Deflation-based preconditioning for immersed finite element methods and immersogeometric analysis

Yannis Voet [1] Matthias Möller [2] Kees Vuik [2] Pablo Antolin [1]

[1]Chair of Modeling and Numerical Simulation, EPFL

[2]Numerical Analysis group, TU Delft

February 26, 2026

COSSE workshop, TU Berlin

Setting

Solve a large sparse linear system

$$A\boldsymbol{x} = \boldsymbol{b}$$

$A \in \mathbb{R}^{n \times n}$ is symmetric positive definite (SPD) and stems from a FEM discretization. Typical examples:

1. $A = K$ (Poisson problem, elliptic PDEs,...)
2. $A = M$ (L^2 projection, explicit time stepping,...)
3. $A = K + \gamma M$ with $\gamma > 0$ (Implicit time stepping)

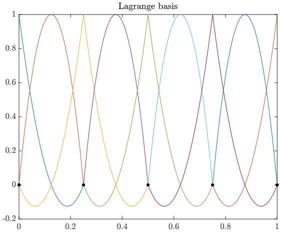
$$K_{ij} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \quad \text{(Stiffness matrix)} \qquad M_{ij} = \int_\Omega \varphi_i\varphi_j \quad \text{(Mass matrix)}$$

for a finite element basis $\Phi = \{\varphi_1, \ldots, \varphi_n\}$.

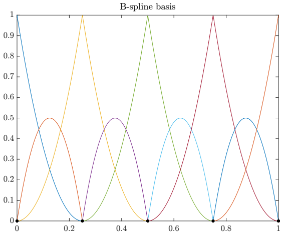
Finite element bases

The basis Φ may be:

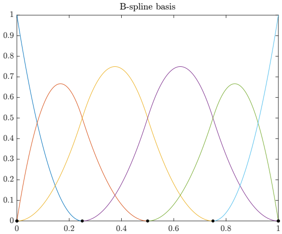
- $\Phi = \{L_1, \ldots, L_n\}$, the Lagrange basis (standard FEM)
- $\Phi = \{B_1, \ldots, B_n\}$, a spline basis, e.g. the B-spline basis (Isogeometric analysis). Spline spaces are C^k continuous with $0 \leq k \leq p-1$.



(a) Lagrange basis



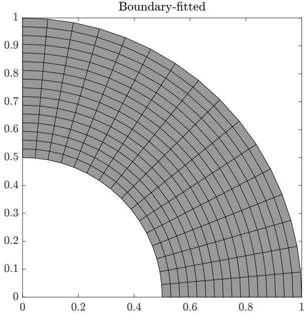
(b) C^0 B-spline basis

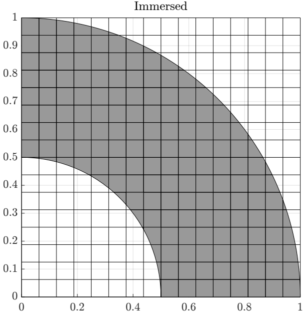


(c) C^1 B-spline basis

Boundary-fitted vs immersed finite element discretizations

How to solve a PDE on a complicated domain Ω. How can we build a finite element mesh?





+ Imposition of boundary conditions
+ Availability of h, p-robust preconditioners
– Mesh quality
– Flexibility, complexity

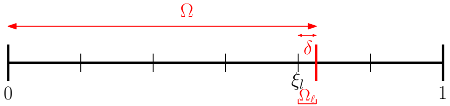
+ Flexibility
– Imposition of boundary conditions
– Integration on trimmed elements
– Stability, CFL condition
– Ill-conditioning

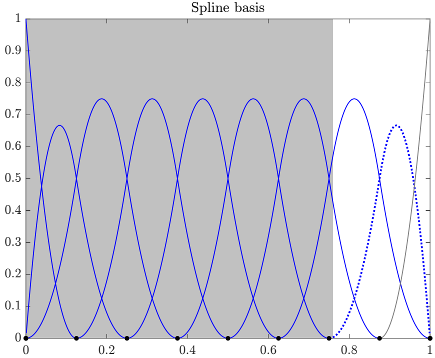
Source of ill-conditioning

Small cut elements cause extreme ill-conditioning; i.e. there exists a function $u \in V_h$ such that

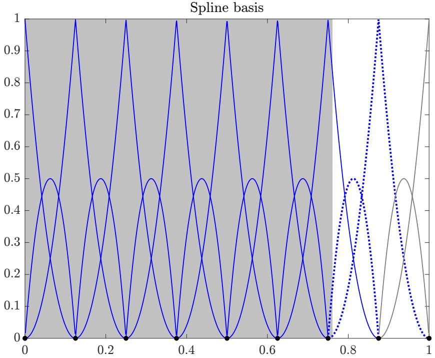
$$\|u\|_a^2 = a(u,u) \lesssim \epsilon \|\boldsymbol{u}\|_2^2 \implies \lambda_{\min}(A) \lesssim \epsilon.$$

Example: consider a quadratic spline discretization on $(0,1) \supset (0, 0.75 + \epsilon)$





(a) C^1 smoothness



(b) C^0 smoothness

Non-preconditioned case

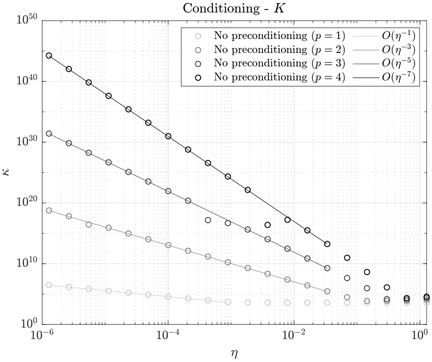
The problem is caused by basis functions whose active support $\mathrm{supp}_\Omega(\varphi) = \mathrm{supp}(\varphi_i) \cap \Omega$ only contains trimmed elements.

Under some shape regularity assumption, we can prove that (Prenter et al. 2017)

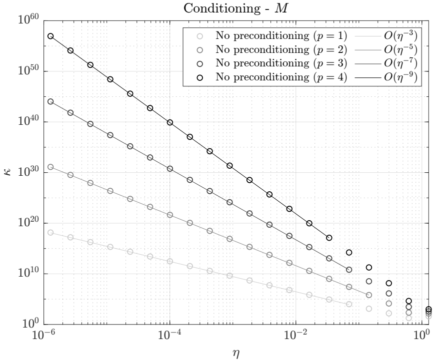
$$\kappa(K) \gtrsim \eta^{-(2p+1-2/d)}$$

$$\kappa(M) \gtrsim \eta^{-(2p+1)}$$

where $\eta = \min_{T\in\mathcal{T}_h} |T\cap\Omega|/|T|$ and $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$ is the spectral condition number (for an SPD matrix A).



(a) $\kappa(K)$



(b) $\kappa(M)$

Overview of solutions

Multiple solutions to the ill-conditioning issue

- Fictitious domain stabilization (or α-stabilization, finite cell method) (Parvizian et al. 2007) integrates the weak form over the entire extended domain by weighting the fictitious part by a small parameter α.
  - \+ Improves the conditioning
  - – to the detriment of accuracy...
- Polynomial extension techniques; e.g. extended B-splines (Höllig 2003; Marussig et al. 2017; Buffa et al. 2020)
  - \+ Solves the conditioning and stability issues
  - – Quite intrusive (modifies the approximation space)
- Eigenvalue stabilization (Eisenträger et al. 2024). Local modifications to the element matrices during assembly.
  - \+ Solves the conditioning issue
  - – Quite intrusive.
  - – Parameter-dependent. Bad choices may affect the accuracy.
- Preconditioning (Prenter et al. 2017; Prenter et al. 2019). Chooses a different basis for the approximation space.
  - \+ Operates on the solver, not the approximation space.
  - – Does not resolve stability issues.

Jacobi preconditioning

- In some cases, a Jacobi preconditioner is enough! Solve $\hat{A}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}$, where

$$\hat{A} = D^{-1}AD^{-1}, \quad \hat{\boldsymbol{x}} = D\boldsymbol{x}, \quad \text{and} \quad \hat{\boldsymbol{b}} = D^{-1}\boldsymbol{b}$$

  for $D = \sqrt{\mathrm{diag}(A)}$.
- In most cases, Jacobi preconditioning helps but is insufficient... The issue stems from near linear dependencies among rescaled basis functions.

Counter-example for the Lagrange basis in 1D

Define

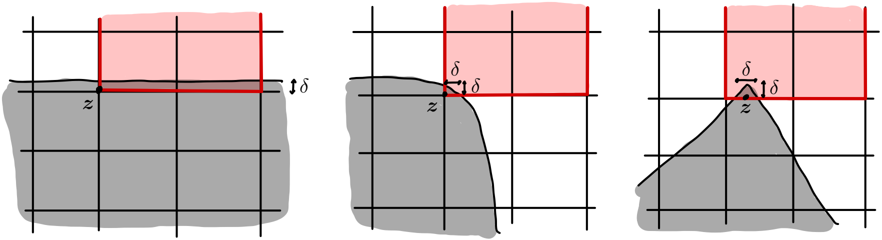
$$u(\xi) = \begin{cases} (\xi - \xi_l)^p & \text{if } \xi \in \Omega_\ell, \\ 0 & \text{otherwise.} \end{cases}$$

One can easily show that $u \in V_h$ and

$$\lambda_{\min}(\hat{A}) = \min_{\substack{\hat{\boldsymbol{v}} \in \mathbb{R}^n \\ \hat{\boldsymbol{v}} \neq 0}} \frac{\hat{\boldsymbol{v}}^T \hat{A} \hat{\boldsymbol{v}}}{\|\hat{\boldsymbol{v}}\|_2^2} \leq \frac{\hat{\boldsymbol{u}}^T \hat{A} \hat{\boldsymbol{u}}}{\|\hat{\boldsymbol{u}}\|_2^2} = \frac{\|u\|_a^2}{\|\hat{\boldsymbol{u}}\|_2^2} \sim \eta^{2(p-1)} \implies \kappa(\hat{A}) \gtrsim \eta^{-2(p-1)}.$$

Extension

The analysis in 2D is much more complicated and depends on the basis, continuity, position of the knots or interpolation points and cut configuration. Three representative cases:



Configuration (a) Configuration (b) Configuration (c)

Cut configuration	Lagrange basis	B-spline basis
1D	$\gtrsim \eta^{-2(p-1)}$	~ 1
2D - (a) ("sliver cut")	$\gtrsim \eta^{-2(p-1)}$	~ 1
2D - (b) ("corner cut")	$\gtrsim \eta^{-2(p-1)}$	~ 1
2D - (c) ("middle cut")	$\begin{cases} \gtrsim \eta^{-(2p-1)} & \text{if } z_1 \notin \mathcal{B}_{\hat{\xi}_{1,i}}(\delta) \\ \gtrsim \eta^{-2(p-1)} & \text{if } z_1 \in \mathcal{B}_{\hat{\xi}_{1,i}}(\delta) \end{cases}$	$\begin{cases} \gtrsim \eta^{-p} & \text{if } z_1 \notin \mathcal{B}_{\xi_{l_1}}(\delta) \\ \gtrsim \eta^{-k} & \text{if } z_1 \in \mathcal{B}_{\xi_{l_1}}(\delta) \end{cases}$

State-of-the-art (I)

- Prenter et al. 2017 proposed the Symmetric Incomplete Permuted Inverse Cholesky (SIPIC), constructed as follows:
  1. Initial Jacobi preconditioning
  2. Detect near linear dependencies by looking for the off-diagonal elements of $\hat{A}$ of magnitude ≈ 1
  3. Locally orthogonalize the functions identified
  4. Update the preconditioner and repeat the process until all off-diagonal elements are small enough (in magnitude).

– The detection procedure of SIPIC is sometimes insufficient for detecting near linear dependencies. Counter-example: the matrix

$$\begin{pmatrix} 1 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 \end{pmatrix}$$

is singular but does not have any off-diagonal entry close to 1 $(1/\sqrt{2} \approx 0.707)$.

State-of-the-art (II)

- Two years later, Prenter et al. 2019 suggested an additive Schwarz-type preconditioner by forming the approximate inverse preconditioner

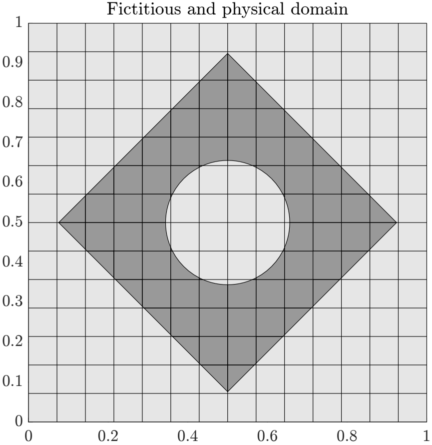
$$S = \sum_{i=1}^{N} P_i (P_i^T A P_i)^{-1} P_i^T,$$

where $P_i = [\boldsymbol{e}_{\mathcal{K}_i(1)}, \ldots, \boldsymbol{e}_{\mathcal{K}_i(m)}]$ contains the columns of the identity matrix for an index block $\mathcal{K}_i$ selected based on the geometry.

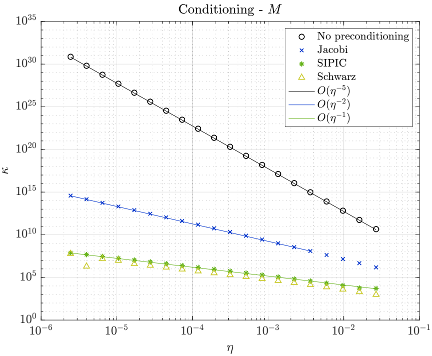
– The quality of the preconditioner critically depends on the choice of index blocks and none of the strategies proposed in the literature are entirely robust.

Counter-example

Geometry from Prenter et al. 2017



(a) Geometry



(b) Condition number

Deflation - Introduction

- The convergence of iterative methods applied to $A\boldsymbol{x} = \boldsymbol{b}$ is plagued by the small eigenvalues. Vuik et al. 1999
- Define a projector

$$P = I - AZ(Z^T AZ)^{-1} Z^T$$

  for a full-rank matrix $Z \in \mathbb{R}^{n\times r}$ with $r \ll n$ and solve the projected system $PA\tilde{\boldsymbol{x}} = P\boldsymbol{b}$ with e.g. Conjugate Gradients (CG).
- Recover the solution of $A\boldsymbol{x} = \boldsymbol{b}$ through the transformation

$$\boldsymbol{x} = Z(Z^T AZ)^{-1} Z^T \boldsymbol{b} + P^T \tilde{\boldsymbol{x}}.$$

Lemma

If A is SPD, then

1. $\ker(PA) = R(Z)$
2. PA *is symmetric positive semidefinite*

We define the *effective condition number* of PA

$$\kappa_{\text{eff}}(PA) = \frac{\lambda_n(PA)}{\lambda_{r+1}(PA)}.$$

Lemma

$$\kappa_{\text{eff}}(PA) \le \kappa(A).$$

Deflation - Construction

Let $(\lambda_k, \boldsymbol{v}_k)$ denote the eigenpairs of A.

Lemma

If $Z = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r]$ contains the r smallest eigenvectors of A, then

$$PA\boldsymbol{v}_k = \begin{cases} 0 & \text{for } k = 1, \ldots, r, \\ \lambda_k \boldsymbol{v}_k & \text{for } k = r+1, \ldots, n. \end{cases}$$

But computing eigenvectors is never practical, especially not in this setting.

How to construct Z? Recall that $\ker(PA) = R(Z)$. We actually don't need eigenvectors. We only need a basis for the smallest eigenspace.

- Define the set of cut elements

$$\mathcal{T}_C = \{T \in \mathcal{T}_h \colon |T \cap \Omega| < |T|\}$$

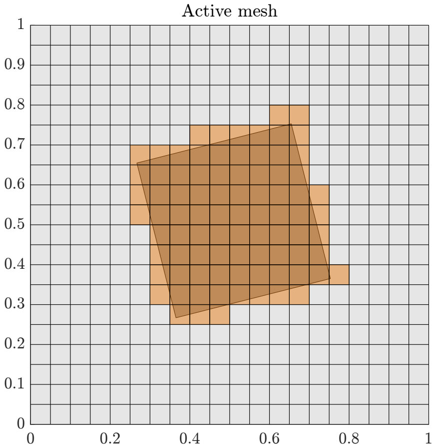
  and its complement $\mathcal{T}_C' = \mathcal{T}_h \setminus \mathcal{T}_C$, the set of uncut elements.
- Define the cut and uncut regions of the computational domain

$$\Omega_C = \bigcup_{T \in \mathcal{T}_C} T \quad \text{and} \quad \Omega_C' = \bigcup_{T \in \mathcal{T}_C'} T.$$

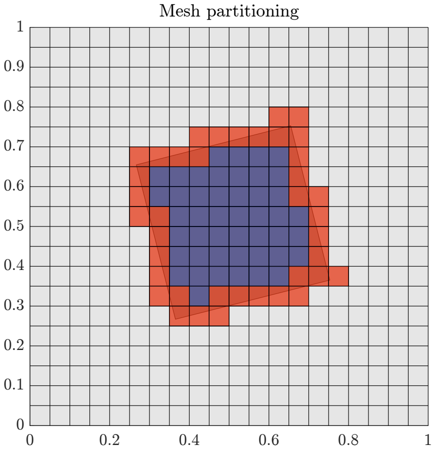
- The set of "cut basis functions" is

$$\Phi_C = \{\varphi \in \Phi \colon \operatorname{supp}_\Omega(\varphi) \subseteq \Omega_C\}.$$

Deflation - Construction



(a) Fictitious domain $\widehat{\Omega}$ (light gray), physical domain Ω (dark gray) and computational domain Ω_h (orange)



(b) Partitioning of the computational domain Ω_h in cut Ω_C (red) and uncut Ω'_C (blue) regions

For $Z = [e_{i_1}, \ldots, e_{i_r}]$ for indices i_k corresponding to functions in Φ_C.

Remarks

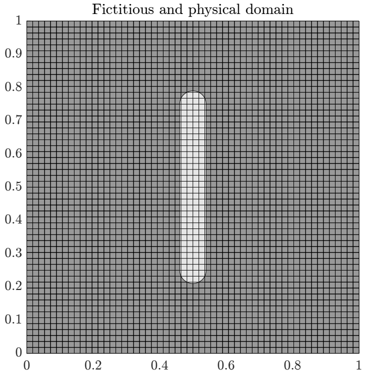
- Deflation is often combined with a standard SPD preconditioner M (e.g. incomplete Cholesky, Jacobi,...). Here we use Jacobi.
- It is sometimes possible to reduce Φ_C to only consider functions that cannot be cured by deflation.

Numerical experiment

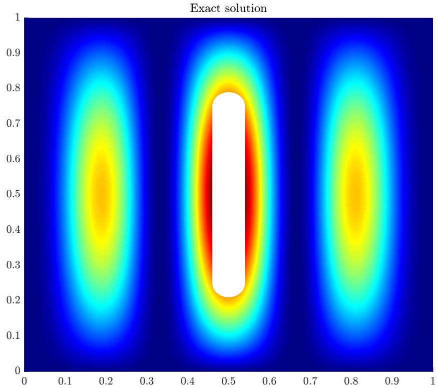
Solve the Poisson problem with the manufactured solution

$$u(x, y) = x(1 - x)\sin(3\pi x)^2 \sin(\pi y)$$

on a plate with a cut-out discretized with quadratic Lagrange polynomials. Solve a sequence of problems by slightly increasing the radius of the cut-out.



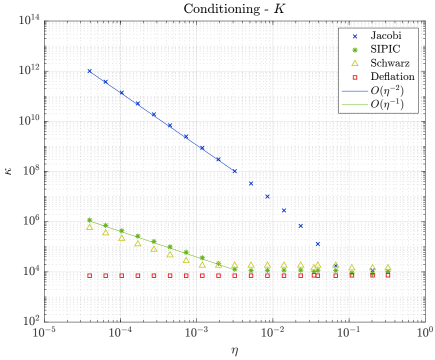
(a) Fictitious domain $\widehat{\Omega}$ (light gray) and physical domain Ω (dark gray)



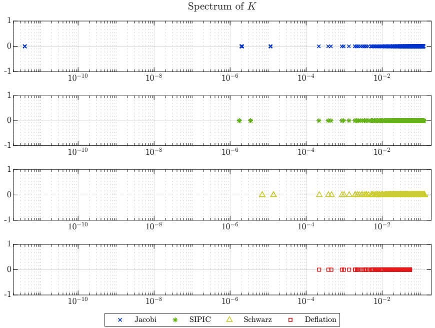
(b) Solution

Figure: Geometry and manufactured solution

Conditioning and spectrum



(a) Condition numbers for 20 logarithmically spaced values of ϵ between 10^{-4} and 10^{-2}

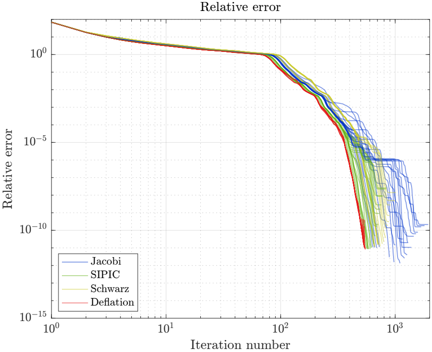


(b) First 500 eigenvalues of the preconditioned spectrum for $\epsilon = 10^{-4}$

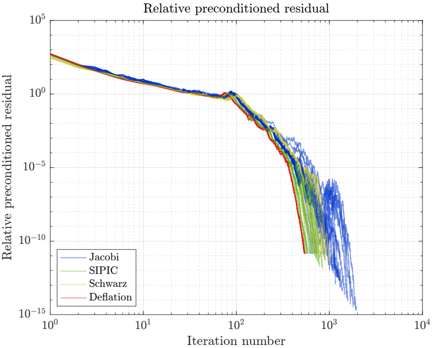
Figure: Condition numbers and eigenvalues of the preconditioned systems

Error and preconditioned residual

$$\frac{\|\boldsymbol{u} - \boldsymbol{u}_k\|_A}{\|\boldsymbol{u}\|_A} = \frac{|u_h - u_{h,k}|_{H^1}}{|u_h|_{H^1}} \quad \text{(Error)} \qquad \frac{\|\boldsymbol{r}_k\|_{M^{-1}}}{\|\boldsymbol{b}\|_{M^{-1}}} \quad \text{(Preconditioned residual)}$$



(a) Relative error



(b) Relative preconditioned residual

Figure: Convergence of the relative error and preconditioned residuals

Number of iterations

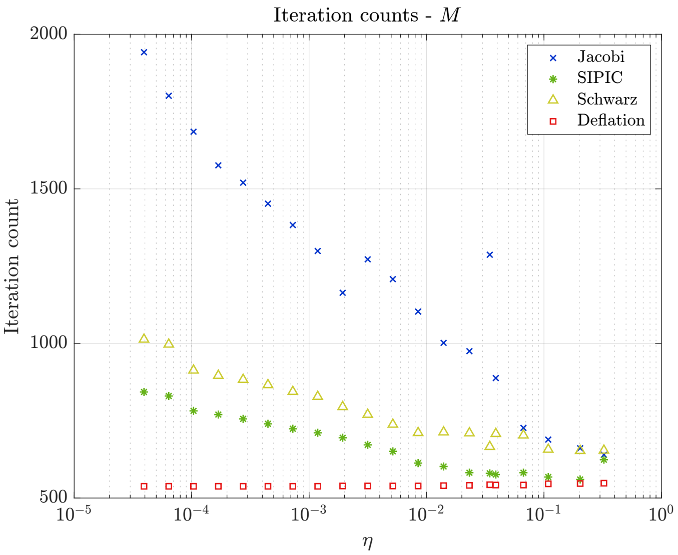


Figure: Number of iterations

Number of iterations

η	Jacobi	SIPIC	Schwarz	Deflation
3.24×10^{-1}	640	624	654	548
2.03×10^{-1}	661	560	653	547
1.09×10^{-1}	689	568	657	546
3.46×10^{-2}	1287	580	666	543
6.69×10^{-2}	727	582	703	542
3.89×10^{-2}	888	576	708	542
2.32×10^{-2}	975	582	710	541
1.40×10^{-2}	1002	602	713	540
8.48×10^{-3}	1103	613	711	539
5.17×10^{-3}	1208	651	738	539
3.16×10^{-3}	1272	672	770	539
1.93×10^{-3}	1164	695	795	539
1.18×10^{-3}	1299	711	828	538
7.30×10^{-4}	1383	724	844	538
4.50×10^{-4}	1452	740	866	538
2.70×10^{-4}	1520	756	883	538
1.70×10^{-4}	1576	770	896	538
1.00×10^{-4}	1685	782	913	538
6.00×10^{-5}	1801	830	997	538
4.00×10^{-5}	1942	843	1013	538

Table: Iteration counts

Conclusion

+ Deflation-based preconditioning perfectly resolves the ill-conditioning caused by badly cut elements.
– But must be combined with other h, p-robust preconditioners.
– Deflation remains a “global” strategy (contrary to Schwarz) and the deflation rank may become quite large for C^0 discretizations.

Future work: extension to indefinite and non-symmetric systems.

Thank you!

References I

[1] Annalisa Buffa, Riccardo Puppi, and Rafael Vázquez. "A minimal stabilization procedure for isogeometric methods on trimmed geometries". In: *SIAM Journal on Numerical Analysis* 58.5 (2020), pp. 2711–2735.

[2] Sascha Eisenträger et al. "An eigenvalue stabilization technique for immersed boundary finite element methods in explicit dynamics". In: *Computers & Mathematics with Applications* 166 (2024), pp. 129–168.

[3] Klaus Höllig. *Finite element methods with B-splines*. SIAM, 2003.

[4] Benjamin Marussig et al. "Stable isogeometric analysis of trimmed geometries". In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 497–521.

[5] Jamshid Parvizian, Alexander Düster, and Ernst Rank. "Finite cell method: h-and p-extension for embedded domain problems in solid mechanics". In: *Computational Mechanics* 41.1 (2007), pp. 121–133.

[6] Frits de Prenter, CV Verhoosel, and EH Van Brummelen. "Preconditioning immersed isogeometric finite element methods with application to flow problems". In: *Computer Methods in Applied Mechanics and Engineering* 348 (2019), pp. 604–631.

[7] Frits de Prenter et al. "Condition number analysis and preconditioning of the finite cell method". In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 297–327.

[8] C. Vuik, A. Segal, and J.A. Meijerink. "An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients". In: *Journal of Computational Physics* 152 (1999), pp. 385–403.