

BDDC preconditioners for non-conforming polytopal hybrid discretisation methods. Part II: The preconditioner.

Jordi Manyer¹

joint work with S. Badia¹, J. Droniou^{1,2} and J. Tushar¹

¹Monash University, Melbourne, Australia,

²CNRS & University of Montpellier, Montpellier, France

29th International Conference on
Domain Decomposition Methods
PoliMi, June 2025

Polytopal methods

- Increasingly popular choice for dealing with **complex geometries**.
- Often **non-conforming** and **hybrid** in nature (DG, HHO, HDG, non-conforming VEM, ...).
- Efficient solvers for the resulting **large sparse linear systems** are needed.

Balancing domain decomposition by constraints (BDDC)

- Popular family of **robust** and **scalable** preconditioners.
- Traditional robustness proofs rely on **continuous trace and lifting inequalities**, which are not available for non-conforming methods.

Table of Contents

1. Truncation estimates for piece-wise broken polynomials

2. Preconditioner & condition number estimates

3. Numerical results

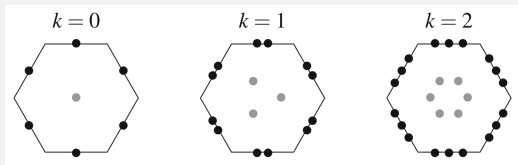
Hybrid spaces

We consider the **hybrid space** $\underline{U}_h = U_h \times U_h^\partial$, where

$$U_h \doteq \bigtimes_{t \in \mathcal{T}_h} \mathbb{P}_k(t), \quad U_h^\partial \doteq \bigtimes_{f \in \mathcal{F}_h} \mathbb{P}_k(f), \quad U_h^{\partial, \text{bd}} \doteq \bigtimes_{f \in \mathcal{F}_h^{\text{bd}}} \mathbb{P}_k(f)$$

For each $\underline{v}_h = ((v_t)_{t \in \mathcal{T}_h}, (v_f)_{f \in \mathcal{F}_t}) \in \underline{U}_h$ we define the discrete $H^1(\Omega)$ -seminorm

$$|\underline{v}_h|_{1,h} \doteq \left(\sum_{t \in \mathcal{T}_h} |\underline{v}_h|_{1,t}^2 \right)^{1/2}$$
$$|\underline{v}_h|_{1,t}^2 \doteq \|\nabla v_t\|_{L^2(t)}^2 + \sum_{f \in \mathcal{F}_t} h_t^{-1} \|v_f - v_t\|_{L^2(f)}^2.$$



Discrete trace theory (I)¹

We define the discrete $H^{1/2}(\partial\Omega)$ -seminorm

$$|w_h|_{1/2,h}^2 \doteq \sum_{f \in \mathcal{F}_h^{\text{bd}}} h_f^{-1} \|w_f - \bar{w}_f\|_{L^2(f)}^2 + \sum_{(f,f') \in \mathcal{FF}_h^{\text{bd}}} |f|_{d-1} |f'|_{d-1} \frac{|\bar{w}_f - \bar{w}_{f'}|^2}{|x_f - x_{f'}|^d},$$

Theorem: Trace inequality

$$|\gamma(\underline{v}_h)|_{1/2,h} \lesssim |\underline{v}_h|_{1,h} \quad \forall \underline{v}_h \in \underline{U}_h$$

Theorem: Lifting inequality

There exists a lifting $\mathcal{L}_h : U_h^{\partial,\text{bd}} \rightarrow \underline{U}_h$ such that

$$|\mathcal{L}_h(w_h)|_{1,h} \lesssim |w_h|_{1/2,h} \quad \forall w_h \in U_h^{\partial,\text{bd}}$$

¹Santiago Badia, Jérôme Droniou, and Jai Tushar. *A discrete trace theory for non-conforming polytopal hybrid discretisation methods*. 2024. arXiv: 2409.15863 [math.NA].

Discrete trace theory (II)

Theorem: Truncation estimate

Consider $\Gamma = \bigcup_{f \in \mathcal{F}_h(\Gamma)} \text{cl}(f) \subset \partial\Omega$ and $\underline{v}_h \in \underline{U}_h$ such that

$$\int_{\Gamma} \gamma(\underline{v}_h) = 0$$

Define $w_h \in U_h^{\partial, \text{bd}}$ as

$$\forall f \in \mathcal{F}_h^{\text{bd}} \quad w_f = \begin{cases} v_f & \text{if } f \in \mathcal{F}_h(\Gamma), \\ 0 & \text{if } f \notin \mathcal{F}_h(\Gamma). \end{cases}$$

Then, under regularity assumptions for Γ , we have

$$|w_h|_{1/2, h} \lesssim \left(1 + \ln \left(\frac{\text{diam}(\Omega)}{h} \right) \right) |\underline{v}_h|_{1, h},$$

Table of Contents

1. Truncation estimates for piece-wise broken polynomials

2. Preconditioner & condition number estimates

3. Numerical results

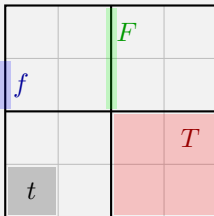
Domains and meshes

Fine mesh: For a domain Ω with boundary $\partial\Omega$, we consider

- A **partition** $\mathcal{T}_h$ of Ω into **polytopes** t .
- Its **skeleton** $\mathcal{F}_h = \mathcal{F}_h^{\text{bd}} \cup \mathcal{F}_h^{\text{in}}$ with **faces** f .
- For each $t \in \mathcal{T}_h$, we denote by $\mathcal{F}_h(t)$ the subset of its faces.

Coarse mesh: we consider an agglomeration $(\mathcal{T}_H, \mathcal{F}_H)$ s.t.

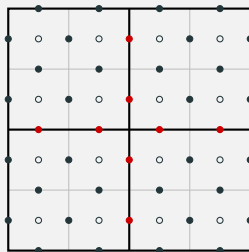
- Each **subdomain** $T \in \mathcal{T}_H$ is a union of polytopes $t \in \mathcal{T}_h(T)$.
- Each **coarse face** $F \in \mathcal{F}_H$ is a union of faces $f \in \mathcal{F}_h(F)$.
- For each $T \in \mathcal{T}_H$, we denote by $\mathcal{F}_H(T)$ the subset of its coarse faces.



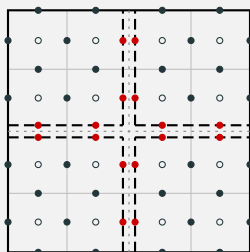
Sub-assembled spaces

Given the fine mesh $(\mathcal{T}_h, \mathcal{F}_h)$ and its coarse agglomeration $(\mathcal{T}_H, \mathcal{F}_H)$:

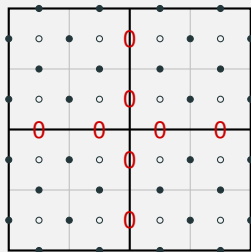
- **Hybrid space** $\hat{\underline{U}}_h$ - Single-valued on interfaces
- **Broken space** $\underline{U}_h$ - Double-valued on interfaces
- **Bubble space** $\underline{U}_{h,0}$ - Zero on interfaces



$$\hat{\underline{U}}_h = U_h \times \hat{U}_h^\partial$$



$$\underline{U}_h = U_h \times U_h^\partial$$



$$\underline{U}_{h,0} = U_h \times U_{h,0}^\partial$$

Coarse constraints

We take as **coarse DoFs** the **average on each coarse face** $F \in \mathcal{F}_H$

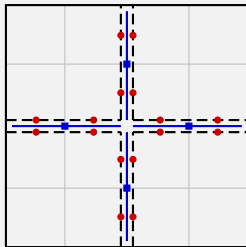
$$\lambda_F(\underline{v}_h) = \frac{1}{|F|} \int_F \gamma(\underline{v}_h) , \quad \Lambda = \bigcup_{F \in \mathcal{F}_H} \lambda_F$$

We define the **BDDC space** $\tilde{\underline{U}}_h$ of functions that are

- double valued on interfaces
- **single-valued coarse DoFs**

$$\lambda_F(\underline{v}_h(T)) = \lambda_F(\underline{v}_h(T'))$$

for T, T' neighbors of F .



$$\tilde{\underline{U}}_h = U_h \times \tilde{U}_h^\partial$$

Abstract problem formulation

We consider $\mathcal{A}$ a coercive differential operator on Ω , and l a linear functional. We assume homogeneous Dirichlet boundary conditions.

Continuous problem: Find $u \in U$ such that for all $v \in U_0$ we have

$$\langle \mathcal{A}u, v \rangle = \langle l, v \rangle$$

Discrete problem: Find $\underline{u}_h \in \hat{\underline{U}}_h$ such that for all $\underline{v}_h \in \hat{\underline{U}}_{h,0}$ we have

$$\langle \hat{\mathcal{A}}_h \underline{u}_h, \underline{v}_h \rangle = \langle l_h, \underline{v}_h \rangle$$

where:

- $\mathcal{A}_h$ the discrete approximation of $\mathcal{A}$ on $\underline{U}_h$.
- $\hat{\mathcal{A}}_h$ the restriction of $\mathcal{A}_h$ on $\hat{\underline{U}}_h$.
- $\tilde{\mathcal{A}}_h$ the restriction of $\mathcal{A}_h$ on $\tilde{\underline{U}}_h$.
- $\mathcal{A}_{h,0}$ the restriction of $\mathcal{A}_h$ on $\underline{U}_{h,0}$.

Discrete harmonic spaces

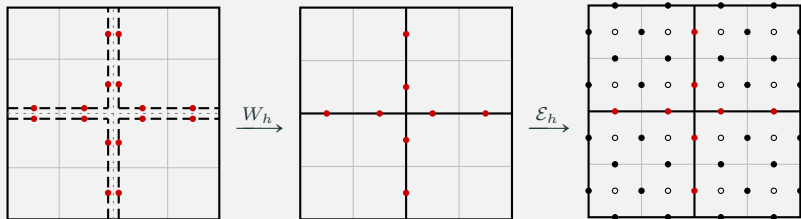
Discrete harmonic space $\underline{H}_h$ such that $\underline{H}_h \perp_{\mathcal{A}_h} \underline{U}_{h,0}$

Harmonic extension operator $\mathcal{E}_h : \underline{U}_h \rightarrow \underline{H}_h$ such that given $\underline{w}_h \in \underline{U}_h$,

$$\langle \mathcal{A}_h \mathcal{E}_h \underline{w}_h, \underline{v}_h \rangle = 0 \quad \forall \underline{v}_h \in \underline{U}_{h,0}$$

$$\mathcal{E}_h \underline{w}_h = \underline{w}_h \quad \text{on } \mathcal{F}_H$$

- Functions in $\underline{H}_h$ are fully determined by their values on the interface.
- $\mathcal{E}_h$ is the **minimum energy lifting** w.r.t $\mathcal{A}_h$.



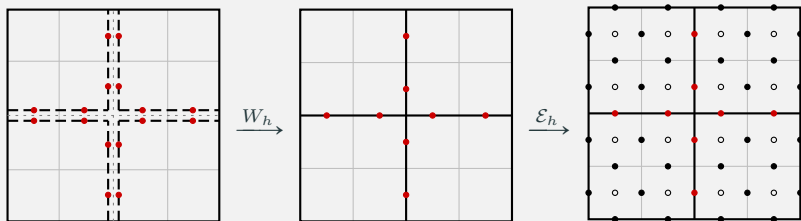
Weighting operator

Define the **weighting operator** $W_h : \underline{U}_h \rightarrow \hat{\underline{U}}_h$ such that

$$(W_h \underline{u}_h)|_F = \frac{1}{2} (\underline{u}_h(T)|_F + \underline{u}_h(T')|_F) \quad \forall F \in \mathcal{F}_H$$

where T and T' are the two subdomains sharing the face F .

We consider the operator $\mathcal{Q}_h = \mathcal{E}_h W_h : \underline{U}_h \rightarrow \hat{\underline{H}}_h$.



We define the BDDC preconditioner $\hat{\mathcal{B}}_h : \hat{\underline{U}}_h^* \rightarrow \hat{\underline{U}}_h$ as

$$\hat{\mathcal{B}}_h = \mathcal{Q}_h \tilde{\mathcal{A}}_h^{-1} \mathcal{Q}_h^T + I_0 \mathcal{A}_{h,0}^{-1} I_0^T$$

with $I_0 : \underline{U}_{h,0} \rightarrow \hat{\underline{U}}_h$ the natural injection ¹.

Theorem: Spectral equivalence

Under the assumption that $\|\cdot\|_{\mathcal{A}_h} \simeq \|\cdot\|_{1,h}$

$$\kappa \left(\hat{\mathcal{B}}_h \hat{\mathcal{A}}_h \right) \lesssim \left(1 + \ln \frac{H}{h} \right)^2$$

where the hidden constant does not depend on H , h and the number of subdomains.

¹See Appendix A for implementation details.

Proof: Spectral equivalence (I)

From abstract additive Schwartz theory¹, we have

$$\begin{aligned}\kappa\left(\hat{\mathcal{B}}_h\hat{\mathcal{A}}_h\right) &\leq \max\left\{1, \sup_{\underline{v}_h \in \tilde{\mathcal{U}}_h} \frac{\|\mathcal{Q}_h\underline{v}_h\|_{\mathcal{A}_h}^2}{\|\underline{v}_h\|_{\mathcal{A}_h}^2}\right\} \\ &\leq \max\left\{1, 1 + \sup_{\underline{v}_h \in \tilde{\mathcal{H}}_h} \frac{\|\mathcal{Q}_h\underline{v}_h - \underline{v}_h\|_{\mathcal{A}_h}^2}{\|\underline{v}_h\|_{\mathcal{A}_h}^2}\right\}.\end{aligned}$$

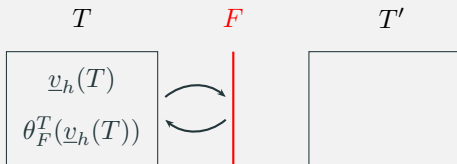
Let $\underline{w}_h = \mathcal{Q}_h\underline{v}_h - \underline{v}_h$.

¹Jan Mandel, Bedřich Sousedík, and Clark R. Dohrmann. “Multispace and multilevel BDDC”. In: *Computing* 83.2–3 (Sept. 2008), pp. 55–85. DOI: 10.1007/s00607-008-0014-7.

Proof: Spectral equivalence (II)

We define the face restriction operators θ_F^T which takes a function $\underline{v}_h$ and

1. Restrict a function to the macro face F .
2. Extends it by zero to ∂T
3. Extends it harmonically to T



Corollary: Truncation estimate

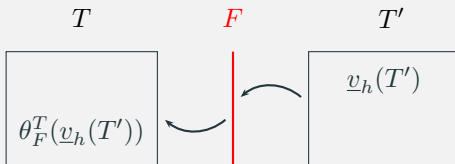
Let $F \in \mathcal{F}_H(T)$ and $\underline{v}_h(T^*) \in \underline{U}_h(T^*)$, for $T^* \in \{T, T'\}$, which satisfies $\int_F \gamma_F(\underline{v}_h(T^*)) = 0$. Then the following estimate holds:

$$|\gamma_F(\theta_F^T(\underline{v}_h))|_{1/2, h, \partial T} \lesssim \left(1 + \ln \frac{H}{h}\right) |\underline{v}_h|_{1, h, T^*}.$$

Proof: Spectral equivalence (II)

We define the face restriction operators θ_F^T which takes a function $\underline{v}_h$ and

1. Restrict a function to the macro face F .
2. Extends it by zero to ∂T
3. Extends it harmonically to T



Corollary: Truncation estimate

Let $F \in \mathcal{F}_H(T)$ and $\underline{v}_h(T^*) \in \underline{U}_h(T^*)$, for $T^* \in \{T, T'\}$, which satisfies $\int_F \gamma_F(\underline{v}_h(T^*)) = 0$. Then the following estimate holds:

$$|\gamma_F(\theta_F^T(\underline{v}_h))|_{1/2, h, \partial T} \lesssim \left(1 + \ln \frac{H}{h}\right) |\underline{v}_h|_{1, h, T^*}.$$

Proof: Spectral equivalence (III)

One can see that for $\underline{w}_h = \mathcal{Q}_h \underline{v}_h - \underline{v}_h$ we have

$$\underline{w}_h(T) = \sum_{F \in \mathcal{F}_H(T)} \theta_F^T(\underline{w}_h), \quad \theta_F^T(\underline{w}_h) = \frac{1}{2} [\theta_F^T(\underline{v}_h(T')) - \theta_F^T(\underline{v}_h(T))]$$

Then using the above and triangle inequalities, we get

$$\|\mathcal{Q}_h \underline{v}_h - \underline{v}_h\|_{\mathcal{A}_h}^2 \lesssim \sum_{T \in \mathcal{T}_H} \sum_{F \in \mathcal{F}_H(T)} \|\theta_F^T(\underline{w}_h)\|_{\mathcal{A}_h, T}^2,$$

Each of the terms can be bounded by

$$\begin{aligned} \|\theta_F^T(\underline{w}_h)\|_{\mathcal{A}_h, T}^2 &\lesssim |\gamma_{\partial T}(\theta_F^T(\underline{w}_h))|_{1/2, h, \partial T}^2 \triangleright \text{norm eq. and lifting} \\ &\lesssim |\theta_F^T(\underline{v}_h(T)) - \alpha_F|_{1/2, h, \partial T}^2 + |\theta_F^T(\underline{v}_h(T')) - \alpha_F|_{1/2, h, \partial T}^2 \end{aligned}$$

where we have added and subtracted a constant $\alpha_F \in \mathbb{R}$.

Proof: Spectral equivalence (IV)

Since $\underline{v}_h \in \widetilde{H}_h$, we can choose $\alpha_F = \lambda_F(\underline{v}_h(T)) = \lambda_F(\underline{v}_h(T'))$ so that

$$\int_F \gamma_F (\underline{v}_h(T') - \alpha_F) = \int_F \gamma_F (\underline{v}_h(T) - \alpha_F) = 0$$

Then

$$\begin{aligned} \|\theta_F^T(\underline{w}_h)\|_{\mathcal{A}_h, T}^2 &\lesssim \left(1 + \ln \frac{H}{h}\right)^2 (|\underline{v}_h(T)|_{1,h,T}^2 + |\underline{v}_h(T')|_{1,h,T'}^2) &> \text{truncation} \\ &\lesssim \left(1 + \ln \frac{H}{h}\right)^2 (\|\underline{v}_h(T)\|_{\mathcal{A}_h, T}^2 + \|\underline{v}_h(T')\|_{\mathcal{A}_h, T'}^2) &> \text{norm eq.} \end{aligned}$$

and finally

$$\|\underline{w}_h\|_{\mathcal{A}_h}^2 \lesssim \left(1 + \ln \frac{H}{h}\right)^2 \|\underline{v}_h\|_{\mathcal{A}_h}^2$$

Table of Contents

1. Truncation estimates for piece-wise broken polynomials

2. Preconditioner & condition number estimates

3. Numerical results

Problem setup

We solve the Poisson problem with Dirichlet bcs on $\Omega = [0, 1]^2$

$$-\Delta u = f \quad \text{in } \Omega, \quad u = g \quad \text{on } \partial\Omega$$

We use $N_c \times N_c$ simplexified cartesian meshes, and their Voronoi counterparts, partitioned into $N_p \times N_p$ processors.

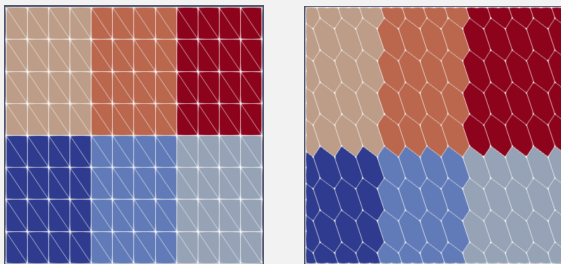


Figure 1: Left: simplexified mesh. **Right:** Voronoi mesh. Distributed amongst $N_p = 6$ processors, with colors mapping to the processor id.

We test our preconditioner for several popular schemes in the literature:

- Hybrid High-Order (HHO)¹
- Mixed-order HHO²
- Hybridizable Discontinuous Galerkin (HDG)^{3,4}

¹Daniele A. Di Pietro and Alexandre Ern. “A hybrid high-order locking-free method for linear elasticity on general meshes”. In: *Computer Methods in Applied Mechanics and Engineering* 283 (2015), pp. 1–21.

DOI: <https://doi.org/10.1016/j.cma.2014.09.009>.

²Matteo Cicuttin, Alexandre Ern, and Nicolas Pignet. *Hybrid High-Order Methods: A Primer with Applications to Solid Mechanics*. Mathematical Engineering. Springer International Publishing, 2021.

DOI: 10.1007/978-3-030-81477-9.

³Bernardo Cockburn, Jayadeep Gopalakrishnan, and Raytcho Lazarov. “Unified Hybridization of Discontinuous Galerkin, Mixed, and Continuous Galerkin Methods for Second Order Elliptic Problems”. In: *SIAM Journal on Numerical Analysis* (2009). DOI: 10.1137/070706616.

⁴Shukai Du and Francisco–Javier Sayas. “A note on devising HDG+ projections on polyhedral elements”. In: *arXiv: Numerical Analysis* (2020). DOI: 10.1090/mcom/3573.

Implementation Julia, using the **Gridap**¹ ecosystem. MPI support through **GridapDistributed.jl** and **PartitionedArrays.jl**². Solvers implemented in **GridapSolvers.jl**³.

We perform **weak scalability** tests, that is

- constant local problem size, $N_c/N_p \sim H/h$
- increasing N_p up to 768 processors

All experiments are run on the Gadi supercomputer at NCI.

¹Santiago Badia and Francesc Verdugo. “Gridap: An extensible Finite Element toolbox in Julia”. In: *Journal of Open Source Software* 5.52 (2020), p. 2520. DOI: 10.21105/joss.02520.

²Santiago Badia, Alberto F. Martín, and Francesc Verdugo. “GridapDistributed: a massively parallel finite element toolbox in Julia”. In: *Journal of Open Source Software* 7.74 (2022), p. 4157. DOI: 10.21105/joss.04157.

³Jordi Manyer, Alberto F. Martín, and Santiago Badia. “GridapSolvers.jl: Scalable multiphysics finite element solvers in Julia”. In: *Journal of Open Source Software* 9.102 (2024), p. 7162. DOI: 10.21105/joss.07162.

Results (I)

HDG of structured meshes

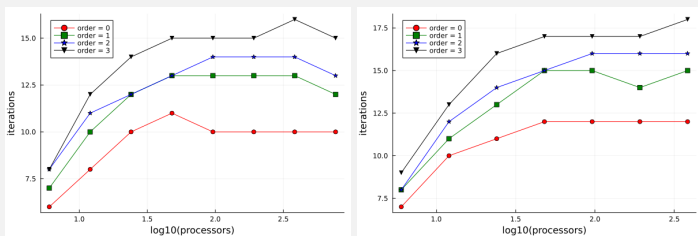


Figure 2: Number of CG iterations, as a function of the number of processors. Cases: HDG with $H/h = 8$ (left) and $H/h = 16$ (right).

Results (II)

HHO of structured meshes

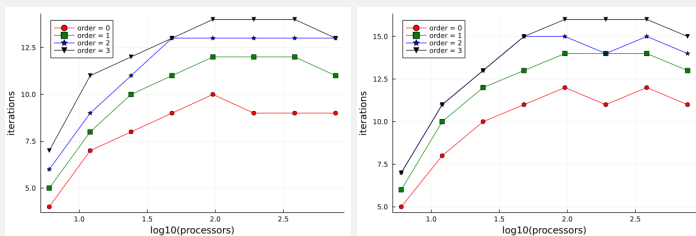


Figure 3: Number of CG iterations, as a function of the number of processors. Cases: HHO with $H/h = 8$ (left) and $H/h = 16$ (right).

Results (III)

Mixed-order HHO of structured meshes

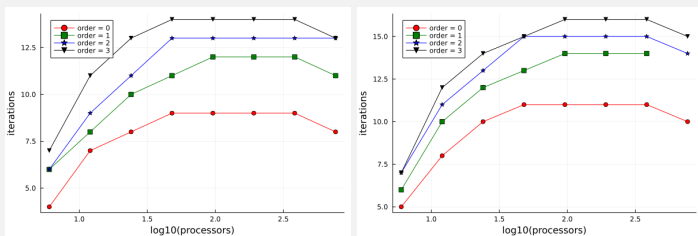


Figure 4: Number of CG iterations, as a function of the number of processors. Cases: Mixed-order HHO with $H/h = 8$ (left) and $H/h = 16$ (right).

Results (IV)

HDG+ of voronoi meshes

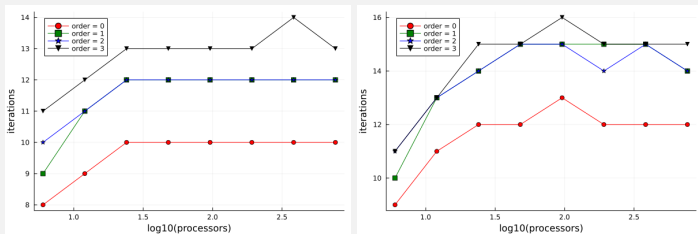


Figure 5: Number of CG iterations, as a function of the number of processors. Cases: HDG+ with $H/h = 8$ (left) and $H/h = 16$ (right).

Results (V)

HHO of voronoi meshes

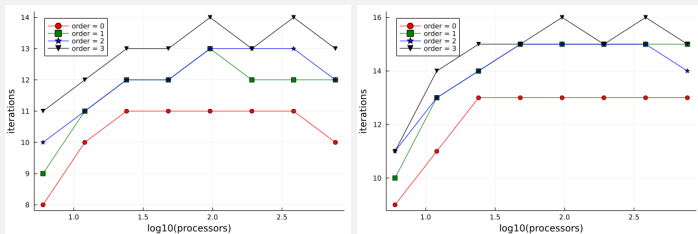


Figure 6: Number of CG iterations, as a function of the number of processors. Cases: HHO with $H/h = 8$ (left) and $H/h = 16$ (right).

Results (VI)

Mixed-order HHO of voronoi meshes

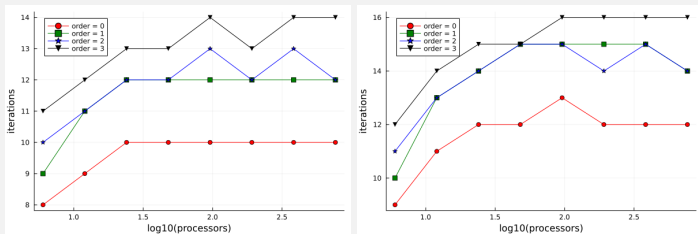


Figure 7: Number of CG iterations, as a function of the number of processors. Cases: Mixed-order HHO with $H/h = 8$ (left) and $H/h = 16$ (right).

Results 3D (I)

HDG of 3D structured meshes

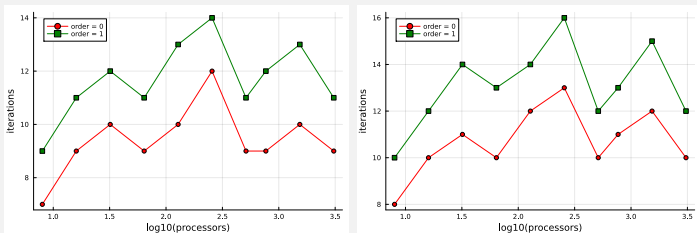


Figure 8: Number of CG iterations, as a function of the number of processors. Cases: HDG with $H/h = 4$ (left) and $H/h = 8$ (right).

Results 3D (II)

Mixed-order HHO of 3D structured meshes

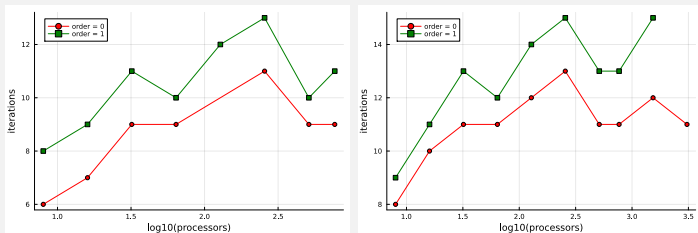


Figure 9: Number of CG iterations, as a function of the number of processors. Cases: Mixed-order HHO with $H/h = 4$ (left) and $H/h = 8$ (right).

Thank you!

S. Badia, J. Droniou, J. Manyer, J. Tushar, “Analysis of BDDC preconditioners for non-conforming polytopal hybrid discretisation methods”, preprint available.

S. Badia, J. Droniou, J. Tushar, “A discrete trace theory for non-conforming polytopal hybrid discretisation methods.”, preprint available.

J. Manyer, A. Martin, S. Badia “GridapSolvers: Scalable multiphysics finite element solvers in Julia”, Journal of Open Source Software, 2024.

▷ Get slides for **Part I** and **Part II** of this talk.



Computational resources for this work were provided by NCI through the NCMAS public scheme and the Monash-NCI partnership.

- [1] Santiago Badia, Jerome Droniou, Jordi Manyer, and Jai Tushar. *Analysis of BDDC preconditioners for non-conforming polytopal hybrid discretisation methods*. 2025. arXiv: 2506.11956 [math.NA].
- [2] Santiago Badia, Jérôme Droniou, and Jai Tushar. *A discrete trace theory for non-conforming polytopal hybrid discretisation methods*. 2024. arXiv: 2409.15863 [math.NA].
- [3] Santiago Badia, Alberto F. Martín, and Javier Principe. “A Highly Scalable Parallel Implementation of Balancing Domain Decomposition by Constraints”. In: *SIAM Journal on Scientific Computing* 36.2 (2014), pp. C190–C218. DOI: 10.1137/130931989. eprint: <https://doi.org/10.1137/130931989>.
- [4] Santiago Badia, Alberto F. Martín, and Francesc Verdugo. “GridapDistributed: a massively parallel finite element toolbox in Julia”. In: *Journal of Open Source Software* 7.74 (2022), p. 4157. DOI: 10.21105/joss.04157.

- [5] Santiago Badia and Francesc Verdugo. “Gridap: An extensible Finite Element toolbox in Julia”. In: *Journal of Open Source Software* 5.52 (2020), p. 2520. DOI: 10.21105/joss.02520.
- [6] Matteo Cicuttin, Alexandre Ern, and Nicolas Pignet. *Hybrid High-Order Methods: A Primer with Applications to Solid Mechanics*. Mathematical Engineering. Springer International Publishing, 2021. DOI: 10.1007/978-3-030-81477-9.
- [7] Bernardo Cockburn, Jayadeep Gopalakrishnan, and Raytcho Lazarov. “Unified Hybridization of Discontinuous Galerkin, Mixed, and Continuous Galerkin Methods for Second Order Elliptic Problems”. In: *SIAM Journal on Numerical Analysis* (2009). DOI: 10.1137/070706616.
- [8] Daniele A. Di Pietro and Alexandre Ern. “A hybrid high-order locking-free method for linear elasticity on general meshes”. In: *Computer Methods in Applied Mechanics and Engineering* 283 (2015), pp. 1–21. DOI: <https://doi.org/10.1016/j.cma.2014.09.009>.

- [9] Shukai Du and Francisco–Javier Sayas. “A note on devising HDG+ projections on polyhedral elements”. In: *arXiv: Numerical Analysis* (2020). DOI: 10.1090/mcom/3573.
- [10] Jan Mandel and Clark R. Dohrmann. “Convergence of a balancing domain decomposition by constraints and energy minimization”. In: vol. 10. 7. Dedicated to the 70th birthday of Ivo Marek. 2003, pp. 639–659. DOI: 10.1002/nla.341.
- [11] Jan Mandel, Bedřich Sousedík, and Clark R. Dohrmann. “Multispace and multilevel BDDC”. In: *Computing* 83.2–3 (Sept. 2008), pp. 55–85. DOI: 10.1007/s00607-008-0014-7.
- [12] Jordi Manyer, Alberto F. Martín, and Santiago Badia. “GridapSolvers.jl: Scalable multiphysics finite element solvers in Julia”. In: *Journal of Open Source Software* 9.102 (2024), p. 7162. DOI: 10.21105/joss.07162.

- [13] Daniele Antonio Di Pietro and Jérôme Droniou. *The Hybrid High-Order Method for Polytopal Meshes: Design, Analysis, and Applications*. Vol. 19. MS&A. Springer, 2020. DOI: 10.1007/978-3-030-37203-3.

Appendix A - Implementation details for BDDC (I)

Our preconditioner is given by

$$\hat{\mathcal{B}}_h = \mathcal{Q}_h \tilde{\mathcal{A}}_h^{-1} \mathcal{Q}_h^T + I_0 \mathcal{A}_{h,0}^{-1} I_0^T$$

Given a residual $\hat{r}_h$, it returns a correction $\hat{z}_h = \tilde{z}_h + z_{h,0}$ given by

$$\tilde{z}_h = \mathcal{Q}_h \tilde{\mathcal{A}}_h^{-1} \mathcal{Q}_h^T \hat{r}_h, \quad z_{h,0} = I_0 \mathcal{A}_{h,0}^{-1} I_0^T \hat{r}_h$$

We can further split the problem by considering $\underline{\tilde{U}}_h = \tilde{U}_{h,0} \oplus \tilde{U}_{h,c}$ where

- $\tilde{U}_{h,0}$ is the subset of $\underline{\tilde{U}}_h$ with zero coarse DoF values.
- $\tilde{U}_{h,c}$ is the complementary, i.e the subset of $\underline{\tilde{U}}_h$ with at least one non-zero coarse DoF value.

We can write $\tilde{z}_h = \mathcal{Q}_h(\tilde{z}_{h,0} + \tilde{z}_{h,c})$ with

$$\tilde{z}_{h,0} = \tilde{\mathcal{A}}_{h,0}^{-1} r_h, \quad \tilde{z}_{h,c} = \tilde{\mathcal{A}}_{h,c}^{-1} r_h, \quad r_h = \mathcal{Q}_h^T \hat{r}_h$$

Appendix A - Implementation details for BDDC (II)

Let A be the matrix representation of $\mathcal{A}_h$, i.e the sub-assembled system matrix. We have that

- A is block-diagonal, with blocks A^k corresponding to subdomains.
- Each block A^k can be locally split into interior/interface blocks, i.e

$$A^k = \begin{bmatrix} A_{II}^k & A_{I\Gamma}^k \\ A_{\Gamma I}^k & A_{\Gamma\Gamma}^k \end{bmatrix}$$

Also:

- Let C^k be the matrix representation of the coarse constraints, local to each processor.
- Let $R_I = \text{diag}(R_I^k)$ and $R_\Gamma = \text{diag}(R_\Gamma^k)$ be the interior and interface restrictions, respectively.
- Let W be the matrix representation of W_h .

Appendix A - Implementation details for BDDC (III)

Interior correction

- The block-diagonal matrix $A_{II} = \text{diag}(A_{II}^k)$ is the matrix representation of $\mathcal{A}_{h,0}$.
- The block-diagonal matrix $R_I = \text{diag}(R_I^k)$ is the matrix representation of I_0^T .

Then if we take $z_0, \hat{r}$ the vector representations of $z_{h,0}, \hat{r}_h$ respectively, we have

$$z_0 = R_I^T A_{II}^{-1} R_I \hat{r}$$

which can be solved locally on each processor.

Appendix A - Implementation details for BDDC (IV)

Fine correction

The fine correction

$$\tilde{z}_{h,0} = \tilde{\mathcal{A}}_{h,0}^{-1} r_h$$

can be computed locally by solving

$$\begin{bmatrix} A_{II}^k & A_{I\Gamma}^k & \\ A_{\Gamma I}^k & A_{\Gamma\Gamma}^k & (C^k)^T \\ & C^k & \end{bmatrix} \cdot \begin{bmatrix} \tilde{z}_{0,I}^k \\ \tilde{z}_{0,\Gamma}^k \\ \lambda^k \end{bmatrix} = \begin{bmatrix} r_I^k \\ r_\Gamma^k \\ 0 \end{bmatrix}$$

where $\tilde{z}_0, r$ the vector representations of $\tilde{z}_{h,0}, r_h$ respectively, and

$$\begin{bmatrix} \tilde{z}_{0,I}^k \\ \tilde{z}_{0,\Gamma}^k \end{bmatrix} = \begin{bmatrix} R_I^k \\ R_\Gamma^k \end{bmatrix} \cdot \tilde{z}_0^k, \quad \begin{bmatrix} r_I^k \\ r_\Gamma^k \end{bmatrix} = \begin{bmatrix} R_I^k \\ R_\Gamma^k \end{bmatrix} \cdot r^k$$

Appendix A - Implementation details for BDDC (V)

Coarse correction

We have $\tilde{U}_{h,c} = \text{span}\{\phi_\lambda\}_{\lambda \in \Lambda}$, where ϕ_λ is the global basis function that is one on λ and zero on all other coarse degrees of freedom. Denote by Φ the matrix that has columns ϕ_λ .

We can compute Φ^k , by solving the local problems

$$\begin{bmatrix} A_{II}^k & A_{I\Gamma}^k & (C^k)^T \\ A_{\Gamma I}^k & A_{\Gamma\Gamma}^k & C^k \end{bmatrix} \cdot \begin{bmatrix} \Phi_I^k \\ \Phi_\Gamma^k \\ \lambda^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ I_d \end{bmatrix}$$

The matrix representation of $\tilde{\mathcal{A}}_{h,c}$ is then given by

$$\tilde{A}_c^{-1} = \Phi S_c^{-1} \Phi^T, \quad S_c = \sum_k S_c^k = \sum_k (\Phi^k)^T A^k \Phi^k$$

Appendix A - Implementation details for BDDC (VI)

Coarse correction¹

Generally, S_c is assembled into a single processor and factorized. The coarse correction then involves:

1. gathering the local coarse residuals $r_c^k = (\Phi^k)^T r^k$ into a single processor,
2. solving the coarse problem, and
3. scattering the result back to each processor and doing a linear combination with the basis, to obtain $\tilde{z}_c^k$.

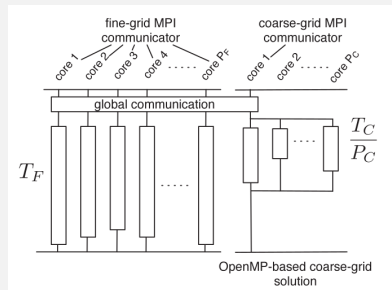


Figure 10: Processor layout for computing BDDC corrections.

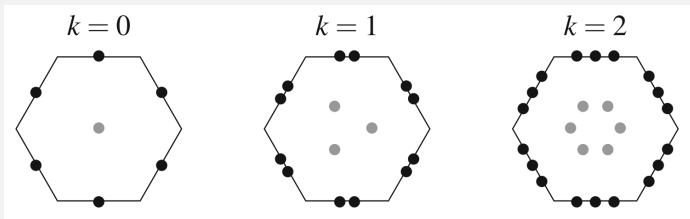
¹Santiago Badia, Alberto F. Martín, and Javier Principe. “A Highly Scalable Parallel Implementation of Balancing Domain Decomposition by Constraints”. In: *SIAM Journal on Scientific Computing* 36.2 (2014), pp. C190–C218. DOI: 10.1137/130931989. eprint: <https://doi.org/10.1137/130931989>.

Appendix B - HHO formulation for the Poisson problem (I)^{1,2}

Given a polynomial order k , we consider the **local** and **global** HHO spaces

$$\underline{U}_T^k = \{ (v_T, (v_F)_{F \in \mathcal{F}_T}) : v_T \in \mathbb{P}^k(T), v_F \in \mathbb{P}^k(F) \forall F \in \mathcal{F}_T \}$$

$$\underline{U}_h = \{ ((v_T)_{T \in \mathcal{T}_h}, (v_F)_{F \in \mathcal{F}_h}) : v_T \in \mathbb{P}^k(T) \forall T \in \mathcal{T}_h, v_F \in \mathbb{P}^k(F) \forall F \in \mathcal{F}_h \}$$



¹Daniele Antonio Di Pietro and Jérôme Droniou. *The Hybrid High-Order Method for Polytopal Meshes: Design, Analysis, and Applications*. Vol. 19. MS&A. Springer, 2020. DOI: 10.1007/978-3-030-37203-3.

²Matteo Cicuttin, Alexandre Ern, and Nicolas Pignet. *Hybrid High-Order Methods: A Primer with Applications to Solid Mechanics*. Mathematical Engineering. Springer International Publishing, 2021. DOI: 10.1007/978-3-030-81477-9.

Appendix B - HHO formulation for the Poisson problem (II)

Potential reconstruction operator

We define $R_T^{k+1} \underline{v}_T : \underline{U}_T^k \rightarrow \mathbb{P}^{k+1}(T)$ such that $\forall w \in \mathbb{P}^{k+1}(T)$ we have

$$\begin{aligned} (\nabla R_T^{k+1} \underline{v}_T, \nabla w)_T &= (\nabla v_T, \nabla w)_T + \sum_{F \in \mathcal{F}_T} (v_F - v_T, \nabla w \cdot \mathbf{n}_{TF})_F \\ (R_T^{k+1} \underline{v}_T, w)_T &= (v_T, w)_T \end{aligned}$$

Difference operators

We define $\delta_T^k \underline{v}_T : \underline{U}_T^k \rightarrow \mathbb{P}^k(T)$ and $\delta_F^k \underline{v}_F : \underline{U}_T^k \rightarrow \mathbb{P}^k(F)$ as

$$\delta_T^k \underline{v}_T = \pi_T^k(R_T^{k+1} \underline{v}_T - v_T), \quad \delta_F^k \underline{v}_T = \pi_F^k(R_T^{k+1} \underline{v}_T - v_F)$$

Appendix B - HHO formulation for the Poisson problem (III)

The global problem is then given by: Find $\underline{u}_h \in \underline{U}_{h,0}$ such that $\forall \underline{v}_h \in \underline{U}_{h,0}$

$$a_h(\underline{u}_T, \underline{v}_T) = b_h(\underline{v}_T)$$

with

$$a_h(\underline{u}_T, \underline{v}_T) = \sum_{T \in \mathcal{T}_h} a_T(\underline{u}_T, \underline{v}_T) , \quad b_h(\underline{v}_T) = \sum_{T \in \mathcal{T}_h} (f, v_T)_T ,$$

and where the local contributions are given by

$$\begin{aligned} a_T(\underline{u}_T, \underline{v}_T) &= (\nabla R_T^{k+1} \underline{u}_T, \nabla R_T^{k+1} \underline{v}_T)_T + s_T(\underline{u}_T, \underline{v}_T) \\ s_T(\underline{u}_T, \underline{v}_T) &= \sum_{F \in \mathcal{F}_T} h_F^{-1} [(\delta_{TF}^k - \delta_T^k) \underline{u}_T, (\delta_{TF}^k - \delta_T^k) \underline{v}_T] \end{aligned}$$