$\frac{1}{\Delta t}\boldsymbol{u}_{n+1} + (\boldsymbol{u}_n \cdot \nabla)\boldsymbol{u}_{n+1} - \nu\Delta\boldsymbol{u}_{n+1} = \frac{1}{\Delta t}\boldsymbol{u}_n - \nabla(p_n + \psi_n) + \boldsymbol{f}$$

2. The pressure corrector ψ is obtained by solving the *Poisson problem* (sometimes called *projection step*)

$$-\Delta\psi_{n+1} = -\frac{1}{\Delta t}\nabla \cdot \boldsymbol{u}_{n+1} \quad \text{for } \boldsymbol{x} \in \Omega$$

3. The new pressure p_n is obtained by a simple update

$$p_{n+1} = p_n + \psi_{n+1} - \nu\nabla \cdot \boldsymbol{u}_{n+1}$$

Finite element method

Discretisation

- hexaheral Taylor-Hood Q_2–Q_1 elements
- velocity – continuous piecewise tri-quadratic polynomial
- pressure – continuous piecewise tri-linear polynomial

Finite element function for pressure corrector

$$\psi_{n+1} = \sum_{i=1}^{m} \varphi_i u_i^{n+1}$$

System of algebraic equations

$$\mathbf{A}\,\mathbf{u}^{n+1} = \mathbf{f}^{n+1} \qquad \mathbf{A} \in \mathbb{R}^{m,m}, \mathbf{u}^{n+1} \in \mathbb{R}^m, \mathbf{f}^{n+1} \in \mathbb{R}^m$$

Finite element method

Discretisation

- hexaheral Taylor-Hood Q_2–Q_1 elements
- velocity – continuous piecewise tri-quadratic polynomial
- pressure – continuous piecewise tri-linear polynomial

Finite element function for pressure corrector

$$\psi_{n+1} = \sum_{i=1}^{m} \varphi_i u_i^{n+1}$$

System of algebraic equations

$$\mathbf{A}\,\mathbf{u}^{n+1} = \mathbf{f}^{n+1} \qquad \mathbf{A} \in \mathbb{R}^{m,m}, \mathbf{u}^{n+1} \in \mathbb{R}^m, \mathbf{f}^{n+1} \in \mathbb{R}^m$$

Finite element method

Discretisation

- hexaheral Taylor-Hood Q_2–Q_1 elements
- velocity – continuous piecewise tri-quadratic polynomial
- pressure – continuous piecewise tri-linear polynomial

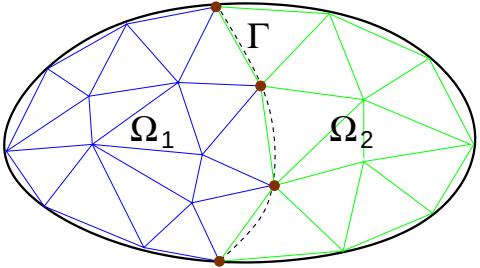
Finite element function for pressure corrector

$$\psi_{n+1} = \sum_{i=1}^{m} \varphi_i u_i^{n+1}$$

System of algebraic equations

$$\mathbf{A}\,\mathbf{u}^{n+1} = \mathbf{f}^{n+1} \qquad \mathbf{A} \in \mathbb{R}^{m,m}, \mathbf{u}^{n+1} \in \mathbb{R}^m, \mathbf{f}^{n+1} \in \mathbb{R}^m$$

Iterative substructuring

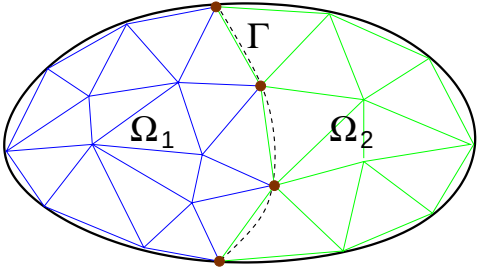


- Ω_1, Ω_2 ... subdomains (substructures)
- Γ ... interface
- The first step is reduction of the problem to the **interface** Γ.

Interface problem

$$\mathbf{S}\mathbf{u}^{\Gamma} = \mathbf{g}$$

Iterative substructuring



- Ω_1, Ω_2 ... subdomains (substructures)
- Γ ... interface
- The first step is reduction of the problem to the **interface** Γ.

Interface problem

$$\mathbf{S}\mathbf{u}^{\Gamma} = \mathbf{g}$$

BDDC preconditioner

Balancing Domain Decomposition based on Constraints (BDDC)

- Selection of the coarse DOFs.
- Building a coarse problem

$$\mathbf{S}_C = \sum_{i=1}^{N} R_{Ci}^T \Psi_i^T \mathbf{S}_i \Psi_i R_{Ci}$$

[Dohrmann (2003)], [Cros (2003)], and [Fragakis and Papadrakakis (2003)]

Multilevel BDDC

- Consider $\mathbf{S}_C \mathbf{u}_C = \mathbf{r}_C^k$ as first coarse level $\rightarrow \mathbf{S}_{C_1} \mathbf{u}_{C_1} = \mathbf{r}_{C_1}^k$.
- Instead of direct solve of $\mathbf{S}_{C_1} \mathbf{u}_{C_1} = \mathbf{r}_{C_1}^k$ — **one step of BDDC**.

[Tu (2007)], [Mandel, Sousedík, Dohrmann (2008)]

Adaptive–Multilevel BDDC

- Adding adaptive selection of constraints.
- Solving eigenvalue problems and selecting eigenvectors for defining optimal coarse DOFs.

[Mandel, Sousedík (2007)],[Mandel, Sousedík, Šístek (2012)],[Sousedík, Šístek, Mandel (2013)]

BDDC preconditioner

Balancing Domain Decomposition based on Constraints (BDDC)

- Selection of the coarse DOFs.
- Building a coarse problem

$$\mathbf{S}_C = \sum_{i=1}^{N} R_{Ci}^T \Psi_i^T \mathbf{S}_i \Psi_i R_{Ci}$$

[Dohrmann (2003)], [Cros (2003)], and [Fragakis and Papadrakakis (2003)]

Multilevel BDDC

- Consider $\mathbf{S}_C \mathbf{u}_C = \mathbf{r}_C^k$ as first coarse level $\rightarrow \mathbf{S}_{C_1} \mathbf{u}_{C_1} = \mathbf{r}_{C_1}^k$.
- Instead of direct solve of $\mathbf{S}_{C_1} \mathbf{u}_{C_1} = \mathbf{r}_{C_1}^k$ — **one step of BDDC**.

[Tu (2007)], [Mandel, Sousedík, Dohrmann (2008)]

Adaptive–Multilevel BDDC

- Adding adaptive selection of constraints.
- Solving eigenvalue problems and selecting eigenvectors for defining optimal coarse DOFs.

[Mandel, Sousedík (2007)],[Mandel, Sousedík, Šístek (2012)],[Sousedík, Šístek, Mandel (2013)]

BDDC preconditioner

Balancing Domain Decomposition based on Constraints (BDDC)

- Selection of the coarse DOFs.
- Building a coarse problem

$$\mathbf{S}_C = \sum_{i=1}^{N} R_{Ci}^T \Psi_i^T \mathbf{S}_i \Psi_i R_{Ci}$$

[Dohrmann (2003)], [Cros (2003)], and [Fragakis and Papadrakakis (2003)]

Multilevel BDDC

- Consider $\mathbf{S}_C \mathbf{u}_C = \mathbf{r}_C^k$ as first coarse level $\rightarrow \mathbf{S}_{C_1} \mathbf{u}_{C_1} = \mathbf{r}_{C_1}^k$.
- Instead of direct solve of $\mathbf{S}_{C_1} \mathbf{u}_{C_1} = \mathbf{r}_{C_1}^k$ — **one step of BDDC**.

[Tu (2007)], [Mandel, Sousedík, Dohrmann (2008)]

Adaptive–Multilevel BDDC

- Adding adaptive selection of constraints.
- Solving eigenvalue problems and selecting eigenvectors for defining optimal coarse DOFs.

[Mandel, Sousedík (2007)],[Mandel, Sousedík, Šístek (2012)],[Sousedík, Šístek, Mandel (2013)]

Sequence of linear systems and its properties

Altogether, we are solving **a sequence of linear algebraic systems** with the same SPD matrix and varying right-hand side.

- The systems are solved by PCG.
- Both the action of the system matrix and the preconditioner are quite costly.
- The length of the vectors is reduced so that we can afford saving more of them.
- The preconditioned system is well conditioned ($\kappa(\mathbf{M}_{BDDC}^{-1}\mathbf{S}) = O(1)$).

We denote the kth system by

$$M^{-1}Ax^{(k)} = M^{-1}b^{(k)}.$$

Steps for acceleration

1. The initial guess and the stopping criterion.
2. Deflated CG with Krylov subspace recycling.
3. Adaptive BDDC preconditioner.

Sequence of linear systems and its properties

Altogether, we are solving **a sequence of linear algebraic systems** with the same SPD matrix and varying right-hand side.

- The systems are solved by PCG.
- Both the action of the system matrix and the preconditioner are quite costly.
- The length of the vectors is reduced so that we can afford saving more of them.
- The preconditioned system is well conditioned ($\kappa(\mathbf{M}_{BDDC}^{-1}\mathbf{S}) = O(1)$).

We denote the kth system by

$$M^{-1}Ax^{(k)} = M^{-1}b^{(k)}.$$

Steps for acceleration

1. The initial guess and the stopping criterion.
2. Deflated CG with Krylov subspace recycling.
3. Adaptive BDDC preconditioner.

Sequence of linear systems and its properties

Altogether, we are solving **a sequence of linear algebraic systems** with the same SPD matrix and varying right-hand side.

- The systems are solved by PCG.
- Both the action of the system matrix and the preconditioner are quite costly.
- The length of the vectors is reduced so that we can afford saving more of them.
- The preconditioned system is well conditioned ($\kappa(\mathbf{M}_{BDDC}^{-1}\mathbf{S}) = O(1)$).

We denote the kth system by

$$M^{-1}Ax^{(k)} = M^{-1}b^{(k)}.$$

Steps for acceleration

1. The initial guess and the stopping criterion.
2. Deflated CG with Krylov subspace recycling.
3. Adaptive BDDC preconditioner.

Stopping criteria

By the starting residual (res_0)

$$\frac{\|r^{(k)}\|}{\|r^{(0)}\|} < 10^{-6}$$

- $r^{(k)}$ is the residual in the k-th iteration, and $r^{(0)} = b - Ax^{(0)}$ is the starting residual

By the norm of the right-hand side (rhs)

$$\frac{\|r^{(k)}\|}{\|b\|} < 10^{-6}$$

- b is the vector of the right-hand side

Stopping criteria

By the starting residual (res_0)

$$\frac{\|r^{(k)}\|}{\|r^{(0)}\|} < 10^{-6}$$

- $r^{(k)}$ is the residual in the k-th iteration, and $r^{(0)} = b - Ax^{(0)}$ is the starting residual

By the norm of the right-hand side (rhs)

$$\frac{\|r^{(k)}\|}{\|b\|} < 10^{-6}$$

- b is the vector of the right-hand side

Deflated CG with Krylov subspace recycling

Space decomposition

$$\mathbb{R}^m = \mathcal{U} + \mathcal{V}$$

- $\mathcal{U}$ is a *deflation* subspace of a small dimension
- $\mathcal{V}$ is the orthogonal complement of $\mathcal{U}$

Seeking solution

- directly on $\mathcal{U}$ with $W = [w_1, w_2, \ldots, w_k]$ being a basis of $\mathcal{U}$
- iteratively on $\mathcal{V}$

[Nicolaides (1987)], [Dostál (1988)], [Farhat, Crivelli, Roux (1993)], [Saad, Yeung, Erhel, Guyomarc'h (2000)]

Deflated CG with Krylov subspace recycling

Space decomposition

$$\mathbb{R}^m = \mathcal{U} + \mathcal{V}$$

- $\mathcal{U}$ is a *deflation* subspace of a small dimension
- $\mathcal{V}$ is the orthogonal complement of $\mathcal{U}$

Seeking solution

- directly on $\mathcal{U}$ with $W = [w_1, w_2, \ldots, w_k]$ being a basis of $\mathcal{U}$
- iteratively on $\mathcal{V}$

[Nicolaides (1987)], [Dostál (1988)], [Farhat, Crivelli, Roux (1993)], [Saad, Yeung, Erhel, Guyomarc'h (2000)]

Deflation spaces — naive choices

Case 1 (first search vectors)

- The matrix W was set as first ℓ search vectors and then remained unchanged for the rest of computation.

Case 2 (last search vectors)

- After solving the $(i-1)$th system and saving k vectors in $P^{(i-1)}$, the deflation basis was updated as

$$W^{(i)} = [w_{k+1}, w_{k+2}, \ldots, w_\ell, p_1, p_2, \ldots, p_k]$$

Deflation spaces — naive choices

Case 1 (first search vectors)

- The matrix W was set as first ℓ search vectors and then remained unchanged for the rest of computation.

Case 2 (last search vectors)

- After solving the $(i-1)$th system and saving k vectors in $P^{(i-1)}$, the deflation basis was updated as

$$W^{(i)} = [w_{k+1}, w_{k+2}, \ldots, w_\ell, p_1, p_2, \ldots, p_k]$$

Ritz approximation

Classical Ritz values

$$V^T A V w = \theta V^T M V w$$

- Setting $y = Vw$ as an approximation to an eigenvector of $M^{-1}A$.

Harmonic Ritz values

- Only the operation $M^{-1}v$ is available

$$V^T A M^{-1} A V w = \theta V^T A V w$$

Case 3 (smallest Ritz values)

- To form $W^{(i)}$ we use Ritz approximations corresponding to the *smallest* Ritz values.

Case 4 (largest Ritz values)

- To form $W^{(i)}$ we use Ritz approximations corresponding to the *largest* Ritz values.

Ritz approximation

Classical Ritz values

$$V^T A V w = \theta V^T M V w$$

- Setting $y = Vw$ as an approximation to an eigenvector of $M^{-1}A$.

Harmonic Ritz values

- Only the operation $M^{-1}v$ is available

$$V^T A M^{-1} A V w = \theta V^T A V w$$

Case 3 (smallest Ritz values)

- To form $W^{(i)}$ we use Ritz approximations corresponding to the *smallest* Ritz values.

Case 4 (largest Ritz values)

- To form $W^{(i)}$ we use Ritz approximations corresponding to the *largest* Ritz values.

Ritz approximation

Classical Ritz values

$$V^T A V w = \theta V^T M V w$$

- Setting $y = Vw$ as an approximation to an eigenvector of $M^{-1}A$.

Harmonic Ritz values

- Only the operation $M^{-1}v$ is available

$$V^T A M^{-1} A V w = \theta V^T A V w$$

Case 3 (smallest Ritz values)

- To form $W^{(i)}$ we use Ritz approximations corresponding to the *smallest* Ritz values.

Case 4 (largest Ritz values)

- To form $W^{(i)}$ we use Ritz approximations corresponding to the *largest* Ritz values.

Ritz approximation

Classical Ritz values

$$V^T A V w = \theta V^T M V w$$

- Setting $y = Vw$ as an approximation to an eigenvector of $M^{-1}A$.

Harmonic Ritz values

- Only the operation $M^{-1}v$ is available

$$V^T A M^{-1} A V w = \theta V^T A V w$$

Case 3 (smallest Ritz values)

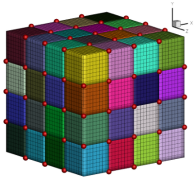
- To form $W^{(i)}$ we use Ritz approximations corresponding to the *smallest* Ritz values.

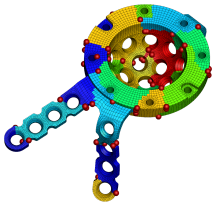
Case 4 (largest Ritz values)

- To form $W^{(i)}$ we use Ritz approximations corresponding to the *largest* Ritz values.

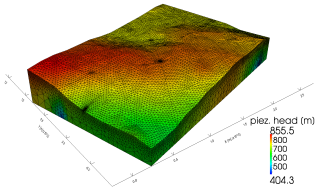
BDDCML solver library

- http://users.math.cas.cz/~sistek/software/bddcml.html
- open-source (LGPL license)
- Fortran 95 + MPI library
- tested on up to 64k cores for systems with up to 2B unknowns

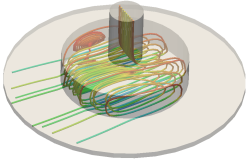




reliability of artificial joints
(IT CAS)

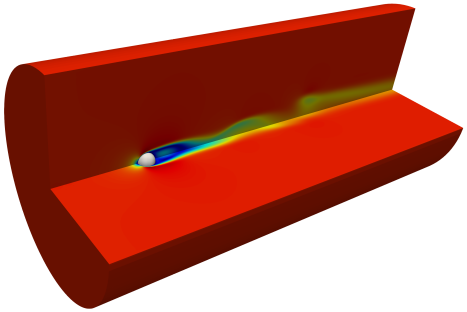


subsurface water flow
(TU Liberec)



flow in hydrostatic bearings
(CTU Prague)

Numerical results

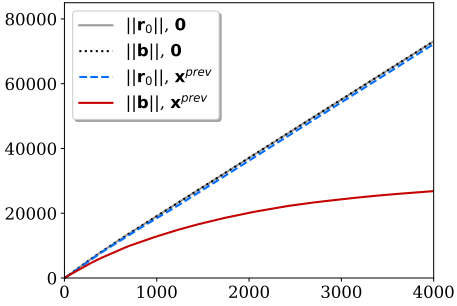


Flow around the unit sphere.

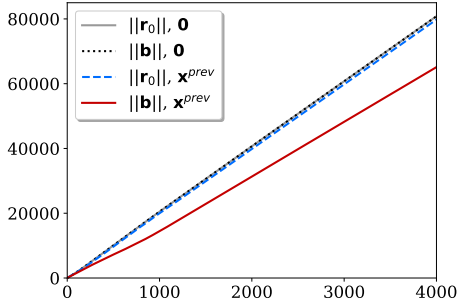
- Reynolds number equal to 100 (transient) and **300** (periodic).
- Decomposed into 1024 subdomains.
- 1.4M unknowns for pressure.
- 3-level BDDC method – baseline approach.
- 4000 time steps.
- Computed at *Karolina@IT4I*.

Effect of the stopping criterion

Cumulative number of iterations over all time steps.



Re = 100 (transient)



Re = 300 (periodic)

Effect of Krylov subspace recycling

Results for $\dim(W) = 50$.

variant	#its. min–max(avg.)	1 iter [s]	t./step [s]
no deflation	14-26(15.5)	0.028	0.43
case 1	13-24(13.2)	0.031	0.40
case 2	13-18(13.9)	0.031	0.43
case 3	14-19(14.3)	0.031	0.44
case 4	11-17(**11.3**)	0.031	**0.36**

All variants with " $x_0 = x_{prev}$ " and "rhs" stopping criterion.

Changing the dimension of the deflation space

basis size dim(W)	#its. min–max(avg.)	1 iter. [s]	t./step [s]
25	12-19(12.4)	0.030	0.37
30	12-18(12.3)	0.029	0.36
40	11-18(11.5)	0.031	**0.35**
50	11-17(11.3)	0.031	0.36
100	9-18(10.7)	0.036	0.38
200	6-18(8.7)	0.044	0.41
400	5-18(8.4)	0.044	0.41

Nevertheless, we use $\dim(W) = 50$ in further experiments.

Preconditioner with adaptive selection of constraints

- Adaptive selection of constraints for changing *condition number indicator* τ.

variant	τ	#its. min–max(avg.)	1 iter. [s]	t./step [s]	#its. step 1	t. step 1 [s]
rec 0	3.0	11-19(11.4)	0.030	0.33	17	45.55
rec 50	3.5	9-12(9.7)	0.033	0.32	17	41.21
	3.0	9-12(9.2)	0.033	**0.31**	17	42.44
	2.5	9-12(9.2)	0.034	0.31	16	42.45
	2.0	8-12(8.5)	0.037	0.32	16	42.84
	1.5	8-12(8.3)	0.052	0.44	16	47.79
all the previous variants					23	<2

Summary

variant	#its. min–max(avg.)	1 iter. [s]	t./step [s]	#its. step 1	t. step 1 [s]
rec0	14-26(15.5)	0.028	0.43	23	1.53
rec50	11-17(11.3)	0.031	0.36	23	1.70
rec0 + adaptive 3.0	11-19(11.4)	0.030	0.33	17	45.55
rec50 + adaptive 3.0	9-12(9.2)	0.033	**0.31**	17	42.44

All variants with " $x_0 = x_{prev}$ " and "rhs" stopping criterion.

Larger problem with 15.6M unknowns

variant		#its. min–max(avg.)	1 iter.[s]	t./step[s]	#its. step 1	t. step 1[s]
no recycling	res_0	34-47(36.0)	0.110	3.98	47	8.37
	rhs	31-54(33.3)	0.107	3.59	47	8.24
rec50		22-35(23.3)	0.11	2.67	47	8.43
rec0+adaptive 3.0	rhs	17-30(18.4)	0.125	2.31	26	179.86
rec50+adaptive 3.0		13-21(14.3)	0.132	**1.89**	26	177.87

All variants with " $x_0 = x_{prev}$ ".

Conclusions

Acceleration of unsteady flow computations

- Starting from the previous solution with norm relative to right-hand side.
- Recycling using the largest harmonic Ritz values.
- Adaptive BDDC with a suitable threshold on eigenvalues in an unusual setting.
- Saving about 50% of the computational time.

Future work

- GPU acceleration.
- Larger problems.

Conclusions

Acceleration of unsteady flow computations

- Starting from the previous solution with norm relative to right-hand side.
- Recycling using the largest harmonic Ritz values.
- Adaptive BDDC with a suitable threshold on eigenvalues in an unusual setting.
- Saving about 50% of the computational time.

Future work

- GPU acceleration.
- Larger problems.

Contacts

Happy Birthday, Zdeněk!

DD27, Prague, 2022

Hanek, M., Papež, J., Šístek, J.
Speeding up an unsteady flow simulation by adaptive BDDC and Krylov subspace recycling.
Computer Methods in Applied Mechanics and Engineering 452 (2026), 118788.

Thank you for your attention!

Contacts

Happy Birthday, Zdeněk!

DD27, Prague, 2022

Hanek, M., Papež, J., Šístek, J.
Speeding up an unsteady flow simulation by adaptive BDDC and Krylov subspace recycling.
Computer Methods in Applied Mechanics and Engineering 452 (2026), 118788.

Thank you for your attention!

Contacts

Happy Birthday, Zdeněk!

DD27, Prague, 2022

Hanek, M., Papež, J., Šístek, J.
Speeding up an unsteady flow simulation by adaptive BDDC and Krylov subspace recycling.
Computer Methods in Applied Mechanics and Engineering 452 (2026), 118788.

Thank you for your attention!