



**HAL**  
open science

## An h-multigrid method for Hybrid High-Order discretizations

Pierre Matalon, Daniele Antonio Di Pietro, Paul Mycek, Ulrich Rude, Daniel Ruiz

► **To cite this version:**

Pierre Matalon, Daniele Antonio Di Pietro, Paul Mycek, Ulrich Rude, Daniel Ruiz. An h-multigrid method for Hybrid High-Order discretizations. 2020. hal-02434411v1

**HAL Id: hal-02434411**

**<https://hal.science/hal-02434411v1>**

Preprint submitted on 10 Jan 2020 (v1), last revised 18 May 2021 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinee au depot et a la diffusion de documents scientifiques de niveau recherche, publies ou non, emanant des tablissements d'enseignement et de recherche francais ou etrangers, des laboratoires publics ou privs.



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  $\Omega$  a bounded polyhedral domain of  $\mathbb{R}^d$ .  $\Omega$  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.  $H_0^1(X)$  defines the subspace of  $H^1(X)$  whose functions vanish on the boundary  $\partial X$  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  $\ell$ ,  $\ell \in \mathbb{N}$ .

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

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

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

$$(2.2) \quad \text{Find } u \in H_0^1(\Omega) \text{ such that } a(u, v) = \int_{\Omega} f v \quad \forall v \in H_0^1(\Omega),$$

where the bilinear form  $a: H^1(\Omega) \times H^1(\Omega) \rightarrow \mathbb{R}$  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  $\Omega$  in such a way that the diffusion tensor is constant inside each element, and we denote  $\mathbf{K}_T := \mathbf{K}|_T$  for all  $T \in \mathcal{T}_h$ .

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

$$87 \quad (2.3) \quad \sum_{T \in \mathcal{T}_h} (\mathbf{K}_T \nabla u, \nabla v)_T = \sum_{T \in \mathcal{T}_h} (f, v)_T \quad \forall v \in H_0^1(\Omega).$$

88 **2.3. Discrete spaces and operators.** We briefly recall the standard HHO  
 89 discretization of problem (2.3). For a more comprehensive presentation, see [3, §3.1].

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

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

$$95 \quad (2.4) \quad \underline{U}_T^k := \{ \underline{v}_T := (v_T, (v_F)_{F \in \mathcal{F}_T}) \mid v_T \in \mathbb{P}^k(T), v_F \in \mathbb{P}^k(F) \quad \forall F \in \mathcal{F}_T \}.$$

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

$$99 \quad (2.5) \quad \underline{I}_T^k v := (\pi_T^k v, (\pi_F^k v)_{F \in \mathcal{F}_T}),$$

100 where, for any  $X \in \mathcal{T}_h \cup \mathcal{F}_h$ ,  $\pi_X^k: L^2(X) \rightarrow \mathbb{P}^k(X)$  denotes the  $L^2$ -orthogonal projector  
 101 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-  
 102 degree approximation of  $v$  can be reconstructed inside  $T$ . This is achieved by means  
 103 of the *local potential reconstructor*  $p_T^{k+1}: \underline{U}_T^k \rightarrow \mathbb{P}^{k+1}(T)$  such that, for all  $\underline{v}_T :=$   
 104  $(v_T, (v_F)_{F \in \mathcal{F}_T}) \in \underline{U}_T^k$ ,  $p_T^{k+1} \underline{v}_T$  is the unique polynomial of degree at most  $k+1$   
 105 verifying

$$106 \quad (2.6a) \quad \begin{cases} (\mathbf{K}_T \nabla p_T^{k+1} \underline{v}_T, \nabla w)_T = -(v_T, \nabla \cdot (\mathbf{K}_T \nabla w))_T + \sum_{F \in \mathcal{F}_T} (v_F, \mathbf{K}_T \nabla w \cdot \mathbf{n}_{TF})_F \\ \forall w \in \mathbb{P}^{k+1}(T), \end{cases}$$

$$107 \quad (2.6b) \quad (p_T^{k+1} \underline{v}_T, 1)_T = (v_T, 1)_T.$$

107 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  
 108 elliptic projection of  $v$  on  $\mathbb{P}^{k+1}(T)$ ; see [3, §3.1.2].

109 The global space of discrete unknowns is defined as

$$110 \quad \underline{U}_h^k := \{ \underline{v}_h := ((v_T)_{T \in \mathcal{T}_h}, (v_F)_{F \in \mathcal{F}_h}) \mid v_T \in \mathbb{P}^k(T) \quad \forall T \in \mathcal{T}_h, \\ v_F \in \mathbb{P}^k(F) \quad \forall F \in \mathcal{F}_h \},$$

111 and for a generic vector of discrete unknowns  $\underline{v}_h \in \underline{U}_h^k$ , we denote its restriction  
 112 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  
 113 boundary face unknowns equal to zero.

114 **2.4. HHO discretization of the model problem.** The global bilinear form  
 115  $a_h: \underline{U}_h^k \times \underline{U}_h^k \rightarrow \mathbb{R}$  is assembled from elementary contributions as follows:

$$116 \quad a_h(\underline{u}_h, \underline{v}_h) := \sum_{T \in \mathcal{T}_h} a_T(\underline{u}_T, \underline{v}_T),$$

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

$$118 \quad (2.7) \quad a_T(\underline{u}_T, \underline{v}_T) := (\mathbf{K}_T \nabla p_T^{k+1} \underline{u}_T, \nabla p_T^{k+1} \underline{v}_T)_T + s_T(\underline{u}_T, \underline{v}_T).$$

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

$$123 \quad (\delta_T^k \underline{v}_T, (\delta_{TF}^k \underline{v}_T)_{F \in \mathcal{F}_T}) := \underline{I}_T^k(p_T^{k+1} \underline{v}_T) - \underline{v}_T.$$

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

$$127 \quad s_T(\underline{v}_T, \underline{w}_T) := \sum_{F \in \mathcal{F}_T} \frac{K_{TF}}{h_F} ((\delta_{TF}^k - \delta_T^k) \underline{v}_T, (\delta_{TF}^k - \delta_T^k) \underline{w}_T)_F,$$

128 where  $K_{TF} := \mathbf{K}_T \mathbf{n}_{TF} \cdot \mathbf{n}_{TF}$ . Note that other expressions are possible but will not  
 129 be considered here. For more details about the stabilization, the reader can refer to  
 130 [3, §2.1.4].

131 The global discrete problem reads

$$132 \quad (2.8) \quad \text{Find } \underline{u}_h \in \underline{U}_{h,0}^k \text{ such that } a_h(\underline{u}_h, \underline{v}_h) = \sum_{T \in \mathcal{T}_h} (f, v_T)_T \quad \forall \underline{v}_h \in \underline{U}_{h,0}^k.$$

133 **2.5. Assembly and static condensation.** The local contributions correspond-  
 134 ing to the representations, in the selected basis for  $\underline{U}_{h,0}^k$ , of the bilinear form  $a_T$  (cf.  
 135 (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  
 136 the matrix  $\mathbf{A}_T$  and to the vector  $\mathbf{B}_T$  such that

$$137 \quad (2.9) \quad \mathbf{A}_T := \begin{pmatrix} \mathbf{A}_{TT} & \mathbf{A}_{T\mathcal{F}_T} \\ \mathbf{A}_{\mathcal{F}_T T} & \mathbf{A}_{\mathcal{F}_T \mathcal{F}_T} \end{pmatrix}, \quad \mathbf{B}_T := \begin{pmatrix} \mathbf{b}_T \\ \mathbf{0} \end{pmatrix},$$

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

$$141 \quad (2.10) \quad \begin{pmatrix} \mathbf{A}_{\mathcal{T}_h \mathcal{T}_h} & \mathbf{A}_{\mathcal{T}_h \mathcal{F}_h^1} \\ \mathbf{A}_{\mathcal{F}_h^1 \mathcal{T}_h} & \mathbf{A}_{\mathcal{F}_h^1 \mathcal{F}_h^1} \end{pmatrix} \begin{pmatrix} \mathbf{v}_{\mathcal{T}_h} \\ \mathbf{v}_{\mathcal{F}_h^1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{\mathcal{T}_h} \\ \mathbf{0} \end{pmatrix},$$

142 where the unknowns corresponding to the boundary faces have been eliminated by  
 143 strongly enforcing the Dirichlet conditions, hence the notation  $\mathcal{F}_h^1$ , denoting the subset  
 144 of interior faces. Because cell-DoFs are entirely decoupled from neighbouring cells,  
 145  $\mathbf{A}_{\mathcal{T}_h \mathcal{T}_h}$  is block-diagonal, therefore inexpensive to invert. The static condensation  
 146 process takes advantage of this property by locally eliminating the cell-DoFs: it goes  
 147 by expressing  $\mathbf{v}_{\mathcal{T}_h}$  in terms of  $\mathbf{v}_{\mathcal{F}_h^1}$  in the first equation of (2.10):

$$148 \quad (2.11) \quad \mathbf{v}_{\mathcal{T}_h} = \mathbf{A}_{\mathcal{T}_h \mathcal{T}_h}^{-1} (\mathbf{b}_{\mathcal{T}_h} - \mathbf{A}_{\mathcal{T}_h \mathcal{F}_h^1} \mathbf{v}_{\mathcal{F}_h^1}),$$

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

$$150 \quad (2.12) \quad (\mathbf{A}_{\mathcal{F}_h^1 \mathcal{F}_h^1} - \mathbf{A}_{\mathcal{F}_h^1 \mathcal{T}_h} \mathbf{A}_{\mathcal{T}_h \mathcal{T}_h}^{-1} \mathbf{A}_{\mathcal{T}_h \mathcal{F}_h^1}) \mathbf{v}_{\mathcal{F}_h^1} = -\mathbf{A}_{\mathcal{F}_h^1 \mathcal{T}_h} \mathbf{A}_{\mathcal{T}_h \mathcal{T}_h}^{-1} \mathbf{b}_{\mathcal{T}_h},$$

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

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

161 **3.1. Coarsening strategy.** The levels of the multigrid method are decreasingly  
 162 numbered from  $L$  to 1,  $L$  being the finest and 1 the coarsest. Relatively to those levels,  
 163 we consider a hierarchy of nested polyhedral meshes  $(\mathcal{T}_\ell)_{\ell=1..L}$ , which we assume to  
 164 successively coarsen not only elements, but also faces. Standard coarsening of struc-  
 165 tured Cartesian and triangular meshes, as well as unstructured meshes obtained from  
 166 successive refinements of an initial coarse mesh by a structured refinement method  
 167 fall under the scope of this assumption; examples of admissible coarsening strategies  
 168 are illustrated in Figure 3.1. Requiring the coarsening of the faces is justified by our  
 169 algorithm being face-defined at every level. Indeed, the smoother applies to faces the  
 170 same way it applies to elements in a classical element-defined multigrid method: once  
 171 the high frequencies of the error have been annihilated on the fine mesh, the smoother  
 172 requires coarser elements to reach the low frequencies on the coarse mesh. Likewise,  
 173 faces need to be coarsened as well. The consequence of a face not being coarsened  
 174 between a fine and a coarse mesh would be to keep the smoother working on the same  
 175 range of frequencies, leaving it unable to efficiently reduce the lowest ones.

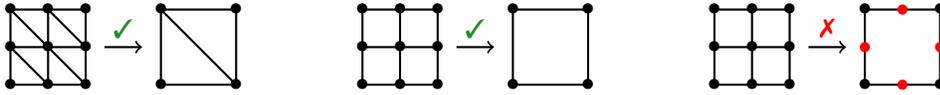


Fig. 3.1: Coarsening examples. The first two are admissible, whereas the third one is not: edges have been removed, but none of the remaining ones has been coarsened.

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

183 **3.2. Approximation spaces.** We consider the same polynomial degree  $k \in \mathbb{N}$   
 184 on the faces of each level. For all  $\ell \in \{1, \dots, L\}$ , we consider the approximation space  
 185  $M_\ell$  defined as the broken polynomial space of total degree at most  $k$  on the mesh  
 186 skeleton:

$$187 \quad M_\ell := \mathbb{P}^k(\mathcal{F}_\ell) := \{v_{\mathcal{F}_\ell} \in L^2(\mathcal{F}_\ell) \mid v_{\mathcal{F}_\ell}|_F \in \mathbb{P}^k(F) \quad \forall F \in \mathcal{F}_\ell\}.$$

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

$$189 \quad V_\ell := \mathbb{P}^{k+1}(\mathcal{T}_\ell) := \{v_{\mathcal{T}_\ell} \in L^2(\mathcal{T}_\ell) \mid v_{\mathcal{T}_\ell}|_T \in \mathbb{P}^{k+1}(T) \quad \forall T \in \mathcal{T}_\ell\}.$$

190 **3.3. Prolongation.** We consider two successive levels  $\ell$  (fine) and  $\ell - 1$  (coarse).  
 191 In this algorithm, *faces* support the functions at every level. To prolongate a coarse  
 192 function onto the fine faces, including some that are not kept on the coarse mesh, we  
 193 propose an intermediary step that passes through the cells (Figure 3.2). Following  
 194 this idea, the prolongation operator  $P: M_{\ell-1} \rightarrow M_\ell$  is defined as the composition

$$195 \quad (3.1) \quad P = I_{V_{\ell-1}}^{M_\ell} \circ I_{M_{\ell-1}}^{V_{\ell-1}},$$

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

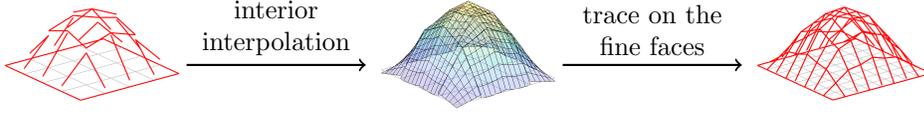


Fig. 3.2: Prolongation from coarse to fine edges.

199 **3.3.1.  $I_{M_\ell}^{V_\ell}$ : from faces to cells.** This operator is at the core of the algorithm  
 200 and is what makes it original. Here, we propose to take advantage of the local re-  
 201 construction operator  $p_T^{k+1}$  defined by (2.6). We denote by  $v_{\mathcal{F}_\ell} \in M_\ell$  the operand of  
 202  $I_{M_\ell}^{V_\ell}$  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  
 203 a polynomial of degree  $k$  on each of the faces, which is given by  $(v_F)_{F \in \mathcal{F}_T}$ , as well  
 204 as a polynomial of degree  $k$  on the cell. To obtain the latter, which we denote by  
 205  $v_T \in \mathbb{P}^k(T)$ , we reverse the static condensation performed during the assembly step  
 206 to recover the cell-based unknowns. Let  $\mathbf{v}_T$  and  $(\mathbf{v}_F)_{F \in \mathcal{F}_T}$  be the respective represen-  
 207 tations of  $v_T$  and  $(v_F)_{F \in \mathcal{F}_T}$  as vectors of coefficients in the chosen polynomial bases.  
 208  $\mathbf{v}_T$  is then given by the local expression of equation (2.11),

$$209 \quad (3.2) \quad \mathbf{v}_T := -\mathbf{A}_{TT}^{-1} \mathbf{A}_{T\mathcal{F}_T} \mathbf{v}_{\mathcal{F}_T},$$

210 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  
 211 vector of coefficients that concatenates  $(\mathbf{v}_F)_{F \in \mathcal{F}_T}$ . Note the absence of the local right-  
 212 hand side contribution  $\mathbf{b}_T$ , yet present in (2.11). This is justified by the operator being  
 213 part of an intergrid operator in a multigrid context, and therefore applied to *error*  
 214 vectors (as opposed to solution vectors), which do not carry affine information. Once  
 215  $v_T$  is retrieved from (3.2),  $p_T^{k+1}$  is finally applied to the hybrid vector  $(v_T, (v_F)_{F \in \mathcal{F}_T})$   
 216 to yield a polynomial of degree  $k + 1$  on the cell:

$$217 \quad (I_{M_\ell}^{V_\ell} v_{\mathcal{F}_\ell})|_T := p_T^{k+1}(v_T, (v_F)_{F \in \mathcal{F}_T}).$$

218 Figure 3.3 summarizes the process.

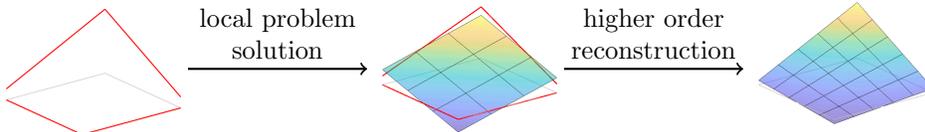


Fig. 3.3: Reconstruction of a polynomial of degree 2 from polynomials of degree 1 on the four edges of a 2D square element.

219 **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  
 220 as the weighted average of the the traces of  $v$  on both sides of  $F$ . The operator  $I_{V_{\ell-1}}^{M_\ell}$   
 221 is then constructed locally from the local  $L^2$ -projectors on the faces. For all  $F \in \mathcal{F}_\ell$ ,  
 222 each element  $T \in \mathcal{T}_F$  of which  $F$  is a face adds a contribution:

$$223 \quad I_{V_{\ell-1}}^{M_\ell} := \sum_{F \in \mathcal{F}_\ell} \sum_{T \in \mathcal{T}_F} w_{TF} \pi_F^k,$$

224 where  $\pi_F^k$  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  
 225 values deriving from the enforcement of two constraints: first, we require the operator  
 226 to preserve constant functions (i.e. a cell-defined constant function must result, after  
 227 application of  $I_{V_{\ell-1}}^{M_\ell}$ , in the face-defined constant function of same value), which leads  
 228 to the condition

$$229 \quad (3.3) \quad \forall F \in \mathcal{F}_\ell, \quad \sum_{T \in \mathcal{T}_F} w_{TF} = 1.$$

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

$$232 \quad (3.4) \quad \frac{w_{T_1 F}}{w_{T_2 F}} = \frac{K_{T_1 F}}{K_{T_2 F}},$$

233 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  
 234 (3.3) and (3.4) finally imposes, for all  $F \in \mathcal{F}_\ell$ ,

$$235 \quad w_{TF} := \frac{K_{TF}}{\sum_{T' \in \mathcal{T}_F} K_{T'F}} \quad \forall T \in \mathcal{T}_F.$$

236 **3.4. Multigrid components.** The prolongation operator  $P$  is defined by (3.1).  
 237 The restriction operator  $R$  is defined as the adjoint of  $P$  in the usual way. Interpreted  
 238 algebraically as matrices,  $R = P^T$ . Note that  $I_{V_{\ell-1}}^{M_\ell}$  does not distinguish the fine  
 239 faces kept on the coarse grid from those removed; consequently, the polynomials on  
 240 coarse faces are not transferred identically to the fine grid, but instead take on new  
 241 values coming from the (weighted) average of the reconstructed cell-polynomials on  
 242 each side. The alternative way of prolongating coarse functions from coarse faces to  
 243 their respective identical fine faces, namely keeping them unchanged, has also been  
 244 tested and does not yield a scalable algorithm. This observation is consistent with  
 245 the fact that solving the local problems brings additional information that the coarse  
 246 polynomials do not possess. In addition, the reconstruction in a higher degree also  
 247 results in higher accuracy in the case where two fine faces are agglomerated into a  
 248 single coarse one: the polynomial of degree  $k + 1$  on the coarse face can induce two  
 249 different polynomials of degree  $k$  on the two corresponding fine faces, which could not  
 250 happen if the reconstruction was only of degree  $k$ .

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

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

258 **4. Numerical results.**

259 **4.1. Experimental setup.** The numerical tests have been performed on the  
 260 diffusion problem (2.1) in the domain  $\Omega := (0, 1)^d$ ,  $d \in \{2, 3\}$ . The source function  
 261  $f$  is chosen so that the analytical solution of the homogeneous problem corresponds  
 262 to a sine function. Given an integer  $N \geq 0$ , the domain is discretized by a Cartesian  
 263 grid composed of  $N^d$  square/cubic elements of width  $h := \frac{1}{N}$ . In what follows,  $k$   
 264 denotes the polynomial degree on the faces (meaning that the HHO method ultimately  
 265 yields an approximation of degree  $k + 1$ ). Our multigrid algorithm is used to solve  
 266 the statically condensed linear system (2.12). When the number of levels is not  
 267 fixed, the mesh is successively coarsened until the coarse system reaches a size with  
 268 less than 1000 unknowns. On the coarsest level, the system is solved by a direct  
 269 solver. Operators on coarse levels are the discretized operators on the respective  
 270 coarse meshes. The smoother is a block Gauss Seidel method, in which the block size  
 271 corresponds to the number of face-DoFs. Unless stated otherwise, the V(1,1)-cycle is  
 272 used: 1 sweep as pre-smoothing, 1 sweep in the reverse order as post-smoothing. The  
 273 stopping criterion is set to  $\|r\|_2/\|b\|_2 < 10^{-8}$ , where  $r$  denotes the residual vector,  $b$   
 274 the right-hand side of the linear system and  $\|\cdot\|_2$  the Euclidean norm on the space  
 275 of coefficients in the linear combinations obtained with respect to the choice of the  
 276  $L^2$ -orthogonal Legendre polynomial basis.

277 **4.2. Homogeneous diffusion.** The diffusion tensor is constant across the do-  
 278 main and equals the identity matrix. Figure 4.1 summarizes the scalability results. It  
 279 shows that the algorithm converges at a rate that is almost independent of the mesh  
 280 size and the number of levels. Although the number of iterations increases moderately  
 281 with the polynomial order of the approximation, the algorithm still exhibits the same  
 282 desirable properties in higher orders.

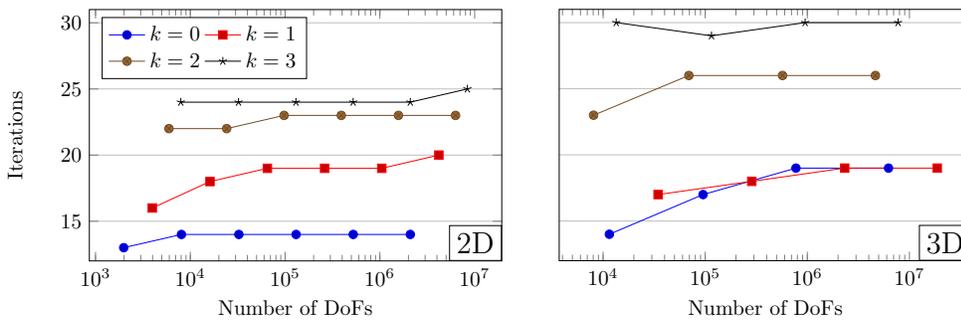


Fig. 4.1: Number of iterations to achieve convergence for the homogeneous problem.

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

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

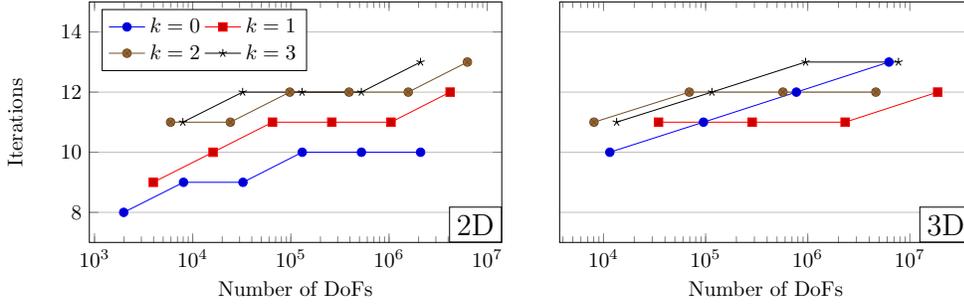


Fig. 4.2: Number of iterations to achieve convergence for the homogeneous problem solved by the Conjugate Gradient preconditioned by our multigrid.

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

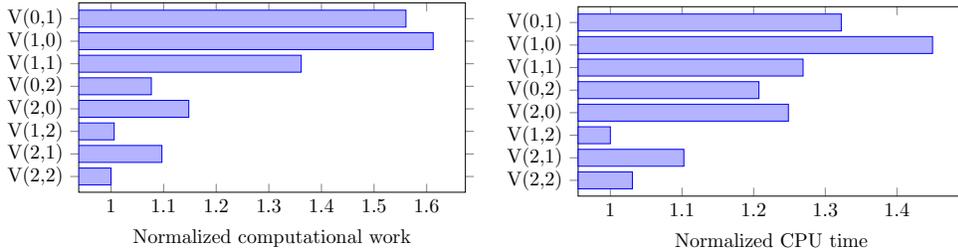


Fig. 4.3: Cycle comparison on the 3D test problem  $N = 64$ ,  $k = 0$ . In each plot, the values are normalized by the lowest one.

299

300 **4.3. Heterogeneous diffusion.** The domain is split into four quadrants. The  
 301 heterogeneity pattern follows the diagonals, so that quadrants 1 and 3 (resp. 2 and  
 302 4) are homogeneous. On each homogeneous part (indexed by  $i = 1, 2$ ), the diffusion  
 303 tensor is defined as  $\mathbf{K}_i := \kappa_i \mathbf{I}_d$ , where  $\kappa_i$  is a positive scalar constant and  $\mathbf{I}_d$  denotes  
 304 the identity matrix of size  $d$ . A first test consists in observing the convergence rate  
 305 according to the jump in the coefficient, i.e. for varying values of the ratio  $\kappa_1/\kappa_2$ .  
 306 Experiments have been conducted with a jump ranging from 1 to  $10^8$ . The results  
 307 demonstrate perfect robustness of the algorithm with respect to heterogeneity; re-  
 308 gardless of the magnitude the jump, the convergence rate remains unchanged and  
 309 matches the homogeneous case.

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

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

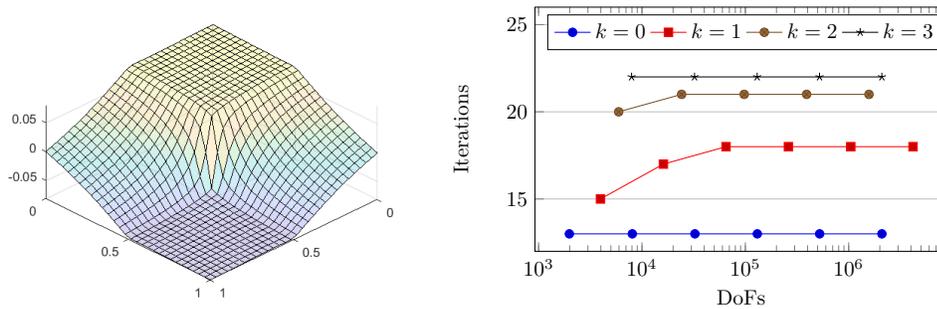


Fig. 4.4: On the left: analytical solution of the Kellogg problem. On the right: Number of multigrid iterations to achieve convergence for a growing number of DoFs.

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

328

## REFERENCES

- 329 [1] B. COCKBURN, D. A. DI PIETRO, AND A. ERN, *Bridging the Hybrid High-Order and Hybridizable*  
 330 *Discontinuous Galerkin methods*, ESAIM: Math. Model Numer. Anal., 50 (2016), pp. 635–  
 331 650, <https://doi.org/10.1051/m2an/2015051>.  
 332 [2] B. COCKBURN, O. DUBOIS, J. GOPALAKRISHNAN, AND S. TAN, *Multigrid for an HDG method*,  
 333 IMA Journal of Numerical Analysis, 34 (2014), pp. 1386–1425.  
 334 [3] D. A. DI PIETRO AND J. DRONIU, *The Hybrid High-Order method for polytopal meshes*, no. 19  
 335 in Modeling, Simulation and Application, Springer International Publishing, 2020.  
 336 [4] D. A. DI PIETRO AND A. ERN, *A hybrid high-order locking-free method for linear elasticity on*  
 337 *general meshes*, Comput. Meth. Appl. Mech. Engrg., 283 (2015), pp. 1–21, [https://doi.org/](https://doi.org/10.1016/j.cma.2014.09.009)  
 338 [10.1016/j.cma.2014.09.009](https://doi.org/10.1016/j.cma.2014.09.009).  
 339 [5] G. KANSCHAT, *Robust smoothers for high-order discontinuous Galerkin discretizations of advec-*  
 340 *tion-diffusion problems*, Journal of Computational and Applied Mathematics, 218 (2008),  
 341 pp. 53–60.  
 342 [6] R. B. KELLOGG, *SINGULARITIES IN INTERFACE PROBLEMS*, in Numerical Solution of  
 343 Partial Differential Equations–II, B. Hubbard, ed., Academic Press, Jan. 1971, pp. 351–400.  
 344 [7] L. N. OLSON AND J. B. SCHRODER, *Smoothed aggregation multigrid solvers for high-order dis-*  
 345 *continuous Galerkin methods for elliptic problems*, Journal of Computational Physics, 230  
 346 (2011), pp. 6959–6976.  
 347 [8] T. WILDEY, S. MURALIKRISHNAN, AND T. BUI-THANH, *Unified geometric multigrid algorithm for*  
 348 *hybridized high-order finite element methods*, arXiv:1811.09909 [math], (2018).