



MOX-Report No. 63/2023

**Iterative solution to the biharmonic equation in mixed form
discretized by the Hybrid High-Order method**

Antonietti, P. F.; Matalon, P.; Verani M.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it

<https://mox.polimi.it>

Iterative solution to the biharmonic equation in mixed form discretized by the Hybrid High-Order method*

P. F. Antonietti[†], P. Matalon^{†‡}, M. Verani[†]

Abstract

We consider the solution to the biharmonic equation in mixed form discretized by the Hybrid High-Order (HHO) methods. The two resulting second-order elliptic problems can be decoupled via the introduction of a new unknown, corresponding to the boundary value of the solution of the first Laplacian problem. This technique yields a global linear problem that can be solved iteratively via a Krylov-type method. More precisely, at each iteration of the scheme, two second-order elliptic problems have to be solved, and a normal derivative on the boundary has to be computed. In this work, we specialize this scheme for the HHO discretization. To this aim, an explicit technique to compute the discrete normal derivative of an HHO solution of a Laplacian problem is proposed. Moreover, we show that the resulting discrete scheme is well-posed. Finally, a new preconditioner is designed to speed up the convergence of the Krylov method. Numerical experiments assessing the performance of the proposed iterative algorithm on both two- and three-dimensional test cases are presented.

Keywords: partial differential equations, biharmonic equation, hybrid high-order.

1 Introduction

Let $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, be an open, bounded, polyhedral domain with smooth boundary $\partial\Omega$. In this work, we address the numerical approximation of the solution to the biharmonic equation

$$\Delta^2 \psi = \mathbf{f} \quad \text{in } \Omega, \tag{1a}$$

$$\psi = \mathbf{g}_D, \quad \partial_n \psi = \mathbf{g}_N \quad \text{on } \partial\Omega, \tag{1b}$$

where the load function $\mathbf{f} \in L^2(\Omega)$, $\mathbf{g}_D \in H^{1/2}(\partial\Omega)$ and $\mathbf{g}_N \in H^{-1/2}(\partial\Omega)$ are prescribed. As usual, for $X \subset \overline{\Omega}$ and $s \in \mathbb{R}$, we denote by $H^s(X)$ the standard Sobolev space of index s . For all $\mathbf{g}: \partial\Omega \rightarrow \mathbb{R}$, we define the subspace $H_{\mathbf{g}}^s(\Omega) := \{v \in H^s(\Omega) \mid v|_{\partial\Omega} = \mathbf{g}\}$. We denote by (\cdot, \cdot) (resp. $\langle \cdot, \cdot \rangle$) the standard L^2 -inner product in Ω (resp. on $\partial\Omega$). Moreover, $\partial_n \cdot$ denotes the outer normal derivative on $\partial\Omega$.

Equation (1) typically models the bending of a clamped plate supporting a load. By introducing the unknown $\omega := -\Delta\psi$, (1a) can be rewritten into two second-order elliptic equations, yielding the mixed formulation

$$\begin{aligned} -\Delta\omega &= \mathbf{f} && \text{in } \Omega, \\ -\Delta\psi &= \omega && \text{in } \Omega, \end{aligned} \tag{2}$$

$$\psi = \mathbf{g}_D, \quad \partial_n \psi = \mathbf{g}_N \quad \text{on } \partial\Omega.$$

This mixed form naturally arises in fluid dynamics, where ψ represents the stream and ω the vorticity. In the plate bending model, ψ represents the deflection and ω the bending moment or shear resultant force. There are several advantages in employing the mixed formulation (2) over the primal form (1). First, while the weak solution of the primal form (1) is to be found in $H^2(\Omega)$, that of the mixed one lies in $H^1(\Omega)$, for which approximation spaces are easier to be constructed. Second, the splitting (2) allows, after introducing an additional unknown, the use of fast, scalable solvers available for second-order elliptic equations. In particular, we will consider here the Hybrid High-Order (HHO) method [9, 10], a non-conforming polyhedral discretization allowing arbitrary polynomial degrees of approximation,

*This work has been partially funded by PRIN2017 research grant n. 2017KLJL. P.F.A. and M.V. also acknowledge the financial support of PRIN2020 research grant n. 20204LN5N5. P.F.A. and M.V. are members of INdAM-GNCS.

[†]MOX, Department of Mathematics, Politecnico di Milano, Italy (paola.antonietti@polimi.it, pierre.matalon@gmail.com, marco.verani@polimi.it)

[‡]Corresponding author

and exhibiting optimal convergence rates. In this context, the mixed form (2) allows to take advantage of the recent works on multigrid methods designed for the diffusion equation discretized by HHO schemes; see [11–14].

Recently, ad-hoc HHO discretizations of the primal formulation (1) have been designed in [17], and [16] generalizes the methods to other boundary conditions. See also [15] for HHO discretizations of singularly perturbed fourth-order problems. Previously, related models have also been tackled with HHO methods, such as the Kirchhoff–Love plate bending model problem in [2], and the Cahn–Hilliard equation in [4]. The mixed form (2) leads to a saddle-point algebraic system, for which special solvers and preconditioners have been proposed, e.g., in [20, 21, 25]. Nonetheless, in this form, the Laplacian equations are coupled, preventing the use of fast and scalable solvers specifically designed for symmetric positive-definite (SPD) matrices. To address this issue, techniques introducing new variables have been designed in order to transform (2) into a series of decoupled problems, whose associated algebraic formulations lead to symmetric and positive-definite algebraic problems. Among them, we mention the method proposed by Glowinski, Ciarlet, Raviart and Pironneau [5, 6, 19], where the authors introduce the unknown $\lambda := \omega|_{\partial\Omega}$. In that setting, one solves a sequence of Dirichlet problems and iteratively improves the solution as to enforce the prescribed Neumann condition. In a symmetric fashion, the technique proposed by Falk [18] introduces $\lambda := \partial_n \omega$, and one solves a sequence of Neumann problems while iteratively improving the solution as to enforce the prescribed Dirichlet condition. In both schemes, λ is the solution of a linear, symmetric, elliptic equation of the form $\mathcal{L}(\lambda) = b$, in which the evaluation of \mathcal{L} involves the solution of two Laplacian problems. In the discrete setting, solving this equation then corresponds to solving an *implicit* linear system, i.e. whose matrix is not explicitly known but one can compute its action on a vector. This specific configuration is well suited for the use of iterative methods where the operator is applied to a vector at each iteration, without requiring explicit knowledge of the matrix coefficients. Gradient descent algorithms and, more specifically, Krylov methods such as conjugate gradients are ideal candidates in this setting.

In this work, we focus on the approach of Glowinski *et al.* [5, 6, 19], that we recall in Section 2. The use of an iterative method for the solution of the global linear problem yields an iterative scheme where each iteration consists of three steps: (i) the solution of a Laplace problem; (ii) the solution of a subsequent Laplacian problem, using the solution of step (i) as a source term; (iii) the computation of the normal derivative on the boundary of the solution at step (ii). In Section 3, we recall the HHO discretization of the Laplacian problem with Dirichlet boundary conditions. Then, Section 4 proposes a computable algorithm for the discrete normal derivative of the HHO solution of a diffusion problem. This step is indeed the main ingredient to define the global discrete scheme.

The work of Glowinski *et al.* was applied to the standard Finite Element Method (FEM). While that setting does not raise any issue regarding the well-posedness of the discrete problem, as we will show in Section 5, in the context of HHO, the resulting problem requires to be stabilized. A stabilization method is proposed and the well-posedness of the discrete problem is proved.

In the context of two-dimensional FEMs, a preconditioner ensuring a convergence rate independent of the mesh size, was proposed in [22]. It is, however, restricted to two-dimensional problems. In Section 6, we therefore propose a novel preconditioner, applicable to general polytopal meshes. The main idea consists in building an approximate, sparse matrix of the problem, where each column j is computed by solving the two Laplacian problems and evaluating the normal derivative only in a restricted neighbourhood of the j^{th} degree of freedom (DoF). Finally, numerical experiments are reported in Section 7. Various types of two- and three-dimensional meshes, including polygonal meshes, are used. The scheme exhibits a convergence rate scaling as $\mathcal{O}(h^{k+2})$ in L^2 -norm, where k denotes the polynomial degree corresponding to the face unknowns of the HHO method.

2 The continuous splitting of the biharmonic problem

Following [5, 6, 19], we start from equation (2) and we introduce the new unknown $\lambda := \omega|_{\partial\Omega} \in H^{1/2}(\partial\Omega)$. Supposing λ known, ω and ψ are successively recovered by solving the following Dirichlet problems:

$$\begin{cases} -\Delta\omega = \mathbf{f} & \text{in } \Omega, \\ \omega = \lambda & \text{on } \partial\Omega, \end{cases} \quad (3a)$$

$$\begin{cases} -\Delta\psi = \omega & \text{in } \Omega, \\ \psi = \mathbf{g}_D & \text{on } \partial\Omega. \end{cases} \quad (3b)$$

While the Dirichlet datum \mathbf{g}_D is explicitly enforced on the solution in (3b), the enforcement of the Neumann condition $\partial_n \psi = \mathbf{g}_N$ defines a problem in the unknown λ , which is derived in the following way. For all $\mu \in H^{1/2}(\partial\Omega)$, we denote by $(\omega(\mu), \psi(\mu))$ the solution of

$$\begin{cases} -\Delta\omega(\mu) = \mathbf{f} & \text{in } \Omega, \\ \omega(\mu) = \mu & \text{on } \partial\Omega, \end{cases} \quad \begin{cases} -\Delta\psi(\mu) = \omega(\mu) & \text{in } \Omega, \\ \psi(\mu) = \mathbf{g}_D & \text{on } \partial\Omega. \end{cases}$$

By construction, the solution (ω, ψ) of (3) corresponds to $(\omega(\lambda), \psi(\lambda))$. Solving (3) then boils down to finding $\lambda \in H^{1/2}(\partial\Omega)$ such that

$$\partial_n \psi(\lambda) = \mathbf{g}_N. \quad (4)$$

In order to derive a linear problem from (4), the constraints related to \mathbf{f} and \mathbf{g}_D are eliminated through the introduction of $(\omega_0, \psi_0) := (\omega(0), \psi(0))$ (the choice of $\mu = 0$ is arbitrary). For all $\mu \in H^{1/2}(\partial\Omega)$, we then denote by $(\hat{\omega}(\mu), \hat{\psi}(\mu))$ the solution of the following sequence of problems, with vanishing load and Dirichlet function:

$$\begin{cases} -\Delta \hat{\omega}(\mu) = 0 & \text{in } \Omega, \\ \hat{\omega}(\mu) = \mu & \text{on } \partial\Omega, \end{cases} \quad \begin{cases} -\Delta \hat{\psi}(\mu) = \hat{\omega}(\mu) & \text{in } \Omega, \\ \hat{\psi}(\mu) = 0 & \text{on } \partial\Omega, \end{cases}$$

respectively. Equation (4) can now be reformulated as the linear problem

$$\mathcal{L}(\lambda) = b, \quad b := \partial_n \psi_0 - \mathbf{g}_N, \quad (6)$$

where $\mathcal{L}: H^{1/2}(\partial\Omega) \rightarrow H^{-1/2}(\partial\Omega)$ is the linear operator defined such that for all $\mu \in H^{1/2}(\partial\Omega)$,

$$\mathcal{L}(\mu) := -\partial_n \hat{\psi}(\mu). \quad (7)$$

The operator \mathcal{L} is proved to be continuous, symmetric and positive-definite in [19, Lem. 2.1].

3 HHO discretization of the Laplacian problem

In this section, we briefly recall (see, e.g., [8, Chap. 2] for extended details) the discrete HHO formulation of the following problem: find $u: \Omega \rightarrow \mathbb{R}$ such that

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = g_D & \text{on } \partial\Omega, \end{cases} \quad (8)$$

where $f: \Omega \rightarrow \mathbb{R}$ and $g_D \in H^{1/2}(\partial\Omega)$. A weak solution of (8) is obtained via the variational formulation: find $u \in H_{g_D}^1(\Omega)$ such that

$$a(u, v) = (f, v) \quad \forall v \in H_0^1(\Omega), \quad (9)$$

where the bilinear form a is such that $a(v, w) = (\nabla v, \nabla w)$, for all $v, w \in H^1(\Omega)$.

3.1 Mesh definition and notation

Let the couple $(\mathcal{T}_h, \mathcal{F}_h)$ define a mesh of the domain $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$: \mathcal{T}_h is a set of disjoint, open, polyhedral elements such that $\bigcup_{T \in \mathcal{T}_h} \bar{T} = \bar{\Omega}$; \mathcal{F}_h is the set of element faces; $h := \max_{T \in \mathcal{T}_h} h_T$ with h_T denoting the diameter of $T \in \mathcal{T}_h$. The mesh is assumed to match the geometrical requirements of [8, Def. 1.4]. Let \mathcal{T}_h^B the subset of \mathcal{T}_h collecting the elements located at the boundary of the domain. We also define the following subsets of \mathcal{F}_h :

- \mathcal{F}_h^I , collecting the interior faces;
- \mathcal{F}_h^B , collecting the boundary faces;
- \mathcal{F}_T , collecting the faces of T , for all $T \in \mathcal{T}_h$.

We denote by $\mathbf{n}_{\partial T}$ the unit normal vector to ∂T pointing outward of T . For all $T \in \mathcal{T}_h$ (resp. $F \in \mathcal{F}_h$), we denote by $(\cdot, \cdot)_T$ (resp. $\langle \cdot, \cdot \rangle_F$) the standard inner product of $L^2(T)$ (resp. $L^2(F)$) or $L^2(T)^d$. We define $(\cdot, \cdot)_{\mathcal{T}_h} := \sum_{T \in \mathcal{T}_h} (\cdot, \cdot)_T$, and for all $\mathcal{G}_h \subset \mathcal{F}_h$, $\langle \cdot, \cdot \rangle_{\mathcal{G}_h} := \sum_{F \in \mathcal{G}_h} \langle \cdot, \cdot \rangle_F$. Finally, we define $|\mathcal{T}_h| := \sum_{T \in \mathcal{T}_h} |T|$ and for all $\mathcal{G}_h \subset \mathcal{F}_h$, $|\mathcal{G}_h| := \sum_{F \in \mathcal{G}_h} |F|$, where $|\cdot|$ is the Hausdorff measure.

3.2 Local and broken polynomial spaces

The HHO method hinges on discrete unknowns representing polynomial functions local to elements and faces. So, for all $m \in \mathbb{N}_0$ and all $T \in \mathcal{T}_h$ (resp. $F \in \mathcal{F}_h$), we denote by $\mathbb{P}^m(T)$ (resp. $\mathbb{P}^m(F)$) the space spanned by the restriction to T (resp. F) of d -variate polynomials of total degree $\leq m$. From these local polynomial spaces, we can construct the following broken polynomial spaces supported by the mesh and its skeleton:

$$\begin{aligned} \mathbb{P}^m(\mathcal{T}_h) &:= \{v_{\mathcal{T}_h} := (v_T)_{T \in \mathcal{T}_h} \mid v_T \in \mathbb{P}^m(T) \ \forall T \in \mathcal{T}_h\}, \\ \mathbb{P}^m(\mathcal{F}_h) &:= \{v_{\mathcal{F}_h} := (v_F)_{F \in \mathcal{F}_h} \mid v_F \in \mathbb{P}^m(F) \ \forall F \in \mathcal{F}_h\}, \end{aligned}$$

respectively. The local space $\mathbb{P}^m(\mathcal{F}_T)$ is defined analogously for all $T \in \mathcal{T}_h$. For all cell or face X , we denote by $\pi_X^m: L^2(X) \rightarrow \mathbb{P}^m(X)$ the local L^2 -orthogonal projector onto the space $\mathbb{P}^m(X)$. By patching up those local projectors, we denote by $\pi_{\mathcal{T}_h}^m: L^2(\Omega) \rightarrow \mathbb{P}^m(\mathcal{T}_h)$ the piecewise L^2 -orthogonal projector onto $\mathbb{P}^m(\mathcal{T}_h)$, and by $\pi_{\mathcal{F}_h^B}^m: L^2(\partial\Omega) \rightarrow \mathbb{P}^m(\mathcal{F}_h^B)$ the piecewise L^2 -orthogonal projector onto $\mathbb{P}^m(\mathcal{F}_h^B)$.

3.3 Discrete hybrid formulation

Given the polynomial degrees $k \in \mathbb{N}_0$ and $l \in \{k, k+1\}$. The global and local spaces of *hybrid* variables are defined as

$$\begin{aligned}\underline{U}_h &:= \{ \underline{v}_h := (v_{\mathcal{T}_h}, v_{\mathcal{F}_h}) \in \mathbb{P}^l(\mathcal{T}_h) \times \mathbb{P}^k(\mathcal{F}_h) \}, \\ \underline{U}_T &:= \{ \underline{v}_T := (v_T, v_{\partial T}) \in \mathbb{P}^l(T) \times \mathbb{P}^k(\mathcal{F}_T) \} \quad \forall T \in \mathcal{T}_h,\end{aligned}$$

respectively. For any $\underline{v}_h \in \underline{U}_h$, we denote by $\underline{v}_T \in \underline{U}_T$ its restriction to $T \in \mathcal{T}_h$. Boundary data are strongly accounted for in the following subspaces:

$$\mathbb{P}^{k, \text{gD}}(\mathcal{F}_h) := \{ v_{\mathcal{F}_h} \in \mathbb{P}^k(\mathcal{F}_h) \mid v_F = \pi_F^k \text{gD} \quad \forall F \in \mathcal{F}_h^{\text{B}} \}, \quad \underline{U}_{h, \text{gD}} := \mathbb{P}^l(\mathcal{T}_h) \times \mathbb{P}^{k, \text{gD}}(\mathcal{F}_h).$$

In particular, homogeneous Dirichlet conditions are strongly enforced in $\underline{U}_{h,0}$. The global HHO bilinear form associated to the variational formulation of problem (8) is defined as $a_h: \underline{U}_h \times \underline{U}_h \rightarrow \mathbb{R}$ such that $a_h(\underline{v}_h, \underline{w}_h) := \sum_{T \in \mathcal{T}_h} a_T(\underline{v}_T, \underline{w}_T)$ where the local bilinear form $a_T: \underline{U}_T \times \underline{U}_T \rightarrow \mathbb{R}$ is defined as

$$a_T(\underline{v}_T, \underline{w}_T) := (\nabla p_T^{k+1} \underline{v}_T, \nabla p_T^{k+1} \underline{w}_T)_T + s_T(\underline{v}_T, \underline{w}_T) \quad \forall T \in \mathcal{T}_h, \quad (10)$$

In this expression, the first term is responsible for consistency while the second is required to ensure stability of the scheme. The consistency term involves the *local potential reconstruction operator* $p_T^{k+1}: \underline{U}_T \rightarrow \mathbb{P}^{k+1}(T)$ defined such that, for all $\underline{v}_T \in \underline{U}_T$, it satisfies

$$\begin{cases} (\nabla p_T^{k+1} \underline{v}_T, \nabla w)_T = -(v_T, \Delta w)_T + \langle v_{\partial T}, \nabla w \cdot \mathbf{n}_{\partial T} \rangle_{\partial T} & \forall w \in \mathbb{P}^{k+1}(T), \\ (p_T^{k+1} \underline{v}_T, 1)_T = (v_T, 1)_T. \end{cases} \quad (11a) \quad (11b)$$

Given the local interpolant $\underline{v}_T \in \underline{U}_T$ of a function $v \in L^2(\Omega)$, p_T^{k+1} reconstructs an approximation of v of degree $k+1$. The stabilization bilinear form s_T depends on its argument only through the *difference operators* $\delta_T: \underline{U}_T \rightarrow \mathbb{P}^l(T)$ and $\delta_{TF}: \underline{U}_T \rightarrow \mathbb{P}^k(F)$ for all $F \in \mathcal{F}_T$, defined such that, for all $\underline{v}_T \in \underline{U}_T$,

$$\delta_T \underline{v}_T := \pi_T^l(p_T^{k+1} \underline{v}_T - v_T) \text{ and } \delta_{TF} \underline{v}_T := \pi_F^k(p_T^{k+1} \underline{v}_T - v_F) \text{ for all } F \in \mathcal{F}_T. \quad (12)$$

These operators capture the higher-order correction that the reconstruction p_T^{k+1} adds to the element and face unknowns, respectively. A classical expression for $s_T: \underline{U}_T \times \underline{U}_T \rightarrow \mathbb{R}$ is

$$s_T(\underline{v}_T, \underline{w}_T) := \sum_{F \in \mathcal{F}_T} h_F^{-1} \mathfrak{s}_{TF}(\underline{v}_T, \underline{w}_T), \quad \mathfrak{s}_{TF}(\underline{v}_T, \underline{w}_T) := \langle (\delta_{TF} - \delta_T) \underline{v}_T, (\delta_{TF} - \delta_T) \underline{w}_T \rangle_F. \quad (13)$$

If $l = k+1$, one can also use the simpler formula (used, e.g., in [4])

$$\mathfrak{s}_{TF}(\underline{v}_T, \underline{w}_T) := \langle (\pi_F^k(v_T - v_F), \pi_F^k(w_T - w_F)) \rangle_F. \quad (14)$$

The global HHO problem reads: find $\underline{u}_h \in \underline{U}_{h, \text{gD}}$ such that

$$a_h(\underline{u}_h, \underline{v}_h) = (f, v_{\mathcal{T}_h})_{\mathcal{T}_h} \quad \forall \underline{v}_h \in \underline{U}_{h,0}. \quad (15)$$

The final approximation is obtained through the post-processing step

$$u_h := p_h^{k+1} \underline{u}_h \in \mathbb{P}^{k+1}(\mathcal{T}_h). \quad (16)$$

3.4 Local problems and cell unknown recovery operator

It follows from this local construction that the cell unknowns are only locally coupled. For all $T \in \mathcal{T}_h$, we define the linear operator $\Theta_T: \mathbb{P}^k(\mathcal{F}_T) \rightarrow \mathbb{P}^l(T)$ such that for all $v_{\partial T} \in \mathbb{P}^k(\mathcal{F}_T)$, $\Theta_T v_{\partial T}$ is the unique solution of the local problem

$$a_T((\Theta_T v_{\partial T}, 0), (w_T, 0)) = -a_T((0, v_{\partial T}), (w_T, 0)) \quad \forall w_T \in \mathbb{P}^l(T). \quad (17)$$

We denote by Θ_T the *cell unknown recovery operator*. In order to shorten the notation of the hybrid couple $\underline{v}_T := (\Theta_{\mathcal{T}_h \mathcal{F}_h} v_{\partial T}, v_{\partial T})$, we define the operator $\underline{\Theta}_T: \mathbb{P}^k(\mathcal{F}_T) \rightarrow \underline{U}_T$ such that for all $v_{\partial T}$,

$$\underline{\Theta}_T v_{\partial T} := (\Theta_T v_{\partial T}, v_{\partial T}). \quad (18)$$

Finally, the associated global operators $\Theta_{\mathcal{T}_h \mathcal{F}_h}: \mathbb{P}^k(\mathcal{F}_h) \rightarrow \mathbb{P}^l(\mathcal{T}_h)$ and $\underline{\Theta}_h: \mathbb{P}^k(\mathcal{F}_h) \rightarrow \underline{U}_h$ are defined locally such that for all $T \in \mathcal{T}_h$:

$$\begin{aligned}(\Theta_{\mathcal{T}_h \mathcal{F}_h} v_{\mathcal{F}_h})|_T &:= \Theta_T v_{\partial T} & \forall v_{\mathcal{F}_h} \in \mathbb{P}^k(\mathcal{F}_h), \\ (\underline{\Theta}_h v_{\mathcal{F}_h})|_{T \times \partial T} &:= \underline{\Theta}_T v_{\partial T} & \forall v_{\mathcal{F}_h} \in \mathbb{P}^k(\mathcal{F}_h).\end{aligned} \quad (19)$$

$\underline{\Theta}_h$ verifies the following useful property:

Lemma 1. For all $v_{\mathcal{F}_h} \in \mathbb{P}^k(\mathcal{F}_h)$ and $\underline{w}_h \in \underline{U}_h$, it holds that

$$a_h(\underline{\Theta}_h v_{\mathcal{F}_h}, \underline{w}_h) = a_h(\underline{\Theta}_h v_{\mathcal{F}_h}, \underline{\Theta}_h w_{\mathcal{F}_h}). \quad (20)$$

Proof. See [Appendix A](#). \square

Remark 1. (Static condensation) In practice, problem (15) is solved through the equivalent, condensed formulation: find $u_{\mathcal{F}_h} \in \mathbb{P}^{k, \text{gD}}(\mathcal{F}_h)$ such that

$$\widehat{a}_h(u_{\mathcal{F}_h}, v_{\mathcal{F}_h}) = (f, \Theta_{\mathcal{T}_h \mathcal{F}_h} v_{\mathcal{F}_h})_{\mathcal{T}_h} \quad \forall v_{\mathcal{F}_h} \in \mathbb{P}^{k, 0}(\mathcal{F}_h), \quad (21)$$

where $\widehat{a}_h: \mathbb{P}^k(\mathcal{F}_h) \times \mathbb{P}^k(\mathcal{F}_h) \rightarrow \mathbb{R}$ is such that for all $v_{\mathcal{F}_h}, w_{\mathcal{F}_h} \in \mathbb{P}^k(\mathcal{F}_h)$,

$$\widehat{a}_h(v_{\mathcal{F}_h}, w_{\mathcal{F}_h}) := a_h(\underline{\Theta}_h v_{\mathcal{F}_h}, \underline{\Theta}_h w_{\mathcal{F}_h}). \quad (22)$$

Refer to [7, Prop. 4] for the proof.

4 Discrete normal derivative

In this section, we derive an approximation, in the HHO context, of the normal derivative on $\partial\Omega$. This formula will play a crucial role in the approximation scheme of the biharmonic problem.

Let u be the solution of the boundary value problem (8) and let $\mathcal{H}: H^{1/2}(\partial\Omega) \rightarrow H^1(\Omega)$ be a linear operator such that for all $v \in H^{1/2}(\partial\Omega)$ defined on the boundary, $\mathcal{H}v$ extends v in the interior of Ω . By Green's formula, it holds that

$$\langle \partial_n u, v \rangle = (\nabla u, \nabla \mathcal{H}v) + (\Delta u, \mathcal{H}v) \quad \forall v \in H^{1/2}(\partial\Omega).$$

Using the bilinear form $a(\cdot, \cdot)$, and given that $-\Delta u = f$, the above equation becomes

$$\langle \partial_n u, v \rangle = a(u, \mathcal{H}v) - (f, \mathcal{H}v) \quad \forall v \in H^{1/2}(\partial\Omega). \quad (23)$$

In the literature, the operator $g_{\text{D}} \mapsto \partial_n u$ is called Poincaré-Steklov operator, or Dirichlet-to-Neumann map; see, e.g. [24]. Equation (23) is the well-known variational formula for the computation of the normal derivative. Now, given the solution \underline{u}_h of the discrete problem (15), we define its normal derivative on the boundary faces, denoted by $\partial_{n,h}(\underline{u}_h) \in \mathbb{P}^k(\mathcal{F}_h^{\text{B}})$, through a discrete counterpart of (23) in the HHO setting. Namely, $\partial_{n,h}(\underline{u}_h)$ verifies

$$\langle \partial_{n,h}(\underline{u}_h), v_{\mathcal{F}_h^{\text{B}}} \rangle_{\mathcal{F}_h^{\text{B}}} = a_h(\underline{u}_h, \underline{\mathcal{H}}_h v_{\mathcal{F}_h^{\text{B}}}) - (f, \mathcal{H}_{\mathcal{T}_h} v_{\mathcal{F}_h^{\text{B}}})_{\mathcal{T}_h} \quad \forall v_{\mathcal{F}_h^{\text{B}}} \in \mathbb{P}^k(\mathcal{F}_h^{\text{B}}), \quad (24)$$

where $\underline{\mathcal{H}}_h$ is an extension/lifting operator expressed in the hybrid setting, i.e.

$$\underline{\mathcal{H}}_h := (\mathcal{H}_{\mathcal{T}_h}, \mathcal{H}_{\mathcal{F}_h}), \quad \mathcal{H}_{\mathcal{T}_h}: \mathbb{P}^k(\mathcal{F}_h^{\text{B}}) \rightarrow \mathbb{P}^l(\mathcal{T}_h), \quad \mathcal{H}_{\mathcal{F}_h}: \mathbb{P}^k(\mathcal{F}_h^{\text{B}}) \rightarrow \mathbb{P}^k(\mathcal{F}_h).$$

For all $v_{\mathcal{F}_h^{\text{B}}} \in \mathbb{P}^k(\mathcal{F}_h^{\text{B}})$ and $F \in \mathcal{F}_h$, we define

$$(\mathcal{H}_{\mathcal{F}_h} v_{\mathcal{F}_h^{\text{B}}})|_F := \begin{cases} (v_{\mathcal{F}_h^{\text{B}}})|_F & \text{if } F \in \mathcal{F}_h^{\text{B}}, \\ 0 & \text{otherwise.} \end{cases}$$

Then, we set

$$\mathcal{H}_{\mathcal{T}_h} := \Theta_{\mathcal{T}_h \mathcal{F}_h} \mathcal{H}_{\mathcal{F}_h},$$

where $\Theta_{\mathcal{T}_h \mathcal{F}_h}$ is defined as in (19).

Remark 2. (Condensed formula) In order to facilitate the practical computation of $\partial_{n,h}(\underline{u}_h)$, equation (24) can be rewritten in terms of the condensed bilinear form \widehat{a}_h (cf. (22)). First of all, it follows from the definition of $\underline{\Theta}_h$ that $\underline{\mathcal{H}}_h = \underline{\Theta}_h \mathcal{H}_{\mathcal{F}_h}$. Then, by using property (20) and the definition (22) of \widehat{a}_h , we have

$$a_h(\underline{u}_h, \underline{\mathcal{H}}_h v_{\mathcal{F}_h^{\text{B}}}) = a_h(\underline{u}_h, \underline{\Theta}_h \mathcal{H}_{\mathcal{F}_h} v_{\mathcal{F}_h^{\text{B}}}) = a_h(\underline{\Theta}_h u_{\mathcal{F}_h}, \underline{\Theta}_h \mathcal{H}_{\mathcal{F}_h} v_{\mathcal{F}_h^{\text{B}}}) = \widehat{a}_h(u_{\mathcal{F}_h}, \mathcal{H}_{\mathcal{F}_h} v_{\mathcal{F}_h^{\text{B}}}).$$

Equation (24) then becomes

$$\langle \partial_{n,h}(\underline{u}_h), v_{\mathcal{F}_h^{\text{B}}} \rangle_{\mathcal{F}_h^{\text{B}}} = \widehat{a}_h(u_{\mathcal{F}_h}, \mathcal{H}_{\mathcal{F}_h} v_{\mathcal{F}_h^{\text{B}}}) - (f, \mathcal{H}_{\mathcal{T}_h} v_{\mathcal{F}_h^{\text{B}}})_{\mathcal{T}_h} \quad \forall v_{\mathcal{F}_h^{\text{B}}} \in \mathbb{P}^k(\mathcal{F}_h^{\text{B}}). \quad (25)$$

Notice that compared to (24), which requires the knowledge of \underline{u}_h , formula (25) only involves $u_{\mathcal{F}_h}$. Consequently, the following hold: (i) $u_{\mathcal{T}_h}$ does not have to be computed, (ii) the matrix used to compute the first term is smaller, as the matrix representation of \widehat{a}_h is a Schur complement where the cell unknowns have been eliminated.

The computation of the discrete normal derivative as proposed here is numerically validated in [Section 7.2](#). The experiments show a convergence $\mathcal{O}(h^{k+1})$ in L^2 -norm.

5 Discrete HHO problem

In this section, the global, discrete realization of problem (6) is formulated.

5.1 Bilinear form

In order to devise a stable problem, we will make use of the following bilinear form in the hybrid space \underline{U}_h . For all $\underline{v}_h, \underline{w}_h \in \underline{U}_h$, we introduce

$$(\underline{v}_h, \underline{w}_h)_{\mathcal{T}_h}^* := (v_{\mathcal{T}_h}, w_{\mathcal{T}_h})_{\mathcal{T}_h} + \sum_{T \in \mathcal{T}_h^B} \sum_{F \in \mathcal{F}_T} h_F \mathfrak{s}_{TF}(v_T, \underline{w}_T). \quad (26)$$

Notice that (26) defines an inner product-like bilinear form based on the L^2 -inner product of the cell unknowns, to which a stabilizing term has been added. The stabilizing term is inspired from [4, Eq. (26)]. Notwithstanding the exponent of the h_F factor, the latter reproduces, locally in T , the stabilization term (13) of the Laplacian bilinear form. The scaling factor h_F is selected to ensure dimensional homogeneity with the consistency term it is added to. Note that $(\cdot, \cdot)_{\mathcal{T}_h}^*$ does not define an inner product in \underline{U}_h , due to the fact that only the boundary cells are involved in the stabilizing term. In spite of that, we allow ourselves the use of an inner product notation, since $(\cdot, \cdot)_{\mathcal{T}_h}^*$ shall undertake the role of the L^2 -inner product in the discrete problem and, thus, carries the same semantics.

As the stabilizing term of (26) is based on the same local bilinear form \mathfrak{s}_{TF} used to stabilize the bilinear form a_h , it naturally confers to $(\cdot, \cdot)_{\mathcal{T}_h}^*$ some useful properties. First, $(\cdot, \cdot)_{\mathcal{T}_h}^*$ remains symmetric and positive semi-definite. Second, if definiteness is not globally ensured, it is ensured “locally” in the cells where the stabilization term is applied. One might say that $(\cdot, \cdot)_{\mathcal{T}_h}^*$ enjoys a property of *local stability* in the boundary cells, in the sense that

$$(\underline{v}_h, \underline{v}_h)_{\mathcal{T}_h}^* = 0 \quad \implies \quad v_T = 0 \quad \forall T \in \mathcal{T}_h^B. \quad (27)$$

Let us now describe the discrete, variational form of the continuous operator (7). For all $\mu \in \mathbb{P}^k(\mathcal{F}_h^B)$, we denote by $(\hat{\omega}_h^\mu, \hat{\psi}_h^\mu) \in \underline{U}_{h,\mu} \times \underline{U}_{h,0}$ the solution of the discrete problems

$$a_h(\hat{\omega}_h^\mu, \underline{v}_h) = 0 \quad \forall \underline{v}_h \in \underline{U}_{h,0}, \quad (28a)$$

$$a_h(\hat{\psi}_h^\mu, \underline{v}_h) = (\hat{\omega}_h^\mu, \underline{v}_h)_{\mathcal{T}_h}^* \quad \forall \underline{v}_h \in \underline{U}_{h,0}. \quad (28b)$$

For $\mu, \eta \in \mathbb{P}^k(\mathcal{F}_h^B)$, and $(\hat{\omega}_h^\mu, \hat{\psi}_h^\mu)$ the solution of (28) associated with μ , we define the bilinear form $\ell_h: \mathbb{P}^k(\mathcal{F}_h^B) \times \mathbb{P}^k(\mathcal{F}_h^B) \rightarrow \mathbb{R}$ as the computation of $-\partial_{n,h}(\hat{\psi}_h^\mu)$ via formula (24), such that

$$\ell_h(\mu, \eta) := -a_h(\hat{\psi}_h^\mu, \underline{\mathcal{H}}_h \eta) + (\hat{\omega}_h^\mu, \underline{\mathcal{H}}_h \eta)_{\mathcal{T}_h}^*. \quad (29)$$

Remark that we have used $(\cdot, \cdot)_{\mathcal{T}_h}^*$ in (28b) and (29) instead of $(\cdot, \cdot)_{\mathcal{T}_h}$ in the reference formulas (15) and (24), respectively. Given that formula (29) does not explicitly exhibit symmetry or positive-definiteness, we prove an equivalent reformulation, easier to analyze.

Lemma 2. (Reformulation of ℓ_h) For μ and $\eta \in \mathbb{P}^k(\mathcal{F}_h^B)$, denote by $\hat{\omega}_h^\mu$ and $\hat{\omega}_h^\eta$ their associated solutions of (28a), respectively. Then

$$\ell_h(\mu, \eta) = (\hat{\omega}_h^\mu, \hat{\omega}_h^\eta)_{\mathcal{T}_h}^*. \quad (30)$$

Proof. For μ and $\eta \in \mathbb{P}^k(\mathcal{F}_h^B)$, let $(\hat{\omega}_h^\mu, \hat{\psi}_h^\mu)$ and $(\hat{\omega}_h^\eta, \hat{\psi}_h^\eta)$ be their respective, associated solutions of (28). Let us start with the definition (29) of ℓ_h :

$$\ell_h(\mu, \eta) := -a_h(\hat{\psi}_h^\mu, \underline{\mathcal{H}}_h \eta) + (\hat{\omega}_h^\mu, \underline{\mathcal{H}}_h \eta)_{\mathcal{T}_h}^*. \quad (31)$$

By writing $\underline{\mathcal{H}}_h \eta = (\underline{\mathcal{H}}_h \eta - \hat{\omega}_h^\eta) + \hat{\omega}_h^\eta$, the first term becomes

$$a_h(\hat{\psi}_h^\mu, \underline{\mathcal{H}}_h \eta) = a_h(\hat{\psi}_h^\mu, \underline{\mathcal{H}}_h \eta - \hat{\omega}_h^\eta) + a_h(\hat{\psi}_h^\mu, \hat{\omega}_h^\eta). \quad (32)$$

As $\underline{\mathcal{H}}_h$ does not modify the boundary face unknowns, we have $\underline{\mathcal{H}}_h \eta \in \underline{U}_{h,\eta}$. Additionally, $\hat{\omega}_h^\eta \in \underline{U}_{h,\eta}$ by definition, so the difference $(\underline{\mathcal{H}}_h \eta - \hat{\omega}_h^\eta) \in \underline{U}_{h,0}$. Consequently, as $\hat{\psi}_h^\mu$ verifies (28b), it holds that

$$a_h(\hat{\psi}_h^\mu, \underline{\mathcal{H}}_h \eta - \hat{\omega}_h^\eta) = (\hat{\omega}_h^\mu, \underline{\mathcal{H}}_h \eta - \hat{\omega}_h^\eta)_{\mathcal{T}_h}^* = (\hat{\omega}_h^\mu, \underline{\mathcal{H}}_h \eta)_{\mathcal{T}_h}^* - (\hat{\omega}_h^\mu, \hat{\omega}_h^\eta)_{\mathcal{T}_h}^*. \quad (33)$$

For the second term of (32), since $\hat{\omega}_h^\eta$ is solution of (28a) and $\hat{\psi}_h^\mu \in \underline{U}_{h,0}$, we use $\hat{\psi}_h^\mu$ as test function in (28a) to show that

$$a_h(\hat{\psi}_h^\mu, \hat{\omega}_h^\eta) = 0. \quad (34)$$

Now, plugging (33) and (34) into (32) gives

$$a_h(\underline{\psi}_h^\mu, \underline{\mathcal{H}}_h \eta) = (\underline{\dot{\omega}}_h^\mu, \underline{\mathcal{H}}_h \eta)_{\mathcal{T}_h}^* - (\underline{\dot{\omega}}_h^\mu, \underline{\dot{\omega}}_h^\eta)_{\mathcal{T}_h}^*. \quad (35)$$

Finally, plugging (35) into (31) yields (30). \square

Using formulation (30), we clearly have

Theorem 1. ℓ_h , defined by (29), is symmetric and positive-definite.

Proof. Formulation (30) clearly shows symmetry. Additionally, for all $\mu \in \mathbb{P}^k(\mathcal{F}_h^B)$, we have

$$\ell_h(\mu, \mu) = (\underline{\dot{\omega}}_h^\mu, \underline{\dot{\omega}}_h^\mu)_{\mathcal{T}_h}^*. \quad (36)$$

Since $(\cdot, \cdot)_{\mathcal{T}_h}^*$ is positive semi-definite, so is ℓ_h . Now, assume $\ell_h(\mu, \mu) = 0$, which, by (36), means $(\underline{\dot{\omega}}_h^\mu, \underline{\dot{\omega}}_h^\mu)_{\mathcal{T}_h}^* = 0$. The property of local stability (27) implies that, for all $T \in \mathcal{T}_h^B$, we have $\underline{\dot{\omega}}_T^\mu = 0$, i.e. $\dot{\omega}_T^\mu = 0$ and $\dot{\omega}_{\partial T}^\mu = 0$. In particular, for all $F \in \mathcal{F}_h^B$, $\mu_F = \dot{\omega}_F^\mu = 0$. This being true for all $F \in \mathcal{F}_h^B$, we have $\mu = 0$, which proves that ℓ_h is positive-definite. \square

5.2 Discrete scheme

Let $(\underline{\omega}_h^0, \underline{\psi}_h^0) \in \underline{U}_{h,0} \times \underline{U}_{h,\text{gD}}$ be the solution of

$$a_h(\underline{\omega}_h^0, \underline{v}_h) = (\mathbf{f}, v_{\mathcal{T}_h})_{\mathcal{T}_h} \quad \forall \underline{v}_h \in \underline{U}_{h,0}, \quad (37a)$$

$$a_h(\underline{\psi}_h^0, \underline{v}_h) = (\omega_{\mathcal{T}_h}^0, v_{\mathcal{T}_h})_{\mathcal{T}_h} \quad \forall \underline{v}_h \in \underline{U}_{h,0}. \quad (37b)$$

The discrete formulation of problem (6) reads: find $\lambda_{\mathcal{F}_h^B} \in \mathbb{P}^k(\mathcal{F}_h^B)$ such that

$$\ell_h(\lambda_{\mathcal{F}_h^B}, \mu) = \langle \partial_{n,h}(\underline{\psi}_h^0) - \mathbf{g}_N, \mu \rangle_{\mathcal{F}_h^B} \quad \forall \mu \in \mathbb{P}^k(\mathcal{F}_h^B). \quad (38)$$

We also define the linear operator $\mathcal{L}_h: \mathbb{P}^k(\mathcal{F}_h^B) \rightarrow \mathbb{P}^k(\mathcal{F}_h^B)$ associated to the bilinear form ℓ_h such that, for all $\mu \in \mathbb{P}^k(\mathcal{F}_h^B)$,

$$\langle \mathcal{L}_h \mu, \eta \rangle_{\mathcal{F}_h} := \ell_h(\mu, \eta) \quad \forall \eta \in \mathbb{P}^k(\mathcal{F}_h^B). \quad (39)$$

Since ℓ_h is positive-definite, problem (38) is well-posed. Its solution exists and is unique. Once it is solved, we compute $(\underline{\omega}_h, \underline{\psi}_h) \in \underline{U}_{h,\lambda_{\mathcal{F}_h^B}} \times \underline{U}_{h,\text{gD}}$, solution of

$$a_h(\underline{\omega}_h, \underline{v}_h) = (\mathbf{f}, v_{\mathcal{T}_h})_{\mathcal{T}_h} \quad \forall \underline{v}_h \in \underline{U}_{h,0}, \quad (40a)$$

$$a_h(\underline{\psi}_h, \underline{v}_h) = (\omega_{\mathcal{T}_h}, v_{\mathcal{T}_h})_{\mathcal{T}_h} \quad \forall \underline{v}_h \in \underline{U}_{h,0}. \quad (40b)$$

Finally, the discrete approximation of (ω, ψ) is given by

$$(\omega_h, \psi_h) := (p_h^{k+1} \underline{\omega}_h, p_h^{k+1} \underline{\psi}_h).$$

Note that while $(\cdot, \cdot)_{\mathcal{T}_h}^*$ is required within ℓ_h for problem (38) to be well-posed, it has to be used neither for the enforcement of the source functions in (37) and (40), nor for the computation of $\partial_{n,h}(\underline{\psi}_h^0)$ (via formula (24)) in the right-hand side of (38).

6 Preconditioner

In this section, we introduce the preconditioned iterative strategy for the solution of the algebraic realization of (38).

6.1 Algebraic setting

For a fixed basis for the discrete HHO spaces, let $N_B := \dim(\mathbb{P}^k(\mathcal{F}_h^B))$. Introduce the algebraic counterpart $\mathbf{L}_h: \mathbb{R}^{N_B} \rightarrow \mathbb{R}^{N_B}$ of the linear operator \mathcal{L}_h defined by (39). We recall that \mathbf{L}_h can be viewed as a matrix in $\mathbb{R}^{N_B \times N_B}$, whose coefficients are not explicitly known, but whose application to a vector can be computed. The algebraic realization of problem (38) reads: find $\boldsymbol{\lambda} \in \mathbb{R}^{N_B}$ such that

$$\mathbf{L}_h(\boldsymbol{\lambda}) = \mathbf{b}, \quad (41)$$

where $\mathbf{b} \in \mathbb{R}^{N_B}$ is an algebraic representation of $\partial_{n,h}(\underline{\psi}_h^0) - \mathbf{g}_N$ in the chosen basis.

6.2 An approximate, sparse matrix

The implicit system (41) is solved using a preconditioned, flexible conjugate gradient (PF CG) algorithm. This section describes the construction of the preconditioner. The use of a flexible version of the Krylov method is justified by the preconditioner being non-symmetric [1, 3]. Refer to Section 6.4 for a discussion about the symmetric property of the preconditioner.

Introducing $(\mathbf{e}_j)_{j=1, \dots, N_B}$ the canonical basis of \mathbb{R}^{N_B} , the j^{th} column of \mathbf{L}_h is given by the evaluation of $\mathbf{L}_h(\mathbf{e}_j)$. Note that $\mathbf{L}_h(\mathbf{e}_j)$ is a dense vector. Consequently, the explicit computation of \mathbf{L}_h would result in a dense matrix. The preconditioner proposed here consists in the explicit computation of a *sparse* matrix $\tilde{\mathbf{L}}_h$ approximating \mathbf{L}_h . In a nutshell, while the computation of each column $\mathbf{L}_h(\mathbf{e}_j)$ involves solving two Laplacian problems in the whole domain, our approximation $\tilde{\mathbf{L}}_h(\mathbf{e}_j)$ consists in solving those problems in a restricted neighbourhood of the j^{th} DoF.

Let $j \in \{1, \dots, N_B\}$ a fixed unknown index. Let $F^j \in \mathcal{F}_h^B$ the face supporting the DoF associated to j , and $T^j \in \mathcal{T}_h$ the unique element owning F^j . Define $\mathcal{T}_h^j \subset \mathcal{T}_h$ a neighbourhood of T^j : roughly, a set of elements around T^j , including T^j . Let Ω^j be the open, connected set such that $\overline{\Omega^j} := \cup_{T \in \mathcal{T}_h^j} \overline{T}$. Figure 1a illustrates (in grey) such a neighbourhood, where F^j is represented by a thick line and T^j by a darker triangle. Relatively to Ω^j , one defines the sets of interior and boundary faces by $(\mathcal{F}_h^I)^j := \{F \in \mathcal{F}_h^I \mid F \subset \Omega^j\}$ and $(\mathcal{F}_h^B)^j := \{F \in \mathcal{F}_h^B \mid F \subset \partial\Omega^j\}$, respectively.

Consider the operator $\tilde{\mathcal{L}}_h^j$, counterpart of \mathcal{L}_h in Ω^j , in which $(\mathcal{T}_h, \mathcal{F}_h^I, \mathcal{F}_h^B)$ is replaced with $(\mathcal{T}_h^j, (\mathcal{F}_h^I)^j, (\mathcal{F}_h^B)^j)$ for the solution of the Laplacian subproblems, and \mathcal{F}_h^B is replaced with $\mathcal{F}_h^B \cap (\mathcal{F}_h^B)^j$ for the computation of the normal derivative. This latter point indicates that the normal derivative is computed on $\partial\Omega^j \cap \partial\Omega$ only. For both subproblems, homogeneous Dirichlet conditions are enforced on the part of $\partial\Omega^j$ interior to Ω . Define $N_B^j := \dim \mathbb{P}^k((\mathcal{F}_h^B)^j)$, $(\tilde{\mathbf{e}}_j)_{j=1, \dots, N_B^j}$ the canonical basis of $\mathbb{R}^{N_B^j}$, and $\tilde{\mathbf{L}}_h^j: \mathbb{R}^{N_B^j} \rightarrow \mathbb{R}^{N_B^j}$ the algebraic representation of $\tilde{\mathcal{L}}_h^j$.

Supposing that the j^{th} basis function of $\mathbb{P}^k(\mathcal{F}_h^B)$ corresponds to the k^{th} basis function of $\mathbb{P}^k((\mathcal{F}_h^B)^j)$, we compute $\tilde{\mathbf{L}}_h^j(\tilde{\mathbf{e}}_k)$. As $\tilde{\mathbf{L}}_h^j(\tilde{\mathbf{e}}_k)$ yields the normal derivative only on $\partial\Omega^j \cap \partial\Omega$, we build the final, approximate column $\tilde{\mathbf{L}}_h(\mathbf{e}_j)$ from $\tilde{\mathbf{L}}_h^j(\tilde{\mathbf{e}}_k)$ by setting zero coefficients where the domain boundary is not covered by the neighbourhood boundary.

We claim that $\tilde{\mathbf{L}}_h^j(\mathbf{e}_j)$ yields a sparse approximation of $\mathbf{L}_h(\mathbf{e}_j)$. To support our claim, in Figures 1b and 1c we plot, in iso-value curves, the solutions to the first and second Laplacian problems computed in Ω^j . Moreover, Figures 1d and 1e plot the solutions of the Laplacian subproblems computed in the whole domain. We observe that, owing to the homogeneity of the first equation and the homogeneous Dirichlet condition everywhere except on one face, the solution culminates on that face and decreases towards zero as it goes further away. Roughly speaking, our strategy consists in approximating the solution locally in Ω^j while imposing it to be zero in the rest of the domain. Then, this first truncated solution is used as the source function in the second problem, also solved in Ω^j . Here, note that the quantity of interest is not the solution itself, but its normal derivative on the boundary. In particular, the neighbourhood of the considered face concentrates the most information, insofar as the solution flattens as it goes away from it. In that sense, $\tilde{\mathbf{L}}_h^j(\mathbf{e}_j)$ yields a reasonable approximation of $\mathbf{L}_h(\mathbf{e}_j)$.

6.3 Computation in practice

Regarding the actual computation of $\tilde{\mathbf{L}}_h$, one can note that the HHO matrix blocks used to solve the Laplacian problems in Ω^j and to compute the normal derivative can be extracted from the global ones. The sparsity of $\tilde{\mathbf{L}}_h$ is controlled by the number of boundary faces in the chosen neighbourhood, while the accuracy of the approximation depends on the size of the neighbourhood. In particular, remark that choosing $\mathcal{T}_h^j := \mathcal{T}_h$ for all j yields the actual matrix \mathbf{L}_h . In practice, we construct \mathcal{T}_h^j by adding successive layers of neighbours around T^j . The number of layers is defined by the parameter $\alpha \in \mathbb{N}_0$. \mathcal{T}_h^j is defined as $\mathcal{T}_h^j(\alpha)$, where $\mathcal{T}_h^j(0) := \{T^j\}$ and $\mathcal{T}_h^j(\alpha) := \mathcal{T}_h^j(\alpha - 1) \cup \{T \in \mathcal{T}_h \mid \exists T' \in \mathcal{T}_h^j(\alpha - 1) \text{ s.t. } \overline{T} \cap \overline{T'} \neq \emptyset\}$ for $\alpha \geq 1$. Note that this definition understands the neighbouring relationship as having at least one vertex in common. The neighbourhood represented in Figure 1a corresponds to $\alpha = 3$.

6.4 Discussion on symmetry

Choosing a constant α for all j allows to preserve a symmetric sparsity pattern. Indeed, considering $(i, j) \in \{1, \dots, N_B\}^2$, $(\tilde{\mathbf{L}}_h)_{ij} \neq 0$ implies that $F^i \subset \partial\Omega^j$. Consequently, $(\tilde{\mathbf{L}}_h)_{ji} \neq 0$ implies that, reciprocally, $F^j \subset \partial\Omega^i$. However, symmetry itself is not preserved, inasmuch as $(\tilde{\mathbf{L}}_h)_{ij}$ results from computations in the neighbourhood Ω^j whereas $(\tilde{\mathbf{L}}_h)_{ji}$ results from computations in $\Omega^i \neq \Omega^j$.

To enforce the symmetry of the matrix \mathbf{L}_h , one must ensure that for all couple $(i, j) \in \{1, \dots, N_B\}^2$, $\Omega^i = \Omega^j$. This requirement leads to a patch-based method: the set of boundary faces must be split into *patches*, i.e. connected subsets of boundary faces. For each patch, one single neighbourhood must be defined to process all DoFs included in that patch. This yields a block-diagonal matrix $\tilde{\mathbf{L}}_h$, one block corresponding to each patch. Algebraically, such a preconditioner is in fact an approximate block Jacobi preconditioner, insofar as each diagonal block approximates

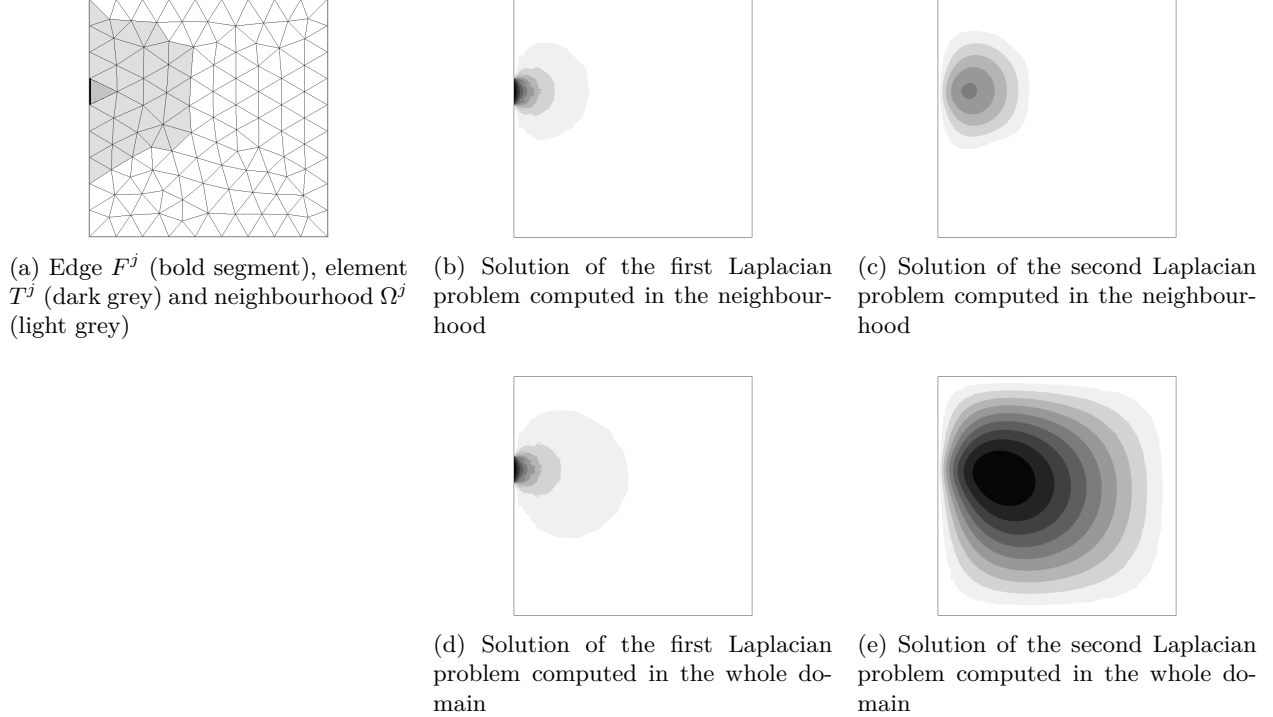


Figure 1: Illustration of the action of the preconditioner.

the corresponding diagonal block of the exact matrix \mathbf{L}_h . This symmetric preconditioner was numerically tested. Even with the facts that a non-flexible Krylov method can be used and that the matrix $\tilde{\mathbf{L}}_h$ is easier to factorize, the convergence of the method is significantly slower than with the unsymmetric preconditioner defined in Section 6.3. Therefore, only the latter has been used in the numerical experiments of Section 7.

7 Numerical experiments

7.1 Experimental setup

The implicit system (41) is solved iteratively by the PFCG method, using the preconditioning technique described in Section 6. Given the iterate $\tilde{\boldsymbol{\lambda}} \in \mathbb{R}^{N_B}$, the corresponding residual vector is defined as $\mathbf{r} := \mathbf{b} - \mathbf{L}_h(\tilde{\boldsymbol{\lambda}})$. The PFCG algorithm stops when $\|\mathbf{r}\|_2 / \|\mathbf{b}\|_2 < \varepsilon$, where $\varepsilon > 0$ is a fixed tolerance and $\|\cdot\|_2$ denotes the Euclidean norm on \mathbb{R}^{N_B} . All experiments are conducted choosing $l = k$, but we stress that the same qualitative results are obtained with $l = k + 1$. In 2D, the linear systems corresponding to the Laplacian subproblems are solved by Cholesky factorization; in 3D, we use a p -multigrid algorithm on top of the algebraic multigrid method designed in [11]. In that latter case, the same tolerance ε must be used to stop the Laplacian solvers and the global PFCG algorithm. To apply the preconditioner, we solve $\tilde{\mathbf{L}}_h$ using the BiCGSTAB algorithm with tolerance ε . The computations are run on an 8-core processor (AMR M1 Pro) clocked at 3228 MHz.

7.2 Preliminary: convergence of the discrete normal derivative

We first assert the validity of the discrete normal derivative proposed in Section 4 by evaluating its order of convergence. Let $\Omega := (0, 1)^2$ and $u: (x, y) \mapsto \sin(4\pi x) \sin(4\pi y)$ the manufactured solution of the boundary value problem (8), where f and g_D are defined accordingly. The problem is discretized on a sequence of successively refined meshes, and we consider $\partial_{n,h}(\underline{u}_h)$ computed by (25). The exact normal derivative $\partial_n u$ is known, and we assess the relative L^2 -error $\|\partial_n u - \partial_{n,h}(\underline{u}_h)\|_{L^2(\partial\Omega)} / \|\partial_n u\|_{L^2(\partial\Omega)}$ for $k \in \{0, 1, 2, 3\}$. Figures 2a and 2b present the experimental results on uniform Cartesian meshes and on unstructured, triangular Delaunay meshes, respectively. These experiments assess a convergence in $\mathcal{O}(h^{k+1})$ (also observed for $l = k + 1$). Note that in the case of a non-convex domain

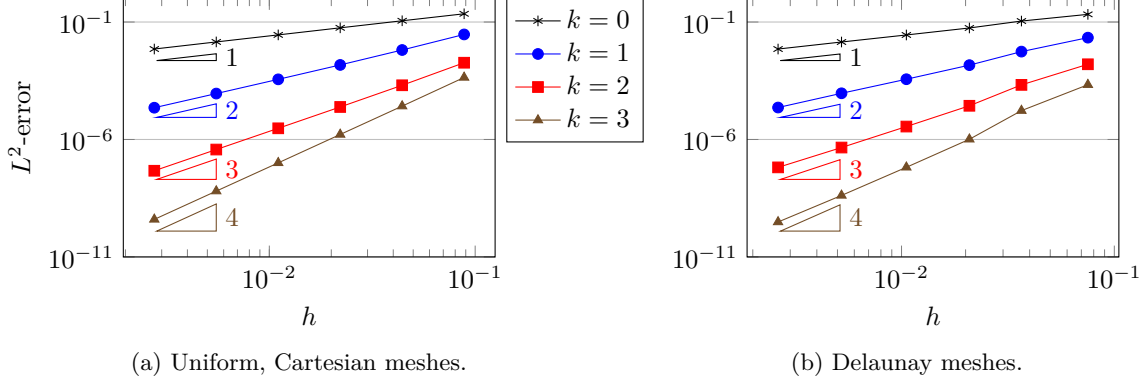


Figure 2: Convergence of the discrete normal derivative $\partial_{n,h}(\underline{u}_h)$ for a smooth solution.

with re-entrant corners, those convergence orders are expected to be reduced due to the corner singularities [23]. One way to mitigate this issue is the use of graded meshes (cf. [23]).

7.3 Convergence rate of the scheme

On the unit square, we consider $\psi(x, y) = x \sin(\pi y) e^{-xy}$ and its corresponding $\omega = -\Delta\psi$, the manufactured solution of (2), with \mathbf{f}, \mathbf{g}_D and \mathbf{g}_N determined accordingly. The square is decomposed into a sequence of successively finer meshes, and the numerical scheme is applied for $k \in \{0, 1, 2, 3\}$. Figure 3 presents the evolution of the relative L^2 -error achieved by the approximation (ω_h, ψ_h) with respect to (ω, ψ) , on a sequence of Cartesian meshes. Similarly, Figure 4 shows the analogous results obtained with a sequence of polygonal meshes, each constructed from an initial Cartesian mesh by agglomeration and face collapsing. One such mesh is represented in Figure 6. We empirically interpret these results in light of Glowinski *et al.*'s theoretical findings in [19] for the continuous high-order FEM. According to [19, Section 3.3], since ψ is regular enough, the FEM solution $(\omega_h^{\text{FEM}}, \psi_h^{\text{FEM}})$ of order $p \geq 3$ verifies

$$\|\psi_h^{\text{FEM}} - \psi\|_{L^2(\Omega)} + h^2 \|\omega_h^{\text{FEM}} - \omega\|_{L^2(\Omega)} \leq Ch^{p+1} \|\psi\|_{H^{p+1}(\Omega)}, \quad (42)$$

where C is a constant independent of h and ψ . In our HHO setting, the above error estimate would translate in the following one: if $k \geq 2$, then

$$\|\psi_h - \psi\|_{L^2(\Omega)} + h^2 \|\omega_h - \omega\|_{L^2(\Omega)} \leq Ch^{k+2} \|\psi\|_{H^{k+2}(\Omega)}. \quad (43)$$

The experimental results of Figure 3 validate the estimate (43), not only for $k \geq 2$, but also for $k \in \{0, 1\}$. More precisely, while Figure 3b shows that the estimate is sharp for ψ (i.e. $\mathcal{O}(h^{k+2})$), Figure 3a shows, for ω , a faster convergence than indicated by (43). Namely, the convergence seems to be $\mathcal{O}(h^{k+1})$ for $k = 0$ and $\mathcal{O}(h^{k+1/2})$ for $k \geq 1$. The experiments of Figure 6 on polygonal meshes exhibit the same convergence orders, except for ω_h with $k = 0$, where the error estimate reduces to $\mathcal{O}(h^{k+1/2})$, in accordance with the other values of k . Based on these experiments, we conjecture that the following error estimate is sharp for all $k \geq 0$:

$$\|\psi_h - \psi\|_{L^2(\Omega)} + h^{3/2} \|\omega_h - \omega\|_{L^2(\Omega)} \leq Ch^{k+2} \|\psi\|_{H^{k+2}(\Omega)}. \quad (44)$$

Nonetheless, we keep in mind that $k = 0$ seems to be a special case for ω_h , where superconvergence may be observed: besides the order 1 obtained on Cartesian meshes (Figure 3a), we observe the order 2 if we choose as an exact solution the polynomial function $\psi(x, y) = x^4(x-1)^2y^4(y-1)^2$, as illustrated in Figure 5.

Remark 3. (Comparison with [17]) The HHO discretizations of [17] rely on the primal formulation (1) of the equation. They hinge on the local approximation space $\mathbb{P}^{k+2}(T) \times \mathbb{P}^{k+1}(\mathcal{F}_T) \times \mathbb{P}^k(\mathcal{F}_T)$ in 2D (resp. $\mathbb{P}^{k+2}(T) \times \mathbb{P}^{k+2}(\mathcal{F}_T) \times \mathbb{P}^k(\mathcal{F}_T)$ in 3D), where the associated set of DoFs aims at approximating $(\psi|_T, \psi|_{\partial T}, \nabla\psi \cdot \mathbf{n}_{\partial T})$. With this setting, the methods achieve a convergence order of $k+3$ in L^2 -norm. In comparison, the present method requires the space $\mathbb{P}^{k+1}(T) \times \mathbb{P}^{k+1}(\mathcal{F}_T)$ to obtain the same convergence order, i.e. one less polynomial degree for the cell unknowns and no DoF dedicated to the approximation of $\nabla\psi \cdot \mathbf{n}_{\partial T}$. However, if our method is structurally lighter in DoFs than [17], one cannot conclude regarding the computational cost, since those DoFs have to be solved multiple times as part of an iterative process (vs. only once for [17]).

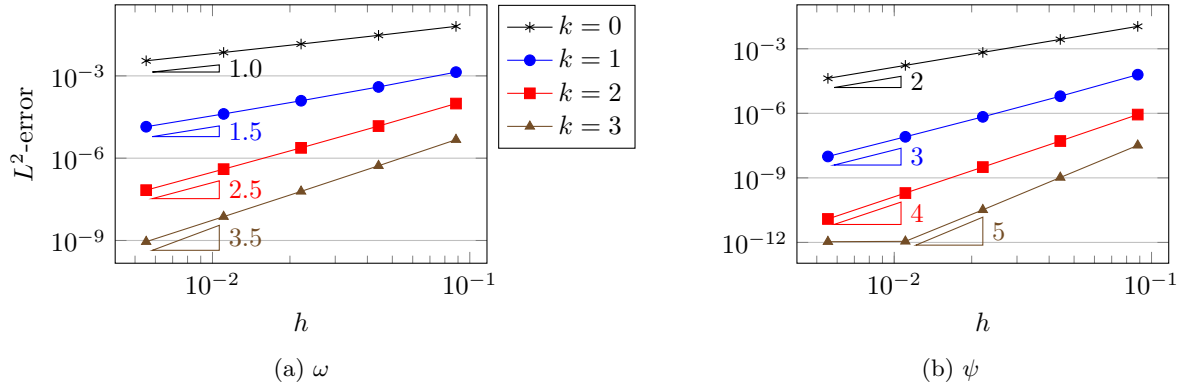


Figure 3: Convergence in L^2 -norm of the discrete solution (ω_h, ψ_h) with respect to the exact solution $\psi(x, y) = x \sin(\pi y)e^{-xy}$. Square domain discretized by Cartesian meshes.

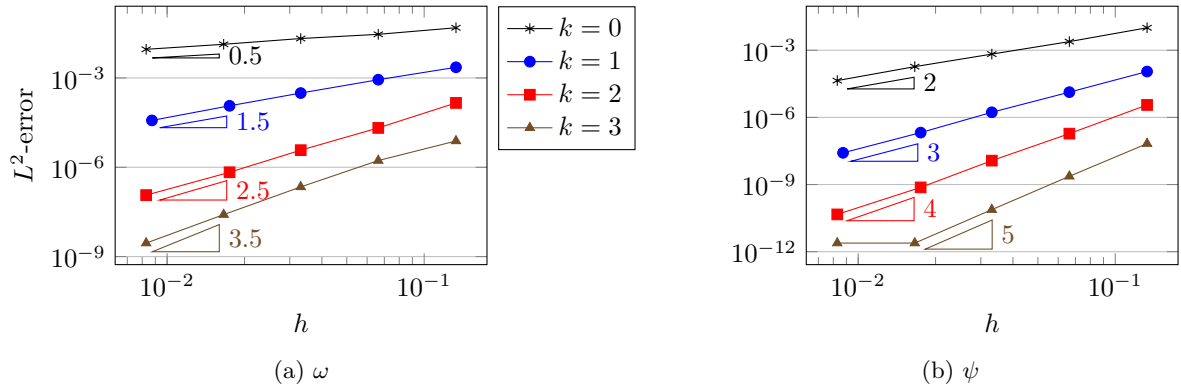


Figure 4: Convergence in L^2 -norm of the discrete solution (ω_h, ψ_h) with respect to the exact solution $\psi(x, y) = x \sin(\pi y)e^{-xy}$. Square domain discretized by polygonal meshes.

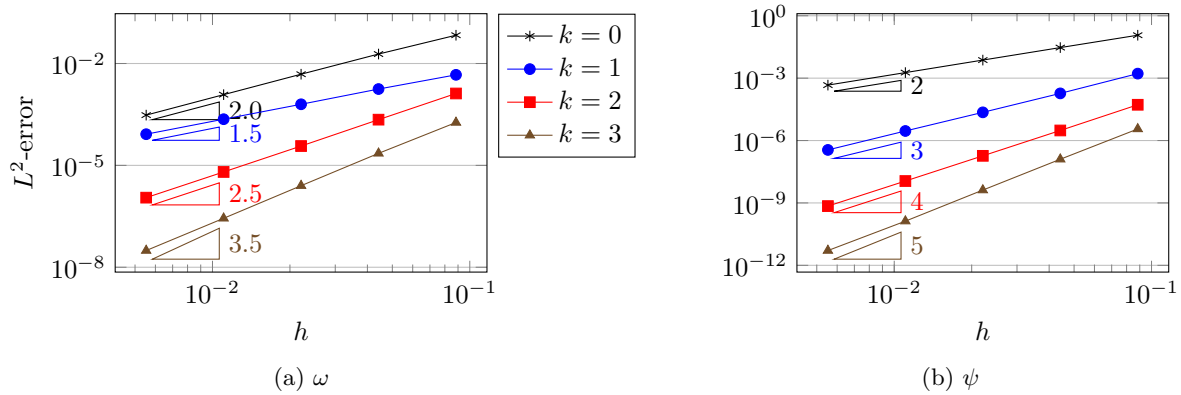


Figure 5: Convergence in L^2 -norm of the discrete solution (ω_h, ψ_h) with respect to the exact solution $\psi(x, y) = x^4(x-1)^2y^4(y-1)^2$. Square domain discretized by Cartesian meshes.

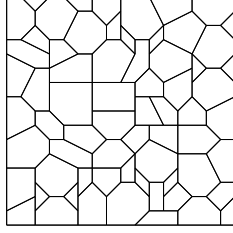


Figure 6: Example of a polygonal mesh of the unit square $\Omega = (0, 1)^2$.

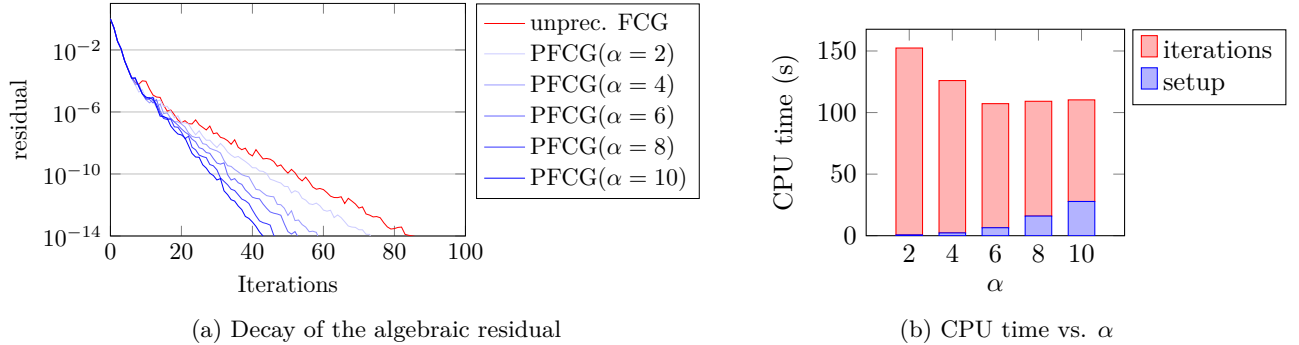


Figure 7: (a) compares the speed of convergence of the (P)FCG w.r.t. the normalized algebraic residual $\|\mathbf{r}\|_2/\|\mathbf{b}\|_2$. (b) compares the CPU time consumed to reach a tolerance of 10^{-14} , in function of α . The test case is the square domain partitioned into 256×256 Cartesian elements, $k = 1$.

7.4 Preconditioner

We evaluate in this section the efficiency of the preconditioner designed in Section 6. We recall that the parameter $\alpha \in \mathbb{N}_0$ is an indicator of the size of the neighbourhoods used for the approximation of the matrix.

7.4.1 Convergence vs. cost: the parameter α

The considered test case is the square domain partitioned into 256×256 Cartesian elements, with $k = 1$. The tolerance is set to $\varepsilon = 10^{-14}$. Let us assess how the preconditioner improves the speed of convergence. Figure 7a plots the decay of the algebraic residual $\|\mathbf{r}\|_2/\|\mathbf{b}\|_2$ without preconditioner (in red) and with preconditioner (in blue), considering increasing values of α . One can confirm that, as expected, the higher the value of α , the better the convergence rate. However, increasing α increases the setup cost of the preconditioner. Indeed, (i) the Laplacian problems are solved in larger subdomains, (ii) the approximate matrix $\tilde{\mathbf{L}}_h$ is denser, which increases the cost of its factorization. Figure 7b presents the computational cost, measured in CPU time, to reach the algebraic solution. One can see that a trade-off between cost and convergence rate must be made to achieve optimal performance. If raising α is a good strategy up to $\alpha = 6$, above that threshold, the better convergence rate does not make up for the cost of the setup phase, hence making the overall CPU time slightly increase. We stress that other choices of k yield the same qualitative results.

7.4.2 PFCG convergence rate: k -independence and h -dependency assessment

Problem (2) on the unit square with exact solution $\psi(x, y) = x \sin(\pi y) e^{-xy}$ is solved on a sequence of 2D grids composed of N^2 Cartesian elements, $N \in \{32, 64, 128, 256, 512\}$. Table 1 presents, for $k \in \{0, 1, 2, 3\}$, the number of PFCG iterations executed to reach the convergence criterion with $\varepsilon = 10^{-8}$. The parameter of the preconditioner is set to $\alpha = 8$. For comparison, the number of iterations without preconditioning is reported in brackets. Firstly, one can remark that the use of the preconditioner allows a convergence rate independent of k , while a dependency is observed without preconditioner. Secondly, we approximate the asymptotic convergence rate with respect to the problem size by using the number of iterations measured on the two finer meshes. This yields a dependency in $\mathcal{O}(h^{-0.41})$. Additionally, Table 2 reports the CPU times of the setup and iteration phases. The setup includes the Cholesky factorization of the Laplacian matrix and the assembly of the preconditioning matrix. In brackets, the same

| $N =$ | 32 | 64 | 128 | 256 | 512 |
|---------|---------|---------|---------|---------|----------|
| $k = 0$ | 13 (19) | 19 (25) | 26 (33) | 33 (42) | 44 (56) |
| $k = 1$ | 13 (30) | 19 (36) | 26 (47) | 33 (58) | 44 (75) |
| $k = 2$ | 13 (37) | 19 (47) | 26 (62) | 33 (77) | 44 (101) |
| $k = 3$ | 13 (44) | 19 (58) | 26 (75) | 33 (96) | 44 (122) |

Table 1: Test case: N^2 Cartesian elements, $\varepsilon = 10^{-8}$. Number of PFCG($\alpha = 8$) iterations. In brackets, number of unpreconditioned FCG iterations.

| $N =$ | | 32 | | 64 | | 128 | | 256 | | 512 | |
|---------|-------|------|-------|------|--------|------|---------|-------|----------|--------|----------|
| $k = 0$ | setup | 0.4 | (0.0) | 0.9 | (0.0) | 1.9 | (0.0) | 3.9 | (0.0) | 10.7 | (2.0) |
| | iter. | 0.1 | (0.1) | 0.3 | (0.4) | 1.8 | (2.3) | 9.4 | (11.8) | 50.8 | (66.7) |
| $k = 1$ | setup | 1.2 | (0.0) | 2.8 | (0.0) | 5.7 | (0.2) | 13.9 | (1.7) | 41.0 | (17.2) |
| | iter. | 0.2 | (0.5) | 1.1 | (2.2) | 6.3 | (11.5) | 34.1 | (60.1) | 194.9 | (336.3) |
| $k = 2$ | setup | 4.8 | (0.0) | 10.1 | (0.0) | 22.0 | (0.6) | 51.0 | (5.6) | 147.8 | (57.0) |
| | iter. | 0.6 | (1.7) | 3.5 | (8.7) | 20.1 | (48.6) | 109.1 | (251.9) | 646.2 | (1450.4) |
| $k = 3$ | setup | 18.2 | (0.0) | 40.5 | (0.1) | 86.8 | (1.2) | 188.4 | (13.4) | 508.0 | (155.7) |
| | iter. | 1.8 | (6.3) | 10.8 | (33.2) | 61.4 | (177.3) | 319.4 | (1028.9) | 2028.4 | (6271.7) |

Table 2: Test case: N^2 Cartesian elements, $\varepsilon = 10^{-8}$. CPU times (in seconds) of the setup and iteration phases. In brackets: analogous quantity without preconditioning.

information without preconditioning is displayed. Based on these results, one can see that the compensation of setup cost by the gain in iteration time is more and more efficient as the problem grows larger.

The equivalent experiment is performed in 3D (exact solution $\psi(x, y) = xz \sin(\pi y)e^{-xy}$) with unstructured tetrahedral meshes. As the preconditioner is costlier in 3D, we choose $\alpha = 2$, and set the tolerance to $\varepsilon = 10^{-5}$. The number of PFCG iterations (independent of k) is presented in Table 3, which exhibits a dependency in $\mathcal{O}(h^{-0.37})$. The CPU times of setup and iteration phases are reported for $k = 0$.

| $h \approx$ | h_0 | $h_0/2$ | $h_0/4$ | $h_0/8$ |
|--------------|-------|---------|---------|-----------|
| Elements | 3373 | 22,869 | 162,167 | 1,224,468 |
| iter. counts | 9 | 12 | 17 | 22 |
| setup time | 5.9 | 24.4 | 98.3 | 426.3 |
| iter. time | 0.2 | 1.5 | 17.8 | 263.5 |

Table 3: Test case: Cubic domain, unstructured tetrahedral mesh, $\varepsilon = 10^{-5}$. Number of PFCG($\alpha = 2$) iterations, along with the setup and iteration CPU times (in seconds) for $k = 0$.

8 Conclusion

In this work, we extended the scheme of Glowinski *et al.* to HHO discretizations, yielding an iterative method for the mixed solution of the biharmonic equation. Its main advantage lies in the fact that it can be implemented from an existing diffusion code with limited development costs. Namely, notwithstanding the PFCG solver, one requires only the implementation of (29) and the right-hand side of (28b). Additionally, for large problems, it allows the use of fast, elliptic solvers for an enhanced time to solution. Future work will focus on the a priori error analysis of the proposed method and on improving the preconditioner, in order to obtain a convergence rate of the PFCG that is independent of the mesh size, and achieve a scalable behaviour.

References

- [1] R. Blaheta. GPCG-generalized preconditioned CG method and its use with non-linear and non-symmetric displacement decomposition preconditioners. *Numer. Linear Algebra Appl.*, 9(6-7):527–550, 2002. doi:10.1002/nla.295.
- [2] F. Bonaldi, D. A. Di Pietro, G. Geymonat, and F. Krasucki. A Hybrid High-Order method for Kirchhoff–Love plate bending problems. *ESAIM: Math. Model. Numer. Anal.*, 52(2):393–421, 2018. doi:10.1051/m2an/2017065.
- [3] H. Bouwmeester, A. Dougherty, and A. V. Knyazev. Nonsymmetric Preconditioning for Conjugate Gradient and Steepest Descent Methods 1. *Procedia Computer Science*, 51:276–285, 2015. doi:10.1016/j.procs.2015.05.241.
- [4] F. Chave, D. A. Di Pietro, F. Marche, and F. Pigeonneau. A Hybrid High-Order method for the Cahn-Hilliard problem in mixed form. *SIAM J. Numer. Anal.*, 54(3):1873–1898, 2016. doi:10.1137/15M1041055.
- [5] P. G. Ciarlet and P.-A. Raviart. A mixed finite element method for the biharmonic equation. In *Mathematical aspects of finite elements in partial differential equations (Proc. Sympos., Math. Res. Center, Univ. Wisconsin, Madison, Wis., 1974)*, pages 125–145. Publication No. 33, 1974.
- [6] P.G. Ciarlet and R. Glowinski. Dual iterative techniques for solving a finite element approximation of the biharmonic equation. *Comput. Methods Appl. Mech. Eng.*, 5(3):277–295, 1975. doi:10.1016/0045-7825(75)90002-X.
- [7] B. Cockburn, D. A. Di Pietro, and A. Ern. Bridging the hybrid high-order and hybridizable discontinuous Galerkin methods. *ESAIM: Math. Model. Numer. Anal.*, 50(3):635–650, 2016. doi:10.1051/m2an/2015051.
- [8] D. A. Di Pietro and J. Droniou. *The Hybrid High-Order method for polytopal meshes*. Number 19 in Modeling, Simulation and Application. Springer International Publishing, 2020. doi:10.1007/978-3-030-37203-3.
- [9] D. A. Di Pietro, A. Ern, and S. Lemaire. An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators. *Comput. Meth. Appl. Math.*, 14(4):461–472, 2014. doi:10.1515/cmam-2014-0018.
- [10] D. A. Di Pietro, L. Formaggia, and R. Masson, editors. *Polyhedral Methods in Geosciences*, volume 27 of *SEMA SIMAI Springer Series*. Springer International Publishing, Cham, 2021. doi:10.1007/978-3-030-69363-3.
- [11] D. A. Di Pietro, F. Hülsemann, P. Matalon, P. Mycek, and U. Rüde. Algebraic multigrid preconditioner for statically condensed systems arising from lowest-order hybrid discretizations. *SIAM J. Sci. Comput.*, 2022. To appear. doi:10.1137/21M1429849.
- [12] D. A. Di Pietro, F. Hülsemann, P. Matalon, P. Mycek, U. Rüde, and D. Ruiz. An h -multigrid method for Hybrid High-Order discretizations. *SIAM J. Sci. Comput.*, 43(5):S839–S861, 2021. doi:10.1137/20M1342471.
- [13] D. A. Di Pietro, F. Hülsemann, P. Matalon, P. Mycek, U. Rüde, and D. Ruiz. Towards robust, fast solutions of elliptic equations on complex domains through hybrid high-order discretizations and non-nested multigrid methods. *Int. J. Numer. Methods Eng.*, 122(22):6576–6595, 2021. doi:10.1002/nme.6803.
- [14] D. A. Di Pietro, P. Matalon, P. Mycek, and U. Rüde. High-order multigrid strategies for hho discretizations of elliptic equations. *Numer. Linear Algebra Appl.*, 30(1):e2456, 2023. doi:10.1002/nla.2456.
- [15] Z. Dong and A. Ern. Hybrid high-order method for singularly perturbed fourth-order problems on curved domains. *ESAIM: Math. Model. Numer. Anal.*, 55(6):3091–3114, 2021. doi:10.1051/m2an/2021081.
- [16] Z. Dong and A. Ern. C^0 -hybrid high-order methods for biharmonic problems, 2022. doi:10.48550/ARXIV.2206.07074.
- [17] Z. Dong and A. Ern. Hybrid High-Order and Weak Galerkin Methods for the Biharmonic Problem. *SIAM J. Numer. Anal.*, 60(5):2626–2656, 2022. doi:10.1137/21M1408555.
- [18] Richard S. Falk. Approximation of the Biharmonic Equation by a Mixed Finite Element Method. *SIAM J. Numer. Anal.*, 15(3):556–567, 1978. doi:10.1137/0715036.
- [19] R. Glowinski and O. Pironneau. Numerical Methods for the First Biharmonic Equation and for the Two-Dimensional Stokes Problem. *SIAM Review*, 21(2):167–212, 1979. doi:10.1137/1021028.
- [20] I. Gustafsson. A Preconditioned Iterative Method for the Solution of the Biharmonic Problem. *IMA Journal of Numerical Analysis*, 4(1):55–67, 1984. doi:10.1093/imanum/4.1.55.
- [21] Milan D. Mihajlović and David J. Silvester. Efficient parallel solvers for the biharmonic equation. *Parallel Computing*, 30(1):35–55, 2004. doi:10.1016/j.parco.2003.07.002.
- [22] P. Peisker. On the numerical solution of the first biharmonic equation. *ESAIM: Math. Model. Numer. Anal.*, 22(4):655–676, 1988. doi:10.1051/m2an/1988220406551.

- [23] J. Pfefferer and M. Winkler. Finite element error estimates for normal derivatives on boundary concentrated meshes. *SIAM J. Numer. Anal.*, 57(5):2043–2073, 2019. doi:10.1137/18M1181341.
- [24] A. Quarteroni and A. Valli. Theory and Application of Steklov-Poincaré Operators for Boundary-Value Problems. In Renato Spigler, editor, *Applied and Industrial Mathematics*, pages 179–203. Springer Netherlands, Dordrecht, 1991. doi:10.1007/978-94-009-1908-2_14.
- [25] M. B. van Gijzen. Conjugate Gradient-like solution algorithms for the mixed finite element approximation of the biharmonic equation, applied to plate bending problems. *Comput. Methods Appl. Mech. Eng.*, 121(1):121–136, 1995. doi:10.1016/0045-7825(94)00737-8.

A Proof of Lemma 1

Let $v_{\mathcal{F}_h} \in \mathbb{P}^k(\mathcal{F}_h)$ and $\underline{w}_h \in \underline{U}_h$. For all $T \in \mathcal{T}_h$, $\Theta_T v_{\partial T}$ is by definition solution of (17). By linearity of a_h , combining both terms of (17) yields

$$a_T(\Theta_T v_{\partial T}, (w_T, 0)) = 0.$$

That being true for all $T \in \mathcal{T}_h$ proves the global counterpart

$$a_h(\Theta_h v_{\mathcal{F}_h}, (w_{\mathcal{T}_h}, 0)) = 0. \tag{45}$$

By splitting \underline{w}_h into $(w_{\mathcal{T}_h}, 0) + (0, w_{\mathcal{F}_h})$ and by linearity of a_h , we have

$$a_h(\Theta_h v_{\mathcal{F}_h}, \underline{w}_h) = a_h(\Theta_h v_{\mathcal{F}_h}, (w_{\mathcal{T}_h}, 0)) + a_h(\Theta_h v_{\mathcal{F}_h}, (0, w_{\mathcal{F}_h})).$$

Property (45) allows to cancel the first term of the right-hand side. Then, by writing $(0, w_{\mathcal{F}_h}) = (\Theta_{\mathcal{T}_h \mathcal{F}_h} w_{\mathcal{F}_h}, w_{\mathcal{F}_h}) - (\Theta_{\mathcal{T}_h \mathcal{F}_h} w_{\mathcal{F}_h}, 0) = \underline{\Theta}_h w_{\mathcal{F}_h} - (\Theta_{\mathcal{T}_h \mathcal{F}_h} w_{\mathcal{F}_h}, 0)$ and using the linearity of a_h , the second term rewrites

$$a_h(\Theta_h v_{\mathcal{F}_h}, (0, w_{\mathcal{F}_h})) = a_h(\Theta_h v_{\mathcal{F}_h}, \underline{\Theta}_h w_{\mathcal{F}_h}) - a_h(\Theta_h v_{\mathcal{F}_h}, (\Theta_{\mathcal{T}_h \mathcal{F}_h} w_{\mathcal{F}_h}, 0)). \tag{46}$$

Using (45) again to cancel the last term of (46) finishes the proof of (20).

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 62/2023** Zappon, E.; Manzoni, A.; Quarteroni, A.
A staggered-in-time and non-conforming-in-space numerical framework for realistic cardiac electrophysiology outputs
- 61/2023** Afriaca, P.C.A; Piersanti, R.; Regazzoni, F.; Bucelli, M.; Salvador, M.; Fedele, M.; Pagani, S.; Dede', L.; Quarteroni, A.
lifex-ep: a robust and efficient software for cardiac electrophysiology simulations
- Federico Gatti, Carlo de Falco, Simona Perotto, Luca Formaggia, Manuel Pastor
A scalable well-balanced numerical scheme for the modeling of two-phase shallow granular landslide consolidation
- 60/2023** Negrini G.; Parolini N.; Verani M.
The Rhie-Chow stabilized Box Method for the Stokes problem
- 59/2023** Corti, M.; Bonizzoni, F.; Antonietti, P.F.
Structure Preserving Polytopal Discontinuous Galerkin Methods for the Numerical Modeling of Neurodegenerative Diseases
- 58/2023** Montino Pelagi, G.; Baggiano, A.; Regazzoni, F.; Fusini, L.; Ali, M.; Pontone, G.; Valbusa, G.; Vergara, C.
Personalized pressure conditions and calibration for a predictive computational model of coronary and myocardial blood flow
- 57/2023** Regazzoni, F.
An optimally convergent Fictitious Domain method for interface problems
- 55/2023** Orlando, G; Barbante, P.F.; Bonaventura, L.
On the evolution equations of interfacial variables in two-phase flows
- 54/2023** Orlando, G.
An implicit DG solver for incompressible two-phase flows with an artificial compressibility formulation
- 53/2023** Rossi, A.; Cappozzo, A.; Ieva, F.
Functional Boxplot Inflation Factor adjustment through Robust Covariance Estimators