An h-multigrid method for Hybrid High-Order discretizations

Pierre Matalon, Daniele Antonio Di Pietro, Paul Mycek, Ulrich Rüde, Daniel Ruiz

AN H-MULTIGRID METHOD FOR HYBRID HIGH-ORDER DISCRETIZATIONS *

P. MATALON †

In collaboration with: D. A. Di Pietro, P. Mycek, U. Rüde, D. Ruiz

Abstract. We consider a second order elliptic PDE discretized by the Hybrid High Order (HHO) method, for which globally coupled unknowns are located at faces. To efficiently solve the resulting linear system, we propose a geometric multigrid algorithm that keeps the degrees of freedom on the faces at every level. The core of the algorithm resides in the design of the prolongation operator that passes information from coarse to fine faces through the reconstruction of an intermediary polynomial of higher degree on the cells. Higher orders are natively handled by the conservation of the same polynomial degree at every level. The proposed algorithm requires a hierarchy of nested meshes where the faces are also successively coarsened. Numerical tests on homogeneous and heterogeneous diffusion problems in square and cubic domains show fast convergence, scalability in the mesh size and polynomial order, and robustness with respect to heterogeneity of the diffusion coefficient.

Key words. Partial Differential Equations, Hybrid High-Order, Multigrid, static condensation.

1. Introduction. We address in this work the solution of large sparse linear systems arising from Hybrid High-Order (HHO, [4, 3]) discretizations. HHO methods hinge on discrete unknowns that are broken polynomials on the mesh and its skeleton, and are designed so that element-based unknowns are not directly coupled with each other. As a result, the corresponding degrees of freedom (DoFs) can be efficiently eliminated from the linear system by computing a Schur complement element by element, a procedure known in the mechanical literature as *static condensation*. The discrete solution can then be obtained in two steps: first, the Schur complement system, hereafter called *trace system*, is solved, yielding the values of the face unknowns; second, interior unknowns are recovered element-wise by solving a small local system. HHO stands amongst hybrid methods as one of the most efficient, owing to a special stabilization term that permits to gain one order of convergence with respect to other methods based on a similar set of unknowns [1].

The main difficulty in designing a geometric h-multigrid algorithm for trace systems lies in the fact that functional spaces on the mesh skeleton may be non-nested when coarsening. This prevents the straightforward construction of a multigrid algorithm based on standard ingredients. Although no existing geometric h-multigrid method has specifically targeted HHO so far, a few trace system solvers have been designed over the last years. In [2], the authors recast the trace functions into functions defined over the elements in order to make use of a known efficient solver. A different approach is considered in [8], where an hp-multigrid algorithm based on trace functions at every level is proposed.

In this paper, we propose a novel geometric h-multigrid algorithm (i) based on approximation spaces supported by the mesh skeleton at every level, (ii) targeting HHO discretizations by making use of the underlying high-order potential reconstruction, (iii) natively managing higher orders (as opposed to, e.g., putting a p-multigrid on top of an h- one). The polynomial order of approximation is preserved at every

*This work is financed by the ANR project Fast4HHO under contract ANR-17-CE23-0019.

†CERFACS (matalon@cerfacs.fr), University of Montpellier, FAU Erlangen-Nürnberg, IRIT Toulouse.

level at the sole cost of using a blockwise smoother instead of a pointwise one. This approach originates from the remark that a high-order finite element discretization yields a block matrix, whose diagonal blocks are formed by the degrees of freedom connected to the same cell. This configuration usually destroys the desirable M- or H-matrix structure and, along with it, the convergence of pointwise smoothers; on the other hand, the block structure paves the way to using block versions of similar smoothers. In a more functional way of thinking, relaxing together the DoFs related to the same polynomial comes as intuitive. The robustness of multigrid algorithms using block smoothers for high-order methods has been experimentally illustrated in [5] and later used in practical solvers such as [7].

The rest of this work is organized as follows. Section 2 summarizes the construction of the HHO method. Section 3 is devoted to the construction of the multigrid algorithm and illustrates how it takes advantage of the HHO potential reconstruction operator. Numerical results for various polynomial degrees are presented in Section 4, considering both homogeneous and heterogeneous diffusion problems in two and three space dimensions. The numerical experiments show that the number of iterations is nearly independent of the mesh size and of the presence of jumps in the diffusion coefficient. Finally, future research directions are discussed in conclusion.

2. HHO formulation.

2.1. Notation. Let $d \in \{2,3\}$ be the space dimension and Ω a bounded polyhedral domain of $\mathbb{R}^d$. Ω is discretized by the mesh $(\mathcal{T}_h, \mathcal{F}_h)$, where $\mathcal{T}_h$ denotes the set of polyhedral elements T, $\mathcal{F}_h$ the set of faces F, and $h := \max_{T\in\mathcal{T}_h} \text{diameter}(T)$. For $T \in \mathcal{T}_h$, $\mathcal{F}_T$ denotes the set of faces of T, and for $F \in \mathcal{F}_T$, $\mathbf{n}_{TF}$ denotes the unit vector normal to F pointing out of T. For $X \subset \Omega$, $L^2(X)$ denotes the Hilbert space of square-integrable functions defined on X, equipped with its usual inner product $(u,v)_X := \int_X uv$. The same notation is used for vector-valued functions of $[L^2(X)]^d$: $(u,v)_X := \int_X u \cdot v$. $H^1(X)$ denotes the Sobolev space of order 1, i.e. the functions of $L^2(X)$ whose partial derivatives are also square-integrable. $H^1_0(X)$ defines the subspace of $H^1(X)$ whose functions vanish on the boundary ∂X in the sense of traces. Finally, $\mathbb{P}^\ell(X)$ denotes the space spanned by the restriction to X of d-variate polynomials of total degree at most ℓ, $\ell \in \mathbb{N}$.

2.2. Model problem. We consider the following diffusion problem with homogeneous Dirichlet boundary conditions:

$$\begin{cases} -\nabla\cdot(\mathbf{K}\nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{2.1}$$

where the diffusion tensor $\mathbf{K}\colon \Omega \to \mathbb{R}^{d\times d}_{\text{sym}}$ (with $\mathbb{R}^{d\times d}_{\text{sym}}$ denoting the space of symmetric $d\times d$ real matrices) is assumed uniformly elliptic and piecewise constant over Ω. The variational formulation of (2.1) reads

$$\text{Find } u \in H^1_0(\Omega) \text{ such that } a(u,v) = \int_\Omega fv \quad \forall v \in H^1_0(\Omega), \tag{2.2}$$

where the bilinear form $a\colon H^1(\Omega)\times H^1(\Omega) \to \mathbb{R}$ is such that, for all $v, w \in H^1(\Omega)$,

$$a(v,w) := (\mathbf{K}\nabla v, \nabla w)_\Omega := \int_\Omega \mathbf{K}\nabla v\cdot\nabla w.$$

We assume in what follows that $\mathcal{T}_h$ partitions Ω in such a way that the diffusion tensor is constant inside each element, and we denote $\mathbf{K}_T := \mathbf{K}|_T$ for all $T \in \mathcal{T}_h$.

Decomposing the global integral in (2.2) as a sum of local integrals on the elements of the mesh $\mathcal{T}_h$, problem (2.2) becomes

$$\sum_{T\in\mathcal{T}_h}(\mathbf{K}_T\nabla u,\nabla v)_T=\sum_{T\in\mathcal{T}_h}(f,v)_T\quad\forall v\in H^1_0(\Omega). \tag{2.3}$$

2.3. Discrete spaces and operators. We briefly recall the standard HHO discretization of problem (2.3). For a more comprehensive presentation, see [3, §3.1].

The HHO method is based on discrete unknowns at cells and faces, and the adjective *hybrid* refers to their different nature. Given an arbitrary polynomial degree $k\geq 0$, the discrete unknowns can be interpreted as the polynomial moments of degree up to k of the solution on the corresponding geometric entity.

Specifically, for all $T\in\mathcal{T}_h$, we introduce the following space of local variables:

$$\underline{U}_T^k:=\left\{\underline{v}_T:=(v_T,(v_F)_{F\in\mathcal{F}_T})\mid v_T\in\mathbb{P}^k(T),\ v_F\in\mathbb{P}^k(F)\ \ \forall F\in\mathcal{F}_T\right\}. \tag{2.4}$$

The discrete variables associated to a function $v\in H^1(T)$ are obtained through the local interpolation operator $\underline{I}_T^k\colon H^1(T)\to\underline{U}_T^k$ defined by

$$\underline{I}_T^k v:=\left(\pi_T^k v,(\pi_F^k v)_{F\in\mathcal{F}_T}\right), \tag{2.5}$$

where, for any $X\in\mathcal{T}_h\cup\mathcal{F}_h$, $\pi_X^k\colon L^2(X)\to\mathbb{P}^k(X)$ denotes the L^2-orthogonal projector on $\mathbb{P}^k(X)$. Given the discrete variables $\underline{I}_T^k v\in\underline{U}_T^k$ associated to $v\in H^1(T)$, a higher-degree approximation of v can be reconstructed inside T. This is achieved by means of the *local potential reconstructor* $p_T^{k+1}\colon\underline{U}_T^k\to\mathbb{P}^{k+1}(T)$ such that, for all $\underline{v}_T:=(v_T,(v_F)_{F\in\mathcal{F}_T})\in\underline{U}_T^k$, $p_T^{k+1}\underline{v}_T$ is the unique polynomial of degree at most $k+1$ verifying

$$\begin{cases}(\mathbf{K}_T\nabla p_T^{k+1}\underline{v}_T,\nabla w)_T=-(v_T,\nabla\cdot(\mathbf{K}_T\nabla w))_T+\displaystyle\sum_{F\in\mathcal{F}_T}(v_F,\mathbf{K}_T\nabla w\cdot\mathbf{n}_{TF})_F\\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\forall w\in\mathbb{P}^{k+1}(T),\end{cases} \tag{2.6a}$$

$$(p_T^{k+1}\underline{v}_T,1)_T=(v_T,1)_T. \tag{2.6b}$$

It can be checked that, for any $v\in H^1(T)$, $p_T^{k+1}(\underline{I}_T^k v)$ coincides with the oblique elliptic projection of v on $\mathbb{P}^{k+1}(T)$; see [3, §3.1.2].

The global space of discrete unknowns is defined as

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

and for a generic vector of discrete unknowns $\underline{v}_h\in\underline{U}_h^k$, we denote its restriction to T by $\underline{v}_T:=(v_T,(v_F)_{F\in\mathcal{F}_T})\in\underline{U}_T^k$. We also define $\underline{U}_{h,0}^k$ as the subset of $\underline{U}_h^k$ with boundary face unknowns equal to zero.

2.4. HHO discretization of the model problem. The global bilinear form $a_h\colon\underline{U}_h^k\times\underline{U}_h^k\to\mathbb{R}$ is assembled from elementary contributions as follows:

$$a_h(\underline{u}_h,\underline{v}_h):=\sum_{T\in\mathcal{T}_h}a_T(\underline{u}_T,\underline{v}_T),$$

where for all $T\in\mathcal{T}_h$, the local bilinear form $a_T\colon\underline{U}_T^k\times\underline{U}_T^k\to\mathbb{R}$ is defined as

$$a_T(\underline{u}_T,\underline{v}_T):=(\mathbf{K}_T\nabla p_T^{k+1}\underline{u}_T,\nabla p_T^{k+1}\underline{v}_T)_T+s_T(\underline{u}_T,\underline{v}_T). \tag{2.7}$$

The first term is responsible for consistency while the second, involving the bilinear form $s_T\colon \underline{U}_T^k \times \underline{U}_T^k \to \mathbb{R}$, is required for stability. For all $T \in \mathcal{T}_h$, s_T is designed to depend on its arguments only through the *difference operators* δ_T^k and $(\delta_{TF}^k)_{F\in\mathcal{F}_T}$ defined for all $\underline{v}_T \in \underline{U}_T^k$ as

$$(\delta_T^k\underline{v}_T, (\delta_{TF}^k\underline{v}_T)_{F\in\mathcal{F}_T}) := \underline{I}_T^k(p_T^{k+1}\underline{v}_T) - \underline{v}_T.$$

These operators capture the higher-order correction that the operator p_T^{k+1} adds to the respective L^2-projections of a function on the cell and faces. A classical expression for s_T is the following:

$$s_T(\underline{v}_T, \underline{w}_T) := \sum_{F\in\mathcal{F}_T} \frac{K_{TF}}{h_F}((\delta_{TF}^k - \delta_T^k)\underline{v}_T, (\delta_{TF}^k - \delta_T^k)\underline{w}_T)_F,$$

where $K_{TF} := \mathbf{K}_T\mathbf{n}_{TF}\cdot\mathbf{n}_{TF}$. Note that other expressions are possible but will not be considered here. For more details about the stabilization, the reader can refer to [3, §2.1.4].

The global discrete problem reads

$$\text{Find } \underline{u}_h \in \underline{U}_{h,0}^k \text{ such that } a_h(\underline{u}_h, \underline{v}_h) = \sum_{T\in\mathcal{T}_h}(f, v_T)_T \quad \forall \underline{v}_h \in \underline{U}_{h,0}^k. \tag{2.8}$$

2.5. Assembly and static condensation. The local contributions corresponding to the representations, in the selected basis for $\underline{U}_{h,0}^k$, of the bilinear form a_T (cf. (2.7)) and of the linear form $\mathbb{P}^k(T) \ni v_T \mapsto (f, v_T)_T \in \mathbb{R}$ correspond, respectively, to the matrix $\mathbf{A}_T$ and to the vector $\mathbf{B}_T$ such that

$$\mathbf{A}_T := \begin{pmatrix} \mathbf{A}_{TT} & \mathbf{A}_{T\mathcal{F}_T} \\ \mathbf{A}_{\mathcal{F}_T T} & \mathbf{A}_{\mathcal{F}_T\mathcal{F}_T} \end{pmatrix}, \qquad \mathbf{B}_T := \begin{pmatrix} \mathbf{b}_T \\ \mathbf{0} \end{pmatrix}, \tag{2.9}$$

in which the unknowns have been numbered so that cell unknowns come first and face unknowns come last. After assembling the local matrices, we end up with a global linear system of the form

$$\begin{pmatrix} \mathbf{A}_{\mathcal{T}_h\mathcal{T}_h} & \mathbf{A}_{\mathcal{T}_h\mathcal{F}_h^{\mathrm{I}}} \\ \mathbf{A}_{\mathcal{F}_h^{\mathrm{I}}\mathcal{T}_h} & \mathbf{A}_{\mathcal{F}_h^{\mathrm{I}}\mathcal{F}_h^{\mathrm{I}}} \end{pmatrix} \begin{pmatrix} \mathbf{v}_{\mathcal{T}_h} \\ \mathbf{v}_{\mathcal{F}_h^{\mathrm{I}}} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{\mathcal{T}_h} \\ \mathbf{0} \end{pmatrix}, \tag{2.10}$$

where the unknowns corresponding to the boundary faces have been eliminated by strongly enforcing the Dirichlet conditions, hence the notation $\mathcal{F}_h^{\mathrm{I}}$, denoting the subset of interior faces. Because cell-DoFs are entirely decoupled from neighbouring cells, $\mathbf{A}_{\mathcal{T}_h\mathcal{T}_h}$ is block-diagonal, therefore inexpensive to invert. The static condensation process takes advantage of this property by locally eliminating the cell-DoFs: it goes by expressing $\mathbf{v}_{\mathcal{T}_h}$ in terms of $\mathbf{v}_{\mathcal{F}_h^{\mathrm{I}}}$ in the first equation of (2.10):

$$\mathbf{v}_{\mathcal{T}_h} = \mathbf{A}_{\mathcal{T}_h\mathcal{T}_h}^{-1}(\mathbf{b}_{\mathcal{T}_h} - \mathbf{A}_{\mathcal{T}_h\mathcal{F}_h^{\mathrm{I}}}\mathbf{v}_{\mathcal{F}_h^{\mathrm{I}}}), \tag{2.11}$$

and then replacing $\mathbf{v}_{\mathcal{T}_h}$ with its expression (2.11) in the second equation:

$$(\mathbf{A}_{\mathcal{F}_h^{\mathrm{I}}\mathcal{F}_h^{\mathrm{I}}} - \mathbf{A}_{\mathcal{F}_h^{\mathrm{I}}\mathcal{T}_h}\mathbf{A}_{\mathcal{T}_h\mathcal{T}_h}^{-1}\mathbf{A}_{\mathcal{T}_h\mathcal{F}_h^{\mathrm{I}}})\mathbf{v}_{\mathcal{F}_h^{\mathrm{I}}} = -\mathbf{A}_{\mathcal{F}_h^{\mathrm{I}}\mathcal{T}_h}\mathbf{A}_{\mathcal{T}_h\mathcal{T}_h}^{-1}\mathbf{b}_{\mathcal{T}_h}, \tag{2.12}$$

thus yielding a smaller system, involving only face unknowns. The main advantage of this technique is the reduction of the problem size, especially at high-order.

3. Multigrid algorithm. In this section, we present a geometric multigrid algorithm to efficiently solve the condensed system (2.12). The method we propose respects the unknowns of the condensed system by maintaining face-defined functions at every level, and works in synergy with the discretization through intergrid transfer operators based on the mathematical operators used to formulate the HHO problem. Moreover, the algorithm is not restricted to the lowest order, it inherently manages any arbitrary order of approximation without resorting to an additional p-multigrid, which, in practice, can be seen as a valuable reduction of the implementation cost.

3.1. Coarsening strategy. The levels of the multigrid method are decreasingly numbered from L to 1, L being the finest and 1 the coarsest. Relatively to those levels, we consider a hierarchy of nested polyhedral meshes $(\mathcal{T}_\ell)_{\ell=1..L}$, which we assume to successively coarsen not only elements, but also faces. Standard coarsening of structured Cartesian and triangular meshes, as well as unstructured meshes obtained from successive refinements of an initial coarse mesh by a structured refinement method fall under the scope of this assumption; examples of admissible coarsening strategies are illustrated in Figure 3.1. Requiring the coarsening of the faces is justified by our algorithm being face-defined at every level. Indeed, the smoother applies to faces the same way it applies to elements in a classical element-defined multigrid method: once the high frequencies of the error have been annihilated on the fine mesh, the smoother requires coarser elements to reach the low frequencies on the coarse mesh. Likewise, faces need to be coarsened as well. The consequence of a face not being coarsened between a fine and a coarse mesh would be to keep the smoother working on the same range of frequencies, leaving it unable to efficiently reduce the lowest ones.

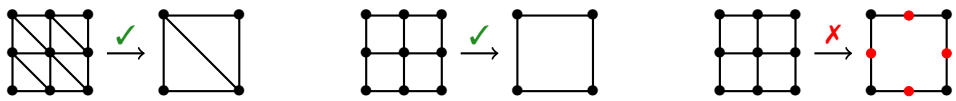

Fig. 3.1: Coarsening examples. The first two are admissible, whereas the third one is not: edges have been removed, but none of the remaining ones has been coarsened.

In order to keep the diffusion coefficient piecewise constant inside each coarse element, we also take the assumption that the mesh hierarchy does not agglomerate elements across discontinuities of the coefficient. For every mesh $\mathcal{T}_\ell$, we denote by $\mathcal{F}_\ell$ the corresponding set of faces. Given an element $T \in \mathcal{T}_\ell$, $\mathcal{F}_T$ still denotes the set of faces in $\mathcal{F}_\ell$ that lie on the boundary of T. Reciprocally, given a face $F \in \mathcal{F}_\ell$, we denote by $\mathcal{T}_F$ the set of elements T such that $F \in \mathcal{F}_T$. (Note that $\mathrm{card}(\mathcal{T}_F) = 2$ for interior faces and $\mathrm{card}(\mathcal{T}_F) = 1$ for boundary faces.)

3.2. Approximation spaces. We consider the same polynomial degree $k \in \mathbb{N}$ on the faces of each level. For all $\ell \in \{1, \dots, L\}$, we consider the approximation space M_ℓ defined as the broken polynomial space of total degree at most k on the mesh skeleton:

$$M_\ell := \mathbb{P}^k(\mathcal{F}_\ell) := \{v_{\mathcal{F}_\ell} \in L^2(\mathcal{F}_\ell) \mid v_{\mathcal{F}_\ell}|_F \in \mathbb{P}^k(F) \quad \forall F \in \mathcal{F}_\ell\}.$$

Additionally, we define the *higher-order* broken space on the mesh itself:

$$V_\ell := \mathbb{P}^{k+1}(\mathcal{T}_\ell) := \{v_{\mathcal{T}_\ell} \in L^2(\mathcal{T}_\ell) \mid v_{\mathcal{T}_\ell}|_T \in \mathbb{P}^{k+1}(T) \quad \forall T \in \mathcal{T}_\ell\}.$$

3.3. Prolongation. We consider two successive levels ℓ (fine) and $\ell-1$ (coarse). In this algorithm, *faces* support the functions at every level. To prolongate a coarse function onto the fine faces, including some that are not kept on the coarse mesh, we propose an intermediary step that passes through the cells (Figure 3.2). Following this idea, the prolongation operator $P\colon M_{\ell-1} \to M_\ell$ is defined as the composition

$$P = I_{V_{\ell-1}}^{M_\ell} \circ I_{M_{\ell-1}}^{V_{\ell-1}}, \tag{3.1}$$

where $I_{M_{\ell-1}}^{V_{\ell-1}}\colon M_{\ell-1} \to V_{\ell-1}$ reconstructs a coarse polynomial of degree $k+1$ defined on the cells from face unknowns; $I_{V_{\ell-1}}^{M_\ell}\colon V_{\ell-1} \to M_\ell$ computes a trace of degree k on the fine faces of the cell-defined polynomial of degree $k+1$.

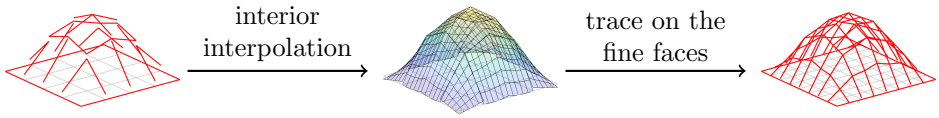

Fig. 3.2: Prolongation from coarse to fine edges.

3.3.1. $I_{M_\ell}^{V_\ell}$: from faces to cells. This operator is at the core of the algorithm and is what makes it original. Here, we propose to take advantage of the local reconstruction operator p_T^{k+1} defined by (2.6). We denote by $v_{\mathcal{F}_\ell} \in M_\ell$ the operand of $I_{M_\ell}^{V_\ell}$ and we define $v_F := v_{\mathcal{F}_\ell}|_F$ for all $F \in \mathcal{F}_\ell$. Let $T \in \mathcal{T}_\ell$. Applying p_T^{k+1} requires a polynomial of degree k on each of the faces, which is given by $(v_F)_{F\in\mathcal{F}_T}$, as well as a polynomial of degree k on the cell. To obtain the latter, which we denote by $v_T \in \mathbb{P}^k(T)$, we reverse the static condensation performed during the assembly step to recover the cell-based unknowns. Let $\mathbf{v}_T$ and $(\mathbf{v}_F)_{F\in\mathcal{F}_T}$ be the respective representations of v_T and $(v_F)_{F\in\mathcal{F}_T}$ as vectors of coefficients in the chosen polynomial bases. $\mathbf{v}_T$ is then given by the local expression of equation (2.11),

$$\mathbf{v}_T := -\mathbf{A}_{TT}^{-1}\mathbf{A}_{T\mathcal{F}_T}\mathbf{v}_{\mathcal{F}_T}, \tag{3.2}$$

where $\mathbf{A}_{TT}$ and $\mathbf{A}_{T\mathcal{F}_T}$ are blocks of the local matrix defined in (2.9), and $\mathbf{v}_{\mathcal{F}_T}$ is the vector of coefficients that concatenates $(\mathbf{v}_F)_{F\in\mathcal{F}_T}$. Note the absence of the local right-hand side contribution $\mathbf{b}_T$, yet present in (2.11). This is justified by the operator being part of an intergrid operator in a multigrid context, and therefore applied to *error* vectors (as opposed to solution vectors), which do not carry affine information. Once v_T is retrieved from (3.2), p_T^{k+1} is finally applied to the hybrid vector $(v_T, (v_F)_{F\in\mathcal{F}_T})$ to yield a polynomial of degree $k+1$ on the cell:

$$(I_{M_\ell}^{V_\ell} v_{\mathcal{F}_\ell})|_T := p_T^{k+1}(v_T, (v_F)_{F\in\mathcal{F}_T}).$$

Figure 3.3 summarizes the process.

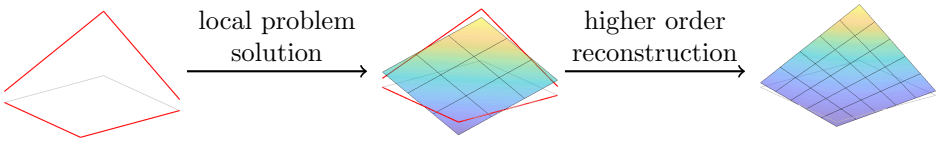

Fig. 3.3: Reconstruction of a polynomial of degree 2 from polynomials of degree 1 on the four edges of a 2D square element.

3.3.2. $I_{V_{\ell-1}}^{M_\ell}$: from cells to faces. For $F \in \mathcal{F}_\ell$ and $v \in V_{\ell-1}$, $(I_{V_{\ell-1}}^{M_\ell} v)|_F$ is built as the weighted average of the the traces of v on both sides of F. The operator $I_{V_{\ell-1}}^{M_\ell}$ is then constructed locally from the local L^2-projectors on the faces. For all $F \in \mathcal{F}_\ell$, each element $T \in \mathcal{T}_F$ of which F is a face adds a contribution:

$$I_{V_{\ell-1}}^{M_\ell} := \sum_{F\in\mathcal{F}_\ell} \sum_{T\in\mathcal{T}_F} w_{TF}\, \pi_F^k,$$

where π_F^k is the L^2-projector on $\mathbb{P}^k(F)$ and $(w_{TF})_{T\in\mathcal{T}_h, F\in\mathcal{F}_T}$ is a family of scalar values deriving from the enforcement of two constraints: first, we require the operator to preserve constant functions (i.e. a cell-defined constant function must result, after application of $I_{V_{\ell-1}}^{M_\ell}$, in the face-defined constant function of same value), which leads to the condition

$$\forall F \in \mathcal{F}_\ell, \qquad \sum_{T\in\mathcal{T}_F} w_{TF} = 1. \tag{3.3}$$

Next, we require the element contribution on each side of F to be weighted proportionally to its diffusion coefficient. For all $T_1, T_2 \in \mathcal{T}_F$, this translates to

$$\frac{w_{T_1F}}{w_{T_2F}} = \frac{K_{T_1F}}{K_{T_2F}}, \tag{3.4}$$

where we recall that, for all $T \in \mathcal{T}_F$, $K_{TF} := \mathbf{K}_T\mathbf{n}_{TF}\cdot\mathbf{n}_{TF}$. Enforcing both constraints (3.3) and (3.4) finally imposes, for all $F \in \mathcal{F}_\ell$,

$$w_{TF} := \frac{K_{TF}}{\sum_{T'\in\mathcal{T}_F} K_{T'F}} \quad \forall T \in \mathcal{T}_F.$$

3.4. Multigrid components. The prolongation operator P is defined by (3.1). The restriction operator R is defined as the adjoint of P in the usual way. Interpreted algebraically as matrices, $R = P^T$. Note that $I_{V_{\ell-1}}^{M_\ell}$ does not distinguish the fine faces kept on the coarse grid from those removed; consequently, the polynomials on coarse faces are not transferred identically to the fine grid, but instead take on new values coming from the (weighted) average of the reconstructed cell-polynomials on each side. The alternative way of prolongating coarse functions from coarse faces to their respective identical fine faces, namely keeping them unchanged, has also been tested and does not yield a scalable algorithm. This observation is consistent with the fact that solving the local problems brings additional information that the coarse polynomials do not possess. In addition, the reconstruction in a higher degree also results in higher accuracy in the case where two fine faces are agglomerated into a single coarse one: the polynomial of degree $k+1$ on the coarse face can induce two different polynomials of degree k on the two corresponding fine faces, which could not happen if the reconstruction was only of degree k.

The coarse grid operator at level $\ell-1$ can be chosen either as the discretized operator on the respective coarse mesh, or as the Galerkin construction: $A_{\ell-1} := RA_\ell P$. The numerical tests show equivalent performances.

In order to relax together the DoFs related to the same polynomial function, block versions of standard fixed-point smoothers are chosen, whose block size corresponds to the number of DoFs per face. Only block Gauss Seidel has been considered in the experiments.

4. Numerical results.

4.1. Experimental setup. The numerical tests have been performed on the diffusion problem (2.1) in the domain $\Omega := (0,1)^d$, $d \in \{2,3\}$. The source function f is chosen so that the analytical solution of the homogeneous problem corresponds to a sine function. Given an integer $N \geq 0$, the domain is discretized by a Cartesian grid composed of N^d square/cubic elements of width $h := \frac{1}{N}$. In what follows, k denotes the polynomial degree on the faces (meaning that the HHO method ultimately yields an approximation of degree $k+1$). Our multigrid algorithm is used to solve the statically condensed linear system (2.12). When the number of levels is not fixed, the mesh is successively coarsened until the coarse system reaches a size with less than 1000 unknowns. On the coarsest level, the system is solved by a direct solver. Operators on coarse levels are the discretized operators on the respective coarse meshes. The smoother is a block Gauss Seidel method, in which the block size corresponds to the number of face-DoFs. Unless stated otherwise, the V(1,1)-cycle is used: 1 sweep as pre-smoothing, 1 sweep in the reverse order as post-smoothing. The stopping criterion is set to $\|r\|_2/\|b\|_2 < 10^{-8}$, where r denotes the residual vector, b the right-hand side of the linear system and $\|\cdot\|_2$ the Euclidean norm on the space of coefficients in the linear combinations obtained with respect to the choice of the L^2-orthogonal Legendre polynomial basis.

4.2. Homogeneous diffusion. The diffusion tensor is constant across the domain and equals the identity matrix. Figure 4.1 summarizes the scalability results. It shows that the algorithm converges at a rate that is almost independent of the mesh size and the number of levels. Although the number of iterations increases moderately with the polynomial order of the approximation, the algorithm still exhibits the same desirable properties in higher orders.

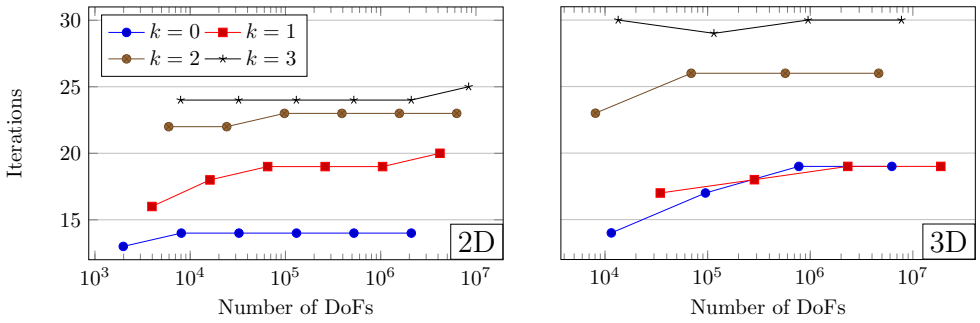

Fig. 4.1: Number of iterations to achieve convergence for the homogeneous problem.

In Figure 4.2, the problem is solved by the Conjugate Gradient algorithm, where our multigrid method is used as preconditioner: it shows the same scalability properties, as well as a milder dependency to the polynomial degree.

Figure 4.3 compares the performance, measured in flops and CPU time, of different multigrid cycles on a 3D test problem with $k = 0$. In the left plot, the numerical values have been obtained by taking the theoretical computational work (in flops) of the multigrid algorithm, using the following simplifying rules: (i) the asymptotic value of the work count is used, meaning that only the largest power term (in the matrix size or non-zero entries) is kept; (ii) the work of the direct solver on the coarsest grid is neglected. The comparison in CPU time plotted on the right-hand side shows

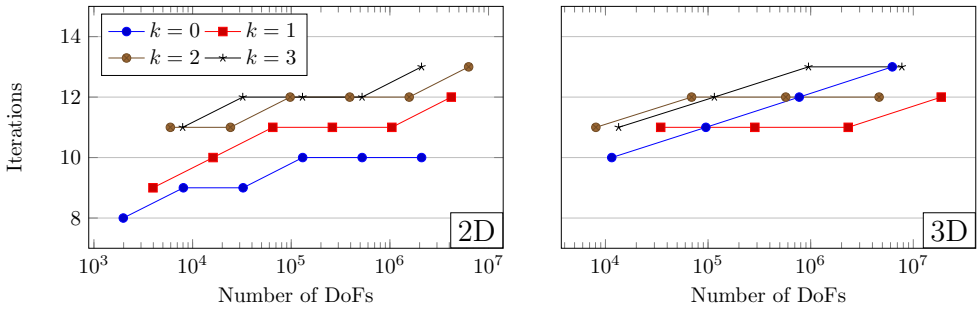

Fig. 4.2: Number of iterations to achieve convergence for the homogeneous problem solved by the Conjugate Gradient preconditioned by our multigrid.

an equivalent ranking. V- and W-cycles have been tested; however, we stress that the W-cycle shows the same convergence rate as the V-cycle, while being costlier by definition; hence it is not presented here. Now, regarding the number of smoothing steps, we can say that V(1,2) and V(2,2) equivalently seem the most efficient cycles, while V(1,1) is about 30% costlier for the same result. In terms of convergence rate, V(1,2) and V(2,2) converge in about half the number of iterations required in V(1,1).

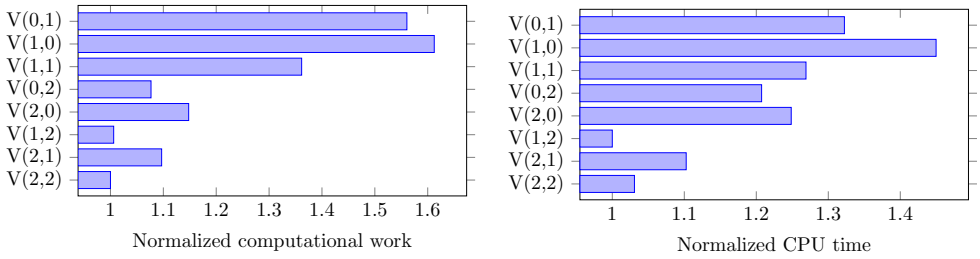

Fig. 4.3: Cycle comparison on the 3D test problem $N = 64$, $k = 0$. In each plot, the values are normalized by the lowest one.

4.3. Heterogeneous diffusion. The domain is split into four quadrants. The heterogeneity pattern follows the diagonals, so that quadrants 1 and 3 (resp. 2 and 4) are homogeneous. On each homogeneous part (indexed by $i = 1, 2$), the diffusion tensor is defined as $\mathbf{K}_i := \kappa_i \mathbf{I}_d$, where κ_i is a positive scalar constant and $\mathbf{I}_d$ denotes the identity matrix of size d. A first test consists in observing the convergence rate according to the jump in the coefficient, i.e. for varying values of the ratio κ_1/κ_2. Experiments have been conducted with a jump ranging from 1 to 10^8. The results demonstrate perfect robustness of the algorithm with respect to heterogeneity; regardless of the magnitude the jump, the convergence rate remains unchanged and matches the homogeneous case.

In [6], Kellogg published the analytical solution of a specific case of such a configuration for a source function $f \equiv 0$ and non-homogeneous Dirichlet boundary conditions. The solution exhibits a singularity at the center of the square, and is known to be of class $H^{1+\epsilon}$, $0 < \epsilon < 1$. Since the strength of the singularity and thus the regu-

larity ϵ can be adjusted by choosing the size of the jump in coefficients, this problem is often used to benchmark discretizations and solvers. The parameters of our Kellogg test problem are set such that $\epsilon = 0.1$ (the jump $\kappa_2/\kappa_1 \approx 161$). Figure 4.4 presents on the left the graphical representation of the solution; on the right, the numerical results show the scalability and robustness of the multigrid solver with respect to the number of DoFs.

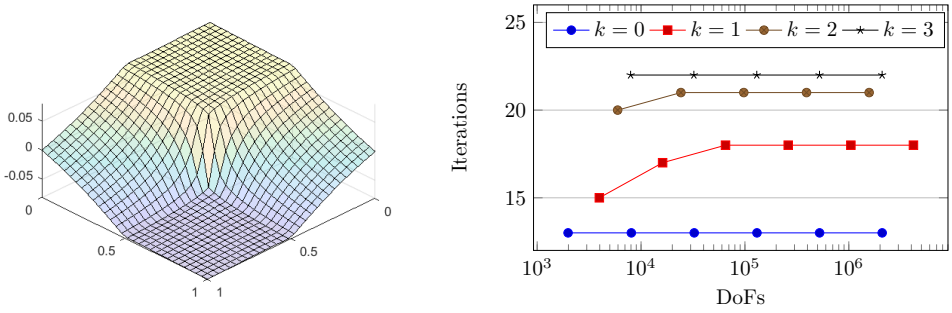

Fig. 4.4: On the left: analytical solution of the Kellogg problem. On the right: Number of multigrid iterations to achieve convergence for a growing number of DoFs.

5. Conclusion. The multigrid solver proposed and developed in this article comes up as fast, scalable and robust to heterogeneity for elliptic problems discretized in HHO. Moreover, none of these desirable properties suffers from raising the order of approximation. Although no assumption is made concerning the mesh structure, imposing that the faces be also coarsened on coarse meshes makes the design of an admissible coarsening strategy for unstructured polyhedral meshes more difficult. Additional complexity must indeed be expected when the faces are not coplanar (resp. colinear in 2D).

REFERENCES

[1] 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 (2016), pp. 635–650, https://doi.org/10.1051/m2an/2015051.

[2] B. Cockburn, O. Dubois, J. Gopalakrishnan, and S. Tan, *Multigrid for an HDG method*, IMA Journal of Numerical Analysis, 34 (2014), pp. 1386–1425.

[3] D. A. Di Pietro and J. Droniou, *The Hybrid High-Order method for polytopal meshes*, no. 19 in Modeling, Simulation and Application, Springer International Publishing, 2020.

[4] D. A. Di Pietro and A. Ern, *A hybrid high-order locking-free method for linear elasticity on general meshes*, Comput. Meth. Appl. Mech. Engrg., 283 (2015), pp. 1–21, https://doi.org/10.1016/j.cma.2014.09.009.

[5] G. Kanschat, *Robust smoothers for high-order discontinuous Galerkin discretizations of advection–diffusion problems*, Journal of Computational and Applied Mathematics, 218 (2008), pp. 53–60.

[6] R. B. Kellogg, *SINGULARITIES IN INTERFACE PROBLEMS*, in Numerical Solution of Partial Differential Equations–II, B. Hubbard, ed., Academic Press, Jan. 1971, pp. 351–400.

[7] L. N. Olson and J. B. Schroder, *Smoothed aggregation multigrid solvers for high-order discontinuous Galerkin methods for elliptic problems*, Journal of Computational Physics, 230 (2011), pp. 6959–6976.

[8] T. Wildey, S. Muralikrishnan, and T. Bui-Thanh, *Unified geometric multigrid algorithm for hybridized high-order finite element methods*, arXiv:1811.09909 [math], (2018).