An unfitted Hybrid High-Order method with cell agglomeration for elliptic interface problems

Erik Burman, Matteo Cicuttin, Guillaume Delay, Alexandre Ern

▶ To cite this version:

Erik Burman, Matteo Cicuttin, Guillaume Delay, Alexandre Ern. An unfitted Hybrid High-Order method with cell agglomeration for elliptic interface problems. SIAM Journal on Scientific Computing, 2021, 43 (2), pp.A859-A882. 10.1137/19M1285901 . hal-02280426v5

HAL Id: hal-02280426
https://hal.science/hal-02280426v5

Submitted on 22 Jan 2021

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 destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An unfitted Hybrid High-Order method with cell agglomeration for elliptic interface problems[*]

Erik Burman[†] Matteo Cicuttin[‡] Guillaume Delay[‡] Alexandre Ern[‡]

January 22, 2021

Abstract

We design and analyze a Hybrid High-Order (HHO) method on unfitted meshes to approximate elliptic interface problems by means of a consistent penalty method à la Nitsche. The curved interface can cut through the mesh cells in a rather general fashion. Robustness with respect to the cuts is achieved by using a cell agglomeration technique, and robustness with respect to the contrast in the diffusion coefficients is achieved by using a different gradient reconstruction on each side of the interface. A key novel feature of the gradient reconstruction is to incorporate a jump term across the interface, thereby releasing the Nitsche penalty parameter from the constraint of being large enough. Error estimates with optimal convergence rates are established. A robust cell agglomeration procedure limiting the agglomerations to the nearest neighbors is devised. Numerical simulations for various interface shapes corroborate the theoretical results.

1 Introduction

Generating meshes to solve problems posed on domains with a curved boundary and/or interfaces separating subdomains with different properties can be a difficult task. The use of unfitted meshes that do not fit the boundary and interfaces greatly simplifies the meshing process since such meshes can be chosen in a very simple manner. For instance one can consider meshing a rectangular or cubic domain that contains the actual physical domain. The analysis of finite element methods (FEM) on unfitted meshes was started in [3, 4]. An important advance achieved in [26] was to double the polynomial unknowns in the cells cut by the interface and to use a consistent penalty method inspired by Nitsche's method [34] to enforce weakly the jump conditions at the interface or the Dirichlet conditions at the curved boundary. We refer the reader, e.g., to [9, 11] for further advances and overviews on the topic. One difficulty with the penalty method is the presence of small cuts, that is, of mesh cells having only a small fraction of their volume on one side of the interface. Small cuts have an adverse effect on the conditioning of the method and can even hamper

[*] Draft version, January 22, 2021. The research of the first author was supported by the EPSRC grant EP/P01576X/1.

[†] Department of Mathematics, University College London, London, UK–WC1E 6BT, UK.

[‡] Université Paris-Est, CERMICS (ENPC), 77455 Marne-la-Vallée cedex 2, and INRIA, Paris, France.

convergence (see [18] for a recent discussion on the subject). There are essentially two ways to cure the issue of small cuts: one can either consider adding some stabilization such as the ghost penalty technique devised in [7] or one can agglomerate cells in the vicinity of small cuts in such a way that the newly created mesh presents no small cuts [29, 36], see also [2, 8]. Cell agglomeration is somewhat delicate in the context of conforming FEM since it requires a careful handling of hanging nodes and specific mesh structures as in [27], whereas this technique is somewhat more natural when combined with discretization methods supporting polyhedral meshes such as, e.g., discontinuous Galerkin methods.

In the present work, we devise and analyze a novel Hybrid High-Order (HHO) method to approximate elliptic interface problems on unfitted meshes. Robustness with respect to small cuts is achieved by using a cell agglomeration technique as mentioned above, whereas robustness with respect to the contrast in the diffusion coefficients is achieved by using a different gradient reconstruction on each side of the interface. HHO methods have been introduced and analyzed for diffusion and locking-free linear elasticity problems on fitted meshes in [20, 21]. As shown in [17], these methods are closely related to Hybridizable discontinuous Galerkin (HDG) methods and to nonconforming Virtual element methods (ncVEM). An unfitted HHO method for elliptic problems with a curved boundary or an interface has been proposed and analyzed in [8]. The idea therein was to extend the consistent penalty method from [26] to the HHO context and to use the polyhedral capabilities of HHO to deploy a cell agglomeration procedure to handle small cuts. To this purpose the HHO unknowns (comprising both cell and face polynomials) were doubled in the cells cut by the interface, whereas no unknowns were attached to the interface or the curved boundary. Moreover a mixed-order polynomial setting was considered with the cell unknowns being of one degree higher than the face unknowns. In addition the Nitsche-type consistency terms at the interface were evaluated using the unknowns from the less diffusive side thereby achieving robustness with respect to contrasted coefficients in the spirit of [10, 23]. In the present work, we achieve three important advances with respect to [8]. First the gradient reconstruction operators in the cut cells are modified by adding a jump term on the less diffusive side of the interface. This offers the crucial advantage of releasing the penalty parameter from any constraint of being large enough with respect to the size of the constant from a discrete trace inequality. As a consequence, contrary to the classical Nitsche's method where this coefficient has to be tuned to make the scheme stable, we can set this parameter to any simple value (for instance, we can set it to one). Second we revise and extend the cell agglomeration procedure to ensure that agglomerated cells do not propagate further than nearest neighbors. We observe that the present cell agglomeration procedure is of (much) broader interest than HHO methods and even unfitted methods. Finally we present for the first time numerical simulations on unfitted HHO methods illustrating the cell agglomeration procedure, the optimality and robustness of the error estimates, and the conditioning of the system matrix.

Let us put our work in perspective with the literature. Recalling that HHO and HDG methods are closely related, we first mention that eXtended HDG (X-HDG) methods have been proposed in [24, 25] where an additional trace variable is introduced at the interface or the curved boundary. In [16, 35], HDG methods without an additional trace variable are devised and analyzed, whereby a transferring technique is used to approximate the data on the curved boundary or the jump conditions across the interface. In contrast these conditions are enforced herein at the boundary or interface by means of a consistent penalty method. Concerning more specifically HHO methods, the gradient reconstruction in the cut cells differs from [8] since the present reconstruction incorporates a jump term. As mentioned above, this offers the crucial advantage of releasing the penalty parameter from any constraint of being large enough. Incidentally we point out that this idea has common features with the parameter-free Nitsche's method devised recently in [31] for unfitted FEM. These links are further discussed below. Another extension with respect to [8] is to allow for reconstructions having full polynomial order without being necessarily curl-free, as motivated by the future treatment of nonlinear problems (not considered herein yet, see, e.g., [6, 1] for fitted HHO methods in nonlinear mechanics). Finally we mention that the present unfitted method can be used as well on geometries with a curved boundary that is then immersed within the mesh. A fitted HHO method for diffusion problems with curved boundaries is assessed numerically in [5].

This work is organized as follows. We present the model elliptic interface problem and the unfitted HHO method in Section 2. The numerical analysis leading to optimal and robust error estimates is presented in Section 3. Implementation aspects, and in particular the cell agglomeration procedure, are covered in Section 4. Finally numerical simulations which confirm the theoretical convergence rates and illustrate the robustness of the method with respect to small cuts and contrasted coefficients are discussed in Section 5.

2 Model problem and unfitted HHO method

In this section we present the elliptic interface problem and its discretization by the unfitted HHO method.

2.1 Model problem

Let Ω be a polyhedral domain in $\mathbb{R}^d$, $d \in \{2,3\}$ (open, bounded, connected, Lipschitz subset of $\mathbb{R}^d$) and consider a partition of Ω into two disjoint subdomains $\overline{\Omega} = \overline{\Omega_1} \cup \overline{\Omega_2}$ with the interface $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$. As in [8] the interface Γ is assumed to be a smooth $(d-1)$-dimensional manifold of class C^2 that is not self-intersecting. This assumption can be relaxed at the price of additional technical points that are not further explored herein. The unit normal vector $\boldsymbol{n}_\Gamma$ to Γ conventionally points from Ω_1 to Ω_2. For a smooth enough function v defined on $\Omega_1 \cup \Omega_2$, we denote its jump across Γ as $[\![v]\!]_\Gamma := v_{|\Omega_1} - v_{|\Omega_2}$. Our goal is to approximate the solution $u \in H^1(\Omega_1 \cup \Omega_2) = \{v \in L^2(\Omega) \mid v_{|\Omega_i} \in H^1(\Omega_i),\ i \in \{1,2\}\}$ of the following elliptic interface problem:

$$-\nabla\cdot(\kappa\nabla u) = f \qquad \text{in } \Omega_1 \cup \Omega_2, \tag{1a}$$

$$[\![u]\!]_\Gamma = g_D \qquad \text{on } \Gamma, \tag{1b}$$

$$[\![\kappa\nabla u]\!]_\Gamma\cdot\boldsymbol{n}_\Gamma = g_N \qquad \text{on } \Gamma, \tag{1c}$$

$$u = 0 \qquad \text{on } \partial\Omega, \tag{1d}$$

with $f \in L^2(\Omega)$, $g_D \in H^{\frac{1}{2}}(\Gamma)$, and $g_N \in L^2(\Gamma)$. For simplicity we consider a homogeneous Dirichlet boundary condition on $\partial\Omega$. To avoid technicalities we assume that the diffusion coefficient κ is scalar-valued and that $\kappa_i := \kappa_{|\Omega_i}$ is constant for each $i \in \{1,2\}$. To fix the ideas we assume that $\kappa_1 \leq \kappa_2$. Our analysis covers the strongly contrasted case where $\kappa_1 \ll \kappa_2$.

2.2 Unfitted meshes

Let $(\mathcal{T}_h)_{h>0}$ be a family of meshes of Ω. The meshes can have cells that are polyhedra in $\mathbb{R}^d$ with planar faces, and hanging nodes are also possible. The mesh cells are taken to be open sets in $\mathbb{R}^d$. For all $T \in \mathcal{T}_h$, h_T denotes the diameter of the cell T and $\boldsymbol{n}_T$ the unit normal on ∂T pointing outward T. We set conventionally $h := \max_{T\in\mathcal{T}_h} h_T$. The mesh faces are collected in the set $\mathcal{F}_h$. Assumptions on the mesh regularity and how the interface cuts the mesh cells are stated in Section 3.1.

Let us define the partition $\mathcal{T}_h = \mathcal{T}_h^1 \cup \mathcal{T}_h^\Gamma \cup \mathcal{T}_h^2$, where the subsets

$$\mathcal{T}_h^i := \{T \in \mathcal{T}_h \mid T \subset \Omega_i\} \qquad \forall i \in \{1,2\}, \tag{2a}$$

$$\mathcal{T}_h^\Gamma := \{T \in \mathcal{T}_h \mid T \cap \Gamma \neq \emptyset\}, \tag{2b}$$

collect respectively the mesh cells inside the subdomain Ω_i, $i \in \{1,2\}$ (the uncut cells), and those cut by the interface Γ (the cut cells). For every cut cell $T \in \mathcal{T}_h^\Gamma$ and all $i \in \{1,2\}$, we define

$$T^i := T \cap \Omega_i, \qquad T^\Gamma := T \cap \Gamma. \tag{3}$$

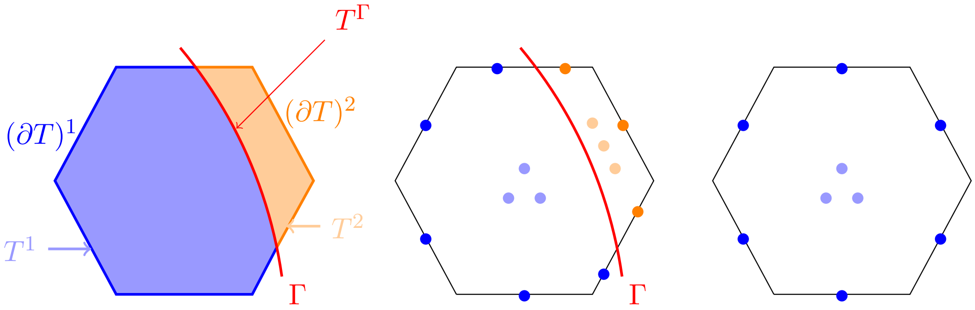
For all $i \in \{1,2\}$, the boundary $\partial(T^i)$ of the subcell T^i is decomposed as

$$\partial(T^i) = (\partial T)^i \cup T^\Gamma, \qquad (\partial T)^i := \partial T \cap (\overline{\Omega_i} \setminus \Gamma). \tag{4}$$

An illustration is presented in the left panel of Fig. 1. To unify the notation, for every uncut cell $T \in \mathcal{T}_h^i$, $i \in \{1,2\}$, we set

$$T^i := T, \quad T^{\bar\imath} := \emptyset, \quad (\partial T)^i := \partial T, \quad (\partial T)^{\bar\imath} := \emptyset, \quad T^\Gamma := \emptyset, \tag{5}$$

where $\bar\imath = 3 - i$ (so that $\bar 1 = 2$ and $\bar 2 = 1$).



The cell decomposition. A cut cell. An uncut cell.

Figure 1: Basic notation for a cut cell $T \in \mathcal{T}_h^\Gamma$ (left). Local degrees of freedom ($k = 0$) for a cut (center) and an uncut (right) cells.

2.3 The local discrete problem

We now describe the local unknowns and operators needed to formulate the unfitted HHO method. For a subset $S \subset \mathbb{R}^d$ consisting of one mesh (sub)cell or one mesh (sub)face, we denote $\mathbb{P}^\ell(S)$ (resp. $\mathbb{P}^\ell(S;\mathbb{R}^d)$) the space composed of the restriction to S of scalar-valued (resp. vector-valued) polynomials of total degree at most $\ell \geq 0$. Moreover, for a subset $S \subset \mathbb{R}^d$, we denote $(\cdot,\cdot)_S$ the $L^2(S)$-inner product with appropriate Lebesgue measure and $\|\cdot\|_S$ the induced norm. Whenever $S = \emptyset$, we abuse the notation by writing $\mathbb{P}^\ell(S) := \{0\}$ and $(\cdot,\cdot)_S := 0$.

Let $k \geq 0$ be the polynomial degree of the method. For every uncut cell $T \in \mathcal{T}_h^i$, $i \in \{1,2\}$, the local discrete HHO unknowns are a pair of functions: one polynomial of degree at most $(k+1)$ attached to the cell T and one polynomial of degree at most k attached to each face $F \in \mathcal{F}_{\partial T}$, where $\mathcal{F}_{\partial T}$ comprises all the faces composing the boundary ∂T of the cell T. We use the notation $\hat{v}_T := (v_T, v_{\partial T}) \in \mathbb{P}^{k+1}(T) \times \mathbb{P}^k(\mathcal{F}_{\partial T})$ with $\mathbb{P}^k(\mathcal{F}_{\partial T}) := \prod_{F\in\mathcal{F}_{\partial T}} \mathbb{P}^k(F)$. In the original HHO method from [21, 20], equal-order unknowns are considered (i.e. polynomials of degree at most k on faces and cells). In unfitted HHO methods it is also possible to consider equal-order unknowns in the uncut cells, but mixed-order unknowns are needed in the cut cells (i.e. polynomials of degree at most k on faces and $k+1$ in cells). For simplicity, we consider mixed-order unknowns everywhere.

For every cut cell $T \in \mathcal{T}_h^\Gamma$, we double the HHO unknowns as in [8] so as to have the usual HHO unknowns available on each subcell, up to the interface Γ where there are no unknowns. Thus the local HHO unknowns in every cut cell $T \in \mathcal{T}_h^\Gamma$ are

$$\hat{v}_T := (\hat{v}_{T^1}, \hat{v}_{T^2}) := (v_{T^1}, v_{(\partial T)^1}, v_{T^2}, v_{(\partial T)^2}) \in \hat{U}_T^k := \hat{U}_{T^1}^k \times \hat{U}_{T^2}^k, \tag{6}$$

with $\hat{U}_{T^i}^k := \mathbb{P}^{k+1}(T^i) \times \mathbb{P}^k(\mathcal{F}_{(\partial T)^i})$, $\mathcal{F}_{(\partial T)^i} := \{F^i := F \cap \Omega_i \mid F \in \mathcal{F}_{\partial T}\}$ is the collection of the (sub)faces composing $(\partial T)^i$, and $\mathbb{P}^k(\mathcal{F}_{(\partial T)^i}) := \prod_{F^i\in\mathcal{F}_{(\partial T)^i}} \mathbb{P}^k(F^i)$, $i \in \{1,2\}$. The HHO unknowns for cut and uncut cells are shown in the central and right panels of Fig. 1. For every geometric entity, the number of points represents the number of degrees of freedom attached to it. To unify the notation between cut and uncut cells, we define $\hat{U}_T^k$ as in (6) for all $T \in \mathcal{T}_h^i$, $i \in \{1,2\}$. With this abuse of notation we have $\hat{v}_T := (v_T, v_{\partial T}, 0, 0)$ for all $T \in \mathcal{T}_h^1$ and $\hat{v}_T := (0, 0, v_T, v_{\partial T})$ for all $T \in \mathcal{T}_h^2$.

Let $T \in \mathcal{T}_h$. As usual in HHO methods, the two local ingredients are a reconstruction operator and a stabilization operator. In every cut cell $T \in \mathcal{T}_h^\Gamma$, there are two gradient reconstruction operators $\boldsymbol{G}_{T^i}^k : \hat{U}_T^k \to \mathbb{G}^k(T^i;\mathbb{R}^d)$, $i \in \{1,2\}$, where $\mathbb{G}^k(T^i;\mathbb{R}^d)$ is composed of the restriction to T^i of $\mathbb{R}^d$-valued polynomial functions with the requirement that $(\boldsymbol{q}\cdot\boldsymbol{n}_T)_{|(\partial T)^i} \in \mathbb{P}^k(\mathcal{F}_{(\partial T)^i})$ for all $\boldsymbol{q} \in \mathbb{G}^k(T^i;\mathbb{R}^d)$. Possible choices are $\mathbb{G}^k(T^i;\mathbb{R}^d) := \mathbb{P}^k(T^i;\mathbb{R}^d)$ and $\mathbb{G}^k(T^i;\mathbb{R}^d) := \nabla\mathbb{P}^{k+1}(T^i)$. The former choice leads to a larger reconstruction space and is more suitable in the case of nonlinear problems [6, 1]. The latter choice is the one made in the original HHO methods [21, 20] and is also made in [8] for unfitted HHO methods. For every cut cell $T \in \mathcal{T}_h^\Gamma$ and every $\hat{v}_T \in \hat{U}_T^k$, letting $[\![v_T]\!]_\Gamma := v_{T^1} - v_{T^2}$, we set

$$(\boldsymbol{G}_{T^1}^k(\hat{v}_T), \boldsymbol{q})_{T^1} := (\nabla v_{T^1}, \boldsymbol{q})_{T^1} + (v_{(\partial T)^1} - v_{T^1}, \boldsymbol{q}\cdot\boldsymbol{n}_T)_{(\partial T)^1} - ([\![v_T]\!]_\Gamma, \boldsymbol{q}\cdot\boldsymbol{n}_\Gamma)_{T^\Gamma}, \tag{7}$$

$$(\boldsymbol{G}_{T^2}^k(\hat{v}_T), \boldsymbol{q})_{T^2} := (\nabla v_{T^2}, \boldsymbol{q})_{T^2} + (v_{(\partial T)^2} - v_{T^2}, \boldsymbol{q}\cdot\boldsymbol{n}_T)_{(\partial T)^2}, \tag{8}$$

for all $\boldsymbol{q} \in \mathbb{G}^k(T^1;\mathbb{R}^d)$ in (7) and all $\boldsymbol{q} \in \mathbb{G}^k(T^2;\mathbb{R}^d)$ in (8). Note that $\boldsymbol{G}^k_{T^2}(\hat{v}_T)$ only depends on $\hat{v}_{T^2}$, whereas $\boldsymbol{G}^k_{T^1}(\hat{v}_T)$ depends on both $\hat{v}_{T^1}$ and $\hat{v}_{T^2}$ owing to the jump term on the right-hand side of (7). The difference in the reconstruction between the two subdomains is important in the highly-contrasted case where $\kappa_1 \ll \kappa_2$ (recall our convention that $\kappa_1 \leq \kappa_2$). Using the above conventions on the notation, we can consider the same definitions for every uncut cell $T \in \mathcal{T}^i_h$, $i \in \{1,2\}$, leading to $(\boldsymbol{G}^k_{T^i}(\hat{v}_T), \boldsymbol{q})_{T^i} := (\nabla v_{T^i}, \boldsymbol{q})_{T^i} + (v_{(\partial T)^i} - v_{T^i}, \boldsymbol{q}{\cdot}\boldsymbol{n}_T)_{(\partial T)^i}$ for all $\boldsymbol{q} \in \mathbb{G}^k(T^i;\mathbb{R}^d)$, and $\boldsymbol{G}^k_{T^{\bar{\imath}}}(\hat{v}_T) = \boldsymbol{0}$. Recalling that $T^i := T$ and $(\partial T)^i := \partial T$, $\boldsymbol{G}^k_{T^i}(\hat{v}_T)$ corresponds to the usual HHO gradient reconstruction in the uncut cells. Furthermore, to weakly enforce the matching between cell- and face-based HHO unknowns, we consider for all $T \in \mathcal{T}_h$ the stabilization bilinear forms $\hat{s}_{T^i}$, $i \in \{1,2\}$, such that for all $\hat{v}_{T^i}, \hat{w}_{T^i} \in \hat{U}^k_{T^i}$,

$$\hat{s}_{T^i}(\hat{v}_{T^i}, \hat{w}_{T^i}) := \kappa_i h_T^{-1}\Big(\Pi^k_{(\partial T)^i}(v_{T^i} - v_{(\partial T)^i}), w_{T^i} - w_{(\partial T)^i}\Big)_{(\partial T)^i}, \tag{9}$$

where $\Pi^k_{(\partial T)^i}$ denotes the L^2-orthogonal projector onto $\mathbb{P}^k(\mathcal{F}_{(\partial T)^i})$. Owing to the above conventions, in every uncut cell $T \in \mathcal{T}^i_h$, $\hat{s}_{T^i}(\hat{v}_{T^i}, \hat{w}_{T^i})$ corresponds to the usual HHO stabilization with mixed-order unknowns (similar to the HDG stabilization from Lehrenfeld and Schöberl [30, 33]), whereas $\hat{s}_{T^{\bar{\imath}}}(\hat{v}_{T^{\bar{\imath}}}, \hat{w}_{T^{\bar{\imath}}}) = 0$.

The discrete HHO bilinear and linear forms read for all $\hat{v}_T, \hat{w}_T \in \hat{U}^k_T$,

$$\hat{a}_T(\hat{v}_T, \hat{w}_T) := \sum_{i\in\{1,2\}} \Big\{\kappa_i(\boldsymbol{G}^k_{T^i}(\hat{v}_T), \boldsymbol{G}^k_{T^i}(\hat{w}_T))_{T^i} + \hat{s}_{T^i}(\hat{v}_{T^i}, \hat{w}_{T^i})\Big\} + \hat{s}^\Gamma_T(\hat{v}_T, \hat{w}_T), \tag{10}$$

$$\hat{\ell}_T(\hat{w}_T) := \sum_{i\in\{1,2\}} (f, w_{T^i})_{T^i} + \kappa_1(g_D, \phi_T(\hat{w}_T))_{T^\Gamma} + (g_N, w_{T^2})_{T^\Gamma}, \tag{11}$$

with

$$\hat{s}^\Gamma_T(\hat{v}_T, \hat{w}_T) := \kappa_1 h_T^{-1}([\![v_T]\!]_\Gamma, [\![w_T]\!]_\Gamma)_{T^\Gamma},\ \phi_T(\hat{w}_T) := h_T^{-1}[\![w_T]\!]_\Gamma - \boldsymbol{G}^k_{T^1}(\hat{w}_T){\cdot}\boldsymbol{n}_\Gamma. \tag{12}$$

Note that for every cut cell $T \in \mathcal{T}^\Gamma_h$, the stabilization bilinear forms $\hat{s}_{T^i}$, $i \in \{1,2\}$, do not couple the HHO unknowns from both sides of the interface, whereas the gradient reconstruction in T^1 couples them. The coupling also occurs by means of the penalty bilinear form $\hat{s}^\Gamma_T$ which enforces weakly the jump conditions across the interface. The quantity $\phi_T(\hat{w}_T)$ defined in (12) is designed so that the formulation has optimal consistency properties.

Remark 2.1. *(Gradient reconstruction)* The gradient reconstruction operators defined in (7)-(8) are one of the main novelties of the present work. They differ from those considered in [8] where the two gradient reconstructions in a cut cell need to be evaluated simultaneously. The main advantage of the present reconstructions is that the incorporation of the jump term in (7) allows one to avoid the need for a large enough penalty parameter scaling the bilinear form $\hat{s}^\Gamma_T$. This is why we have taken this parameter equal to one in (12).

2.4 The global discrete problem

We define the global discrete space

$$\hat{U}_h^k := \prod_{T\in\mathcal{T}_h} \mathbb{P}^{k+1}(T^1) \times \prod_{F\in\mathcal{F}_h} \mathbb{P}^k(F^1) \times \prod_{T\in\mathcal{T}_h} \mathbb{P}^{k+1}(T^2) \times \prod_{F\in\mathcal{F}_h} \mathbb{P}^k(F^2), \tag{13}$$

recalling that $T^i := T \cap \Omega^i$ and $F^i := F \cap \Omega^i$, $i \in \{1,2\}$. For all $\hat{v}_h \in \hat{U}_h^k$ and all T in $\mathcal{T}_h$, we denote $\hat{v}_T := (v_{T^1}, v_{(\partial T)^1}, v_{T^2}, v_{(\partial T)^2})$ the components of $\hat{v}_h$ respectively attached to T^1, $(\partial T)^1$, T^2, and $(\partial T)^2$. Recalling the above conventions on the uncut cells, we have $\hat{v}_T := (v_T, v_{\partial T}, 0, 0)$ for all $T \in \mathcal{T}_h^1$ and $\hat{v}_T := (0, 0, v_T, v_{\partial T})$ for all $T \in \mathcal{T}_h^2$. We denote $\hat{U}_{h0}^k$ the subspace of $\hat{U}_h^k$ where all degrees of freedom attached to the faces composing $\partial\Omega$ are null. The global discrete problem reads as follows: Find $\hat{u}_h \in \hat{U}_{h0}^k$ such that

$$\hat{a}_h(\hat{u}_h, \hat{w}_h) = \hat{\ell}_h(\hat{w}_h), \qquad \forall \hat{w}_h \in \hat{U}_{h0}^k, \tag{14}$$

where the bilinear and linear forms are defined by summing all the local contributions, i.e., for all $\hat{v}_h, \hat{w}_h \in \hat{U}_{h0}^k$, we set $\hat{a}_h(\hat{v}_h, \hat{w}_h) := \sum_{T\in\mathcal{T}_h} \hat{a}_T(\hat{v}_T, \hat{w}_T)$ and $\hat{\ell}_h(\hat{w}_h) := \sum_{T\in\mathcal{T}_h} \hat{\ell}_T(\hat{w}_T)$.

The discrete problem (14) can be solved efficiently by eliminating locally all the cell-based unknowns by means of a static condensation procedure. This local elimination leads to a global transmission problem on the mesh skeleton involving only the face-based unknowns. The resulting stencil couples unknowns attached to neighboring faces (in the sense of cells). Once this global transmission problem is solved, one can recover the values of the cell-based unknowns in every cell by local solves. We refer the reader, e.g., to [15].

2.5 Variants

Let us give two variants of the scheme based on the use of other gradient reconstructions. The contents of this section are not used in the sequel and can be skipped at first reading.

A first variant is to define the gradient reconstructions in the cut cells in such a way that the two subdomains play symmetric roles. In this case one replaces (7)-(8) with the single definition

$$(\widetilde{\boldsymbol{G}}_{T^i}^k(\hat{v}_T), \boldsymbol{q})_{T^i} := (\nabla v_{T^i}, \boldsymbol{q})_{T^i} + (v_{(\partial T)^i} - v_{T^i}, \boldsymbol{q}\cdot\boldsymbol{n}_T)_{(\partial T)^i} - \frac{1}{2}([\![v_T]\!]_\Gamma, \boldsymbol{q}\cdot\boldsymbol{n}_\Gamma)_{T^\Gamma}, \tag{15}$$

for all $\hat{v}_T \in \hat{U}_T^k$ and all $\boldsymbol{q} \in \mathbb{G}^k(T^i;\mathbb{R}^d)$, $i \in \{1,2\}$, and the bilinear and linear forms become

$$\begin{aligned}
\widetilde{a}_T(\hat{v}_T, \hat{w}_T) &:= \sum_{i\in\{1,2\}} \Big\{ \kappa_i(\widetilde{\boldsymbol{G}}_{T^i}^k(\hat{v}_T), \widetilde{\boldsymbol{G}}_{T^i}^k(\hat{w}_T))_{T^i} + \hat{s}_{T^i}(\hat{v}_{T^i}, \hat{w}_{T^i}) \Big\} + \widetilde{s}_T^\Gamma(\hat{v}_T, \hat{w}_T), \\
\widetilde{\ell}_T(\hat{w}_T) &:= \sum_{i\in\{1,2\}} (f, w_{T^i})_{T^i} + (g_D, \widetilde{\phi}_T(\hat{w}_T))_{T^\Gamma} + (g_N, \{w_T\}_\Gamma)_{T^\Gamma},
\end{aligned}$$

with the penalty bilinear form $\widetilde{s}_T^\Gamma(\hat{v}_T, \hat{w}_T) := \kappa_2 h_T^{-1}([\![v_T]\!]_\Gamma, [\![w_T]\!]_\Gamma)_{T^\Gamma}$ (now scaled with κ_2, recalling that $\kappa_1 \le \kappa_2$), $\widetilde{\phi}_T(\hat{w}_T) := \kappa_2 h_T^{-1}[\![w_T]\!]_\Gamma - \{\kappa\widetilde{\boldsymbol{G}}_T^k(\hat{w}_T)\}_\Gamma\cdot\boldsymbol{n}_\Gamma$, and the average values $\{w_T\}_\Gamma := \frac{1}{2}(w_{T^1} + w_{T^2})$, $\{\kappa\widetilde{\boldsymbol{G}}_T^k(\hat{w}_T)\}_\Gamma := \frac{1}{2}(\kappa_1\widetilde{\boldsymbol{G}}_{T^1}^k(\hat{w}_T) + \kappa_2\widetilde{\boldsymbol{G}}_{T^2}^k(\hat{w}_T))$ at the interface. The error analysis (which proceeds as in Section 3 but is omitted for brevity) leads to error estimates with optimal convergence rates but lacking robustness in the highly-contrasted case where $\kappa_1 \ll \kappa_2$.

A second variant, which is closer in spirit to the Nitsche-type approach and the unfitted HHO method previously devised in [8], is to set

$$(\check{\boldsymbol{G}}^k_{T^i}(\hat{v}_T), \boldsymbol{q})_{T^i} := (\nabla v_{T^i}, \boldsymbol{q})_{T^i} + (v_{(\partial T)^i} - v_{T^i}, \boldsymbol{q}{\cdot}\boldsymbol{n}_T)_{(\partial T)^i},$$

for all $\hat{v}_T \in \hat{U}^k_T$, $\boldsymbol{q} \in \mathbb{G}^k(T^i; \mathbb{R}^d)$, $i \in \{1,2\}$. Owing to the absence of the jump term in the gradient reconstruction, the consistency error analysis (which proceeds as in Section 3 but is omitted for brevity) shows that the bilinear and linear forms should now be written as

$$\begin{aligned}
\check{a}_T(\hat{v}_T, \hat{w}_T) &:= \sum_{i\in\{1,2\}} \Big\{ \kappa_i(\check{\boldsymbol{G}}^k_{T^i}(\hat{v}_T), \check{\boldsymbol{G}}^k_{T^i}(\hat{w}_T))_{T^i} + \hat{s}_{T^i}(\hat{v}_{T^i}, \hat{w}_{T^i}) \Big\} \\
&\quad - \hat{n}^\Gamma_T(\hat{v}_T, \hat{w}_T) + \eta \hat{s}^\Gamma_T(\hat{v}_T, \hat{w}_T), \\
\check{\ell}_T(\hat{w}_T) &:= \sum_{i\in\{1,2\}} (f, w_{T^i})_{T^i} + \kappa_1(g_D, \phi_{\eta,T}(\hat{w}_T))_{T^\Gamma} + (g_N, w_{T^2})_{T^\Gamma},
\end{aligned}$$

where $\eta > 0$ is a user-dependent parameter, $\hat{s}^\Gamma_T$ is still defined by (12),

$$\hat{n}^\Gamma_T(\hat{v}_T, \hat{w}_T) := \kappa_1([\![v_T]\!]_\Gamma, \nabla w_{T^1}{\cdot}\boldsymbol{n}_\Gamma)_{T^\Gamma} + \kappa_1(\nabla v_{T^1}{\cdot}\boldsymbol{n}_\Gamma, [\![w_T]\!]_\Gamma)_{T^\Gamma},$$

and $\phi_{\eta,T}(\hat{w}_T) := \eta h_T^{-1}[\![w_T]\!]_\Gamma - \nabla w_{T^1}{\cdot}\boldsymbol{n}_\Gamma$. As is classical in this situation, the need to bound the Nitsche-type bilinear form $\hat{n}^\Gamma_T$ to ensure coercivity makes it necessary to take a value for η that is large enough, depending on the constant from a discrete trace inequality. Note that the handling of the diffusion coefficients on both sides of the interface is inspired from [10, 23] and leads to robust error estimates in the highly contrasted case. However, avoiding a large enough penalty parameter is highly beneficial in practice and is the key motivation for introducing the novel reconstructions (7)-(8). Similar ideas in the context of unfitted FEM were discussed in [31].

3 Analysis

In this section we establish stability and error estimates for the unfitted HHO method introduced in the previous section.

3.1 Admissible meshes

The unfitted HHO method is to be deployed on shape-regular polyhedral mesh sequences, i.e. for all $h > 0$, $\mathcal{T}_h$ admits a matching simplicial submesh $\mathcal{T}'_h$ such that any cell (or face) of $\mathcal{T}'_h$ is a subset of a cell (or face) of $\mathcal{T}_h$. Moreover, there exists a mesh-regularity parameter $\rho > 0$ such that for all $h > 0$, all $T \in \mathcal{T}_h$, and all $S \in \mathcal{T}'_h$ such that $S \subset T$, we have $\rho h_S \leq r_S$ and $\rho h_T \leq h_S$, where r_S denotes the inradius of the simplex S. In the numerical experiments presented in Section 5, we start with structured meshes, which then become (locally) polyhedral after the cell agglomeration procedure. Three additional assumptions on the mesh are needed. The first one quantifies how well the interface cuts the mesh cells, the second one quantifies how well the mesh resolves the interface and the third one requires the mesh not to be excessively graded. In the sequel $B(\boldsymbol{x}, r)$ denotes the ball of center $\boldsymbol{x}$ and radius r.

Assumption 1 (Cut cells). *There is $\delta \in (0,1)$ such that, for all $T \in \mathcal{T}_h^\Gamma$ and all $i \in \{1,2\}$, there is $\tilde{\boldsymbol{x}}_{T^i} \in T^i$ such that $B(\tilde{\boldsymbol{x}}_{T^i}, \delta h_T) \subset T^i$.*

Assumption 2 (Resolving Γ). *There is $\gamma \in (0,1)$ such that, for all $T \in \mathcal{T}_h^\Gamma$, there is a point $\check{\boldsymbol{x}}_T \in \mathbb{R}^d$ such that setting $T^\dagger := B(\check{\boldsymbol{x}}_T, \gamma^{-1} h_T)$ we have the following properties:* (i) *$T \subset T^\dagger$;* (ii) *for all $\boldsymbol{s} \in T^\Gamma$, $d(\check{\boldsymbol{x}}_T, T_{\boldsymbol{s}}\Gamma) \geq \gamma h_T$, where $T_{\boldsymbol{s}}\Gamma$ is the tangent plane to Γ at the point $\boldsymbol{s}$;* (iii) *For all $F \in \mathcal{F}_{\partial T}$, there is $\boldsymbol{x}_F \in T^\dagger$ such that $d(\boldsymbol{x}_F, F) \geq \gamma h_T$.*

For all $T \in \mathcal{T}_h$, let the neighboring layers $\Delta_j(T) \subset \mathbb{R}^d$ be defined by induction as $\Delta_0(T) := T$ and $\Delta_{j+1}(T) := \{T' \in \mathcal{T}_h \mid \overline{T'} \cap \overline{\Delta_j(T)} \neq \emptyset\}$ for all $j \in \mathbb{N}$.

Assumption 3 (Mild mesh grading). *There is $n_0 \in \mathbb{N}$ such that for all $T \in \mathcal{T}_h$, the ball $T^\dagger$ introduced in Assumption 2 satisfies $T^\dagger \subset \Delta_{n_0}(T)$.*

Let us now briefly comment on the foundations and consequences of Assumptions 1 and 2. It is shown in [8, Lem. 6.4] that if the mesh is fine enough, it is possible to devise a cell agglomeration procedure so that, choosing the parameter δ small enough (depending on the regularity parameter ρ), Assumption 1 is fulfilled. In the present work we improve on the procedure outlined in [8] by adding a third step that guarantees that there is no propagation of the cell agglomeration. More details are given in Section 4.3. The role of Assumption 1 in the analysis is to provide the following discrete (inverse) inequalities.

Lemma 4 (Discrete (inverse) inequalities). *Let Assumption 1 be fulfilled. Let $\ell \in \mathbb{N}$. There is $c_{\rm disc}$, depending on ρ, δ, and ℓ, such that, for all $T \in \mathcal{T}_h$, all $i \in \{1,2\}$ and all $v_{T^i} \in \mathbb{P}^\ell(T^i)$, the following inequalities hold true:*

- *(Discrete trace inequality) $\|v_{T^i}\|_{(\partial T)^i} + \|v_{T^i}\|_{T^\Gamma} \leq c_{\rm disc} h_T^{-\frac{1}{2}} \|v_{T^i}\|_{T^i}$.*
- *(Discrete inverse inequality) $\|\nabla v_{T^i}\|_{T^i} \leq c_{\rm disc} h_T^{-1} \|v_{T^i}\|_{T^i}$.*
- *(Discrete Poincaré inequality) Assuming that $(v_{T^i}, 1)_{B(\tilde{\boldsymbol{x}}_{T^i}, h_T)} = 0$, we have $\|v_{T^i}\|_{T^i} \leq c_{\rm disc} h_T \|\nabla v_{T^i}\|_{T^i}$.*

Proof. The discrete trace inequality is shown in [8, Lemma 3.4]. The proof of the other two inequalities uses similar arguments. For brevity we only sketch the proof of the inverse inequality. Since this inequality is classical in uncut cells (see, e.g., [19, Lemma 1.44]), we consider a cut cell $T \in \mathcal{T}_h^\Gamma$. Invoking an inverse inequality in $B(\tilde{\boldsymbol{x}}_{T^i}, h_T)$ (with constant c_0) followed by the inequality $\|v_{T^i}\|_{B(\tilde{\boldsymbol{x}}_{T^i}, h_T)} \leq c_1 \|v_{T^i}\|_{B(\tilde{\boldsymbol{x}}_{T^i}, \delta h_T)}$ leads to $\|\nabla v_{T^i}\|_{T^i} \leq \|\nabla v_{T^i}\|_{B(\tilde{\boldsymbol{x}}_{T^i}, h_T)} \leq c_0 h_T^{-1} \|v_{T^i}\|_{B(\tilde{\boldsymbol{x}}_{T^i}, h_T)} \leq c_0 c_1 h_T^{-1} \|v_{T^i}\|_{B(\tilde{\boldsymbol{x}}_{T^i}, \delta h_T)} \leq c_0 c_1 h_T^{-1} \|v_{T^i}\|_{T^i}$. □

It is shown in [8, Lemma 6.1] that if the mesh is fine enough with respect to the curvature of the interface, the points (i) and (ii) of Assumption 2 hold true. Moreover the point (iii) of this assumption can be established by invoking the shape-regularity of the mesh as shown in the proof of [8, Lemma 3.3]. The role of Assumption 2 is to provide a multiplicative trace inequality that is needed to establish optimal approximation properties on the faces and on the interface within the cut cells. We only state this inequality and refer the reader to [8, Lemma 3.3] for the proof. Notice that the argument of the proof classically invokes a cone condition and that the inequality is stated on the ball $T^\dagger$ since this is the geometric object in which approximation properties are derived.

Lemma 5 (Multiplicative trace inequality)**.** *Let Assumption* 2 *be fulfilled. There is* $c_{\mathrm{mtr}} > 0$*, depending on* ρ *and* γ*, such that for all* $T \in \mathcal{T}_h$*, all* $v \in H^1(T^\dagger)$*, and all* $i \in \{1, 2\}$*,*

$$\|v\|_{(\partial T)^i \cup T^\Gamma} \leq c_{\mathrm{mtr}} \left(h_T^{-\frac{1}{2}} \|v\|_{T^\dagger} + \|v\|_{T^\dagger}^{\frac{1}{2}} \|\nabla v\|_{T^\dagger}^{\frac{1}{2}} \right). \tag{16}$$

Remark 3.1. *(Constant* c_{disc}*)* The constant c_{disc} from Lemma 4 can depend, as stated, on ρ, δ, and ℓ. We leave to future work the study of the dependence of c_{disc} on δ and ℓ. Concerning δ, we merely point out at this stage that the cell agglomeration procedure allows that δ remains comparable to ρ. An illustration of the moderate impact of δ on the condition number is presented in Section 5.3. Concerning ℓ, we notice that we are using here moderate polynomial degrees (up to 3 in our experiments), and that another important issue to be addressed with the use of high-order methods is a finer control of quadrature errors on the cut cells, as further discussed in Section 5. Interestingly, we point out that under suitable (mild) assumptions on the cut cell geometry, inverse and trace inequalities with optimal dependence on the polynomial degree can be proven; see [13, 14].

In what follows we use the convention $A \lesssim B$ to abbreviate the inequality $A \leq CB$ for positive real numbers A and B, where the constant C only depends on the polynomial degree $k \geq 0$, the mesh parameters ρ, δ, γ and n_0, and the above constants c_{disc} and c_{mtr}.

3.2 Stability and well-posedness

For all $T \in \mathcal{T}_h$ and all $\hat{v}_T \in \hat{U}_T^k$, we define the local semi-norm

$$|\hat{v}_T|^2_{\hat{a}_T} := \sum_{i \in \{1,2\}} \kappa_i \Big\{ \|\nabla v_{T^i}\|^2_{T^i} + h_T^{-1} \|v_{(\partial T)^i} - v_{T^i}\|^2_{(\partial T)^i} \Big\} + \kappa_1 h_T^{-1} \|[\![v_T]\!]_\Gamma\|^2_{T^\Gamma}. \tag{17}$$

Lemma 6 (Stability)**.** *Let Assumption* 1 *be fulfilled. We have* $|\hat{v}_T|^2_{\hat{a}_T} \lesssim \hat{a}_T(\hat{v}_T, \hat{v}_T)$ *for all* $T \in \mathcal{T}_h$ *and all* $\hat{v}_T \in \hat{U}_T^k$*.*

Proof. Owing to (7), we infer that

$$\begin{aligned} \|\nabla v_{T^1}\|^2_{T^1} &= (\nabla v_{T^1}, \nabla v_{T^1})_{T^1} \\ &= (\boldsymbol{G}^k_{T^1}(\hat{v}_T), \nabla v_{T^1})_{T^1} - (v_{(\partial T)^1} - v_{T^1}, \nabla v_{T^1} \cdot \boldsymbol{n}_T)_{(\partial T)^1} + ([\![v_T]\!]_\Gamma, \nabla v_{T^1} \boldsymbol{n}_\Gamma)_{T^\Gamma}. \end{aligned}$$

The Cauchy–Schwarz inequality, $\nabla v_{T^1} \cdot \boldsymbol{n}_T \in \mathbb{P}^k(\mathcal{F}_{(\partial T)^1})$, and the discrete trace inequality from Lemma 4 to bound $\|\nabla v_{T^1}\|_{(\partial T)^1 \cup T^\Gamma}$ lead to

$$\|\nabla v_{T^1}\|^2_{T^1} \lesssim \|\boldsymbol{G}^k_{T^1}(\hat{v}_T)\|^2_{T^1} + h_T^{-1} \|\Pi^k_{(\partial T)^1}(v_{(\partial T)^1} - v_{T^1})\|^2_{(\partial T)^1} + h_T^{-1} \|[\![v_T]\!]_\Gamma\|^2_{T^\Gamma}.$$

Similar arguments lead to

$$\|\nabla v_{T^2}\|^2_{T^2} \lesssim \|\boldsymbol{G}^k_{T^2}(\hat{v}_T)\|^2_{T^2} + h_T^{-1} \|\Pi^k_{(\partial T)^2}(v_{(\partial T)^2} - v_{T^2})\|^2_{(\partial T)^2}.$$

Therefore we have $\sum_{i\in\{1,2\}} \kappa_i\|\nabla v_{T^i}\|_{T^i}^2 + \kappa_1 h_T^{-1}\|[\![v_T]\!]_\Gamma\|_{T^\Gamma}^2 \lesssim \hat{a}_T(\hat{v}_T,\hat{v}_T)$, and it remains to bound $\sum_{i\in\{1,2\}} \kappa_i h_T^{-1}\|v_{(\partial T)^i} - v_{T^i}\|_{(\partial T)^i}^2$. Let $i \in \{1,2\}$. Since $v_{(\partial T)^i} - v_{T^i} = \Pi_{(\partial T)^i}^k(v_{(\partial T)^i} - v_{T^i}) - (I - \Pi_{(\partial T)^i}^k)(v_{T^i})$ and recalling the definition (9) of the stabilization bilinear form $\hat{s}_{T^i}$, we only need to bound $h_T^{-1}\|(I-\Pi_{(\partial T)^i}^k)(v_{T^i})\|_{(\partial T)^i}^2$. Invoking the discrete trace inequality and the discrete Poincaré inequality from Lemma 4, we have

$$\begin{aligned}
&h_T^{-1}\|(I-\Pi_{(\partial T)^i}^k)(v_{T^i})\|_{(\partial T)^i}^2\\
&\leq h_T^{-1}\|(I-\Pi_{(\partial T)^i}^0)(v_{T^i})\|_{(\partial T)^i}^2 \leq h_T^{-1}\|(I-\Pi_{(\partial T)^i}^0)(v_{T^i}-\overline{v}_{T^i})\|_{(\partial T)^i}^2\\
&\leq h_T^{-1}\|v_{T^i}-\overline{v}_{T^i}\|_{(\partial T)^i}^2 \lesssim h_T^{-2}\|v_{T^i}-\overline{v}_{T^i}\|_{T^i}^2 \lesssim \|\nabla v_{T^i}\|_{T^i}^2,
\end{aligned}$$

where $\overline{v}_{T^i}$ is the mean value of v_{T^i} over $B(\tilde{\boldsymbol{x}}_{T^i}, h_T)$. This concludes the proof. □

Letting $\|\hat{v}_h\|_{\hat{a}_h}^2 := \sum_{T\in\mathcal{T}_h} |\hat{v}_T|_{\hat{a}_T}^2$, a classical argument shows that we define a norm on $\hat{U}_{h0}^k$ (recall that all the unknowns attached to boundary faces are null by definition of $\hat{U}_{h0}^k$). This norm is in spirit close to the broken H^1-norm used in discontinuous Galerkin methods.

Corollary 7 (Well-posedness). *If Assumption* 1 *is fulfilled, we have*

$$\|\hat{v}_h\|_{\hat{a}_h}^2 \lesssim \hat{a}_h(\hat{v}_h,\hat{v}_h), \qquad \forall \hat{v}_h \in \hat{U}_{h0}^k, \tag{18}$$

and the discrete problem (14) *is well-posed.*

Proof. The coercivity property (18) follows by summing over the mesh cells the bound from Lemma 6, and well-posedness follows from the Lax–Milgram Lemma. □

Remark 3.2. *(Weights)* The jump term $\|[\![v_T]\!]_\Gamma\|_{T^\Gamma}$ in (17) is weighted by the factor κ_1 and not κ_2 (recall that $\kappa_1 \leq \kappa_2$). This is possible because the gradient reconstruction in T^2 does not involve the jump across Γ.

3.3 Approximation

For all $v \in H^1(\Omega_1 \cup \Omega_2)$, we set $v_i := v_{|\Omega_i}$. To perform the error analysis, we assume that there is a real number $t \in (\frac{1}{2}, k+1]$ such that the exact solution satisfies $u_i \in H^{t+1}(\Omega_i)$ for all $i \in \{1,2\}$. Let $E_i : H^{t+1}(\Omega_i) \to H^{t+1}(\mathbb{R}^d)$, $i \in \{1,2\}$, be a stable extension operator. For all $T \in \mathcal{T}_h$, we define

$$I_{T^i}^{k+1}(v_i) := (\Pi_{T^\dagger}^{k+1}(E_i(v_i)))_{|T^i} \in \mathbb{P}^{k+1}(T^i), \tag{19}$$

$$\hat{I}_T^k(v) := (\hat{I}_{T^1}^k(v), \hat{I}_{T^2}^k(v)) := (I_{T^1}^{k+1}(v_1), \Pi_{(\partial T)^1}^k(v_1), I_{T^2}^{k+1}(v_2), \Pi_{(\partial T)^2}^k(v_2)) \in \hat{U}_T^k, \tag{20}$$

where $\Pi_{T^\dagger}^{k+1}$ denotes the L^2-orthogonal projector onto $\mathbb{P}^{k+1}(T^\dagger)$. Note that we do not project using the subcell T^i but the larger set $T^\dagger$ from Assumption 2 so as to invoke the optimal approximation properties of $I_{T^i}^{k+1}$ (see for instance [8, Lemma 5.6]). Indeed if this assumption is fulfilled, we have for all $v \in L^2(\Omega)$ such that $v_i \in H^{t+1}(\Omega_i)$, $i \in \{1,2\}$, and all $T \in \mathcal{T}_h$,

$$\|v_i - I_{T^i}^{k+1}(v_i)\|_{T^i} + h_T^{\frac{1}{2}}\|v_i - I_{T^i}^{k+1}(v_i)\|_{(\partial T)^i} + h_T\|\nabla(v_i - I_{T^i}^{k+1}(v_i))\|_{T^i} \\ + h_T^{\frac{3}{2}}\|\nabla(v_i - I_{T^i}^{k+1}(v_i))\|_{(\partial T)^i} \lesssim h_T^{t+1}|E_i(v_i)|_{H^{t+1}(T^\dagger)}, \tag{21}$$

$$h_T^{\frac{1}{2}}\|[\![v - I_T^{k+1}(v)]\!]_\Gamma\|_{T^\Gamma} \lesssim h_T^{t+1} \sum_{j\in\{1,2\}} |E_j(v_j)|_{H^{t+1}(T^\dagger)}. \tag{22}$$

It is convenient to define a discrete lifting operator $\boldsymbol{L}_{T^1}^k : L^2(\Gamma) \to \mathbb{G}^k(T^1;\mathbb{R}^d)$ for all $T \in \mathcal{T}_h^\Gamma$ such that, for all $g \in L^2(\Gamma)$ and all $\boldsymbol{q} \in \mathbb{G}^k(T^1;\mathbb{R}^d)$,

$$(\boldsymbol{L}_{T^1}^k(g), \boldsymbol{q})_{T^1} := (g, \boldsymbol{q}{\cdot}\boldsymbol{n}_\Gamma)_{T^\Gamma}. \tag{23}$$

With the above abuse of notation, we set $\boldsymbol{L}_{T^1}^k(g) := \boldsymbol{0}$ for all $T \in \mathcal{T}_h^i$, $i \in \{1,2\}$.

Lemma 8 (Approximation for gradient reconstruction)**.** *Let Assumptions* 1 *and* 2 *be fulfilled. For all* $v \in L^2(\Omega)$ *such that* $v_i \in H^{t+1}(\Omega_i)$*,* $i \in \{1,2\}$*, and for all* $T \in \mathcal{T}_h$*, we have*

$$\|\boldsymbol{\delta}_{T^1}(v)\|_{T^1} + h_T^{\frac{1}{2}}\|\boldsymbol{\delta}_{T^1}(v)\|_{(\partial T)^1\cup T^\Gamma} \lesssim h_T^t \sum_{j\in\{1,2\}} |E_j(v_j)|_{H^{t+1}(T^\dagger)}, \tag{24}$$

$$\|\boldsymbol{\delta}_{T^2}(v)\|_{T^2} + h_T^{\frac{1}{2}}\|\boldsymbol{\delta}_{T^2}(v)\|_{(\partial T)^2\cup T^\Gamma} \lesssim h_T^t|E_2(v_2)|_{H^{t+1}(T^\dagger)}, \tag{25}$$

with $\boldsymbol{\delta}_{T^1}(v) := \boldsymbol{G}_{T^1}^k(\hat{I}_T^k(v)) + \boldsymbol{L}_{T^1}^k([\![v]\!]_\Gamma) - \nabla v_1$ *and* $\boldsymbol{\delta}_{T^2}(v) := \boldsymbol{G}_{T^2}^k(\hat{I}_T^k(v)) - \nabla v_2$*, and with* $\boldsymbol{G}_{T^1}^k$ *and* $\boldsymbol{G}_{T^2}^k$ *defined in* (7) *and* (8)*, respectively. (Note that* $\boldsymbol{\delta}_{T^1}(v)$ *depends on* v_1 *and* v_2*, whereas* $\boldsymbol{\delta}_{T^2}(v)$ *only depends on* v_2*.)*

Proof. Let us set $\boldsymbol{\delta}'_{T^1}(v) := \boldsymbol{G}_{T^1}^k(\hat{I}_T^k(v)) + \boldsymbol{L}_{T^1}^k([\![v]\!]_\Gamma) - \nabla I_{T^1}^{k+1}(v_1)$. Owing to (7) and (23), we have

$$\|\boldsymbol{\delta}'_{T^1}(v)\|_{T^1}^2 = (\Pi_{(\partial T)^1}^k(v_1) - I_{T^1}^{k+1}(v_1), \boldsymbol{\delta}'_{T^1}(v){\cdot}\boldsymbol{n}_T)_{(\partial T)^1} + ([\![v - I_T^{k+1}(v)]\!]_\Gamma, \boldsymbol{\delta}'_{T^1}(v){\cdot}\boldsymbol{n}_\Gamma)_{T^\Gamma}.$$

Since $(\boldsymbol{\delta}'_{T^1}(v){\cdot}\boldsymbol{n}_T)_{|(\partial T)^1} \in \mathbb{P}^k(\mathcal{F}_{(\partial T)^1})$ owing to the assumption on the reconstruction space $\mathbb{G}^k(T^i;\mathbb{R}^d)$, we can replace $\Pi_{(\partial T)^1}^k(v_1)$ by v_1 in the first term on the right-hand side. The Cauchy–Schwarz inequality and the discrete trace inequality from Lemma 4 lead to

$$\|\boldsymbol{\delta}'_{T^1}(v)\|_{T^1} + h_T^{\frac{1}{2}}\|\boldsymbol{\delta}'_{T^1}(v)\|_{(\partial T)^1\cup T^\Gamma} \lesssim \|\boldsymbol{\delta}'_{T^1}(v)\|_{T^1} \\ \lesssim h_T^{-\frac{1}{2}}\Big(\|v_1 - I_{T^1}^{k+1}(v_1)\|_{(\partial T)^1} + \|[\![v - I_T^{k+1}(v)]\!]_\Gamma\|_{T^\Gamma}\Big).$$

Invoking (21)-(22) together with $\boldsymbol{\delta}_{T^1}(v) = \boldsymbol{\delta}'_{T^1}(v) + \nabla(I_{T^1}^{k+1}(v_1) - v_1)$ and a triangle inequality then proves (24). The proof of (25) uses similar arguments. □

3.4 Error estimate

Recall that u and $\hat{u}_h$ are the solutions to (1) and (14), respectively. We set $u_i := u_{|\Omega_i}$, $i \in \{1,2\}$.

Lemma 9 (Consistency and boundedness)**.** *For all $\hat{v}_h \in \hat{U}^k_{h0}$, we define the consistency error as $\mathcal{D}_h(\hat{v}_h) := \sum_{T\in\mathcal{T}_h} \Big\{\hat{a}_T(\hat{I}^k_T(u), \hat{v}_T) - \hat{\ell}_T(\hat{v}_T)\Big\}$. Assume that there is $t \in (\frac{1}{2}, k+1]$ such that $u_i \in H^{1+t}(\Omega_i)$ for all $i \in \{1,2\}$. Then we have*

$$|\mathcal{D}_h(\hat{v}_h)| \lesssim \Big(\sum_{T\in\mathcal{T}_h} \|\boldsymbol{g}_T\|^2_{*T} + \|\xi_T\|^2_{\#T}\Big)^{\frac{1}{2}} \times \|\hat{v}_h\|_{\hat{a}_h},$$

where

$$\begin{aligned}\|\boldsymbol{g}_T\|^2_{*T} &:= \kappa_1\Big(\|\boldsymbol{\delta}_{T^1}(u)\|^2_{T^1} + h_T\|\boldsymbol{\delta}_{T^1}(u)\|^2_{(\partial T)^1\cup T^\Gamma}\Big)\\ &\qquad + \kappa_2\Big(\|\boldsymbol{\delta}_{T^2}(u)\|^2_{T^2} + h_T\|\boldsymbol{\delta}_{T^2}(u)\|^2_{(\partial T)^2}\Big), \qquad (26)\\ \|\xi_T\|^2_{\#T} &:= \sum_{i\in\{1,2\}} \kappa_i h_T^{-1}\|u_i - I^{k+1}_{T^i}(u_i)\|^2_{(\partial T)^i} + \kappa_1 h_T^{-1}\|[\![u - I^{k+1}_T(u)]\!]_\Gamma\|^2_{T^\Gamma}, \qquad (27)\end{aligned}$$

with $\boldsymbol{\delta}_{T^1}(u) := \boldsymbol{G}^k_{T^1}(\hat{I}^k_T(u)) + \boldsymbol{L}^k_{T^1}(g_D) - \nabla u_1$ and $\boldsymbol{\delta}_{T^2}(u) := \boldsymbol{G}^k_{T^2}(\hat{I}^k_T(u)) - \nabla u_2$.

Proof. Re-arranging the terms and using the PDE satisfied by the exact solution in $\Omega_1\cup\Omega_2$, we have $\mathcal{D}_h(\hat{v}_h) = \Psi_1 + \Psi_2$ with

$$\begin{aligned}\Psi_1 &:= \sum_{T\in\mathcal{T}_h}\bigg\{\sum_{i\in\{1,2\}}\Big(\kappa_i(\boldsymbol{G}^k_{T^i}(\hat{I}^k_T(u)), \boldsymbol{G}^k_{T^i}(\hat{v}_T))_{T^i} + (\nabla\cdot(\kappa_i\nabla u_i), v_{T^i})_{T^i}\Big)\\ &\qquad + \kappa_1(\boldsymbol{L}^k_{T^1}(g_D), \boldsymbol{G}^k_{T^1}(\hat{\boldsymbol{v}}_T))_{T^1} - (g_N, v_{T^2})_{T^\Gamma}\bigg\},\\ \Psi_2 &:= \sum_{T\in\mathcal{T}_h}\sum_{i\in\{1,2\}} \hat{s}_{T^i}(\hat{I}^k_{T^i}(u), \hat{v}_{T^i}) + \kappa_1 h_T^{-1}([\![I^{k+1}_T(u)]\!]_\Gamma - g_D, [\![v_T]\!]_\Gamma)_{T^\Gamma}.\end{aligned}$$

Using an integration by parts for $\nabla\cdot(\kappa_i\nabla u_i)$, the definitions (7)-(8) of the gradient reconstructions and the fact that $\sum_{T\in\mathcal{T}_h}(v_{(\partial T)^i}, \nabla u_i\cdot\boldsymbol{n}_T)_{(\partial T)^i} = 0$, $i \in \{1,2\}$ (since $u_i \in H^{\frac{3}{2}+\varepsilon}(\Omega_i)$ by assumption), we infer that

$$\begin{aligned}\Psi_1 = &\sum_{T\in\mathcal{T}_h}\bigg\{ - \kappa_1((\boldsymbol{G}^k_{T^1}(\hat{I}^k_T(u)) + \boldsymbol{L}^k_{T^1}(g_D))\cdot\boldsymbol{n}_\Gamma, [\![v_T]\!]_\Gamma)_{T^\Gamma}\\ &+ (\kappa_1\nabla u_1\cdot\boldsymbol{n}_\Gamma, v_{T^1})_{T^\Gamma} - (\kappa_2\nabla u_2\cdot\boldsymbol{n}_\Gamma, v_{T^2})_{T^\Gamma} + ((\kappa_2\nabla u_2 - \kappa_1\nabla u_1)\cdot\boldsymbol{n}_\Gamma, v_{T^2})_{T^\Gamma}\\ &+ \kappa_1(\boldsymbol{\delta}_{T^1}(u), \nabla v_{T^1})_{T^1} + \kappa_1(\boldsymbol{\delta}_{T^1}(u)\cdot\boldsymbol{n}_T, v_{(\partial T)^1} - v_{T^1})_{(\partial T)^1})\\ &+ \kappa_2(\boldsymbol{\delta}_{T^2}(u), \nabla v_{T^2})_{T^2} + \kappa_2(\boldsymbol{\delta}_{T^2}(u)\cdot\boldsymbol{n}_T, v_{(\partial T)^2} - v_{T^2})_{(\partial T)^2}\bigg\}\\ = &\sum_{T\in\mathcal{T}_h}\bigg\{ - \kappa_1(\boldsymbol{\delta}_{T^1}(u)\cdot\boldsymbol{n}_\Gamma, [\![v_T]\!]_\Gamma)_{T^\Gamma}\\ &+ \sum_{i\in\{1,2\}} \kappa_i(\boldsymbol{\delta}_{T^i}(u), \nabla v_{T^i})_{T^i} + \kappa_i(\boldsymbol{\delta}_{T^i}(u)\cdot\boldsymbol{n}_T, v_{(\partial T)^i} - v_{T^i})_{(\partial T)^i}\bigg\}.\end{aligned}$$

Moreover, recalling the definition of the stabilization bilinear forms $\hat{s}_{T^i}$, that of $\hat{I}^k_{T^i}(u)$, and since $[\![u]\!]_\Gamma = g_D$, we have

$$\Psi_2 = \sum_{T\in\mathcal{T}_h} \Bigg\{ \sum_{i\in\{1,2\}} \kappa_i h_T^{-1}(\Pi^k_{(\partial T)^i}(u_i - I^{k+1}_{T^i}(u)), v_{(\partial T)^i} - v_{T^i})_{(\partial T)^i} \\ + \kappa_1 h_T^{-1}([\![I^{k+1}_T(u) - u]\!]_\Gamma, [\![v_T]\!]_\Gamma)_{T^\Gamma} \Bigg\}.$$

We conclude the proof by invoking the Cauchy–Schwarz inequality and the definition of $\|\hat{v}_h\|_{\hat{a}_h}$. □

Theorem 10 (Error estimate)**.** *Let Assumption* 1 *be fulfilled. Assume that there is* $t \in (\frac{1}{2}, k+1]$ *such that* $u_i \in H^{t+1}(\Omega_i)$ *for all* $i \in \{1,2\}$*. Then we have*

$$\sum_{T\in\mathcal{T}_h}\sum_{i\in\{1,2\}} \kappa_i \|\nabla(u_i - u_{T^i})\|^2_{T^i} \lesssim \sum_{T\in\mathcal{T}_h}\sum_{i\in\{1,2\}} \kappa_i \|\nabla(u_i - I^{k+1}_{T^i}(u_i))\|^2_{T^i} \\ + \sum_{T\in\mathcal{T}_h} \Big(\|\boldsymbol{g}_T\|^2_{*T} + \|\xi_T\|^2_{\#T} \Big), \qquad (28)$$

with $\|\boldsymbol{g}_T\|_{*T}$ *and* $\|\xi_T\|_{\#T}$ *defined in* (26)-(27)*, respectively. Moreover, if Assumptions* 2 *and* 3 *are also fulfilled, then we have*

$$\sum_{T\in\mathcal{T}_h}\sum_{i\in\{1,2\}} \kappa_i \|\nabla(u_i - u_{T^i})\|^2_{T^i} \lesssim h^{2t} \sum_{i\in\{1,2\}} \kappa_i |u_i|^2_{H^{t+1}(\Omega_i)}. \qquad (29)$$

Proof. For all $T \in \mathcal{T}_h$, we define the discrete error $\hat{e}_h \in \hat{U}^k_{h0}$ such that $\hat{e}_T := \hat{I}^k_T(u) - \hat{u}_T \in \hat{U}^k_T$ for all $T \in \mathcal{T}_h$. Then we have $\mathcal{D}_h(\hat{e}_h) = \hat{a}_h(\hat{e}_h, \hat{e}_h)$. Lemmas 6 (stability) and 9 (consistency and boundedness) give

$$\|\hat{e}_h\|_{\hat{a}_h} \lesssim \left(\sum_{T\in\mathcal{T}_h} \|\boldsymbol{g}_T\|^2_{*T} + \|\xi_T\|^2_{\#T} \right)^{\frac{1}{2}}.$$

Since $\nabla(u_i - u_{T^i}) = \nabla(u_i - I^{k+1}_{T^i}(u_i)) + \nabla e_{T^i}$, invoking the triangle inequality leads to (28). Let us now prove (29). The bounds (21)-(22) and Lemma 8 (approximation for gradient reconstruction) imply that

$$\sum_{T\in\mathcal{T}_h}\sum_{i\in\{1,2\}} \kappa_i \|\nabla(u_i - u_{T^i})\|^2_{T^i} \lesssim h^{2t} \sum_{T\in\mathcal{T}_h}\sum_{i\in\{1,2\}} \kappa_i |E_i(u_i)|^2_{H^{t+1}(T^\dagger)}.$$

Since $T^\dagger \subset \Delta_{n_0}(T)$ for all $T \in \mathcal{T}_h$, owing to Assumption 3, we infer that for all $i \in \{1,2\}$, $\sum_{T\in\mathcal{T}_h} |E_i(u_i)|^2_{H^{t+1}(T^\dagger)} \lesssim |u_i|^2_{H^{t+1}(\Omega_i)}$. This concludes the proof. □

4 Implementation aspects

The unfitted HHO method is implemented in `ProtoN`, which is a slim, fast prototyping library to test major modifications within the `DiSk++` library[1] [15]. For simplicity we focus on a two-dimensional setting with unfitted meshes composed of quadrangles. We assume that the boundary of every cell is cut at most twice and that every face is cut at most once. These assumptions are classical when implementing unfitted methods (see for instance [26]) and can be satisfied if the unfitted mesh is fine enough. Moreover we assume that the interface is specified by means of a level-set function.

As in `DiSk++`, polynomial basis functions associated with the mesh cells and faces are used. The cell polynomials are centered at the barycenter of the cell and are scaled by its diameter. The face basis functions associated with a subface F^i, $i \in \{1,2\}$, are taken to be Legendre polynomials scaled by $|F^i|^{-1/2}$. This eases the computation of the stabilization term since the face mass matrix that has to be inverted is the identity matrix. Moreover this improves the conditioning of the stiffness matrix.

Remark 4.1. *(3D)* Since the fitted polyhedral HHO method is already implemented for 3D problems in `DiSk++`, developing 3D unfitted HHO methods essentially requires to take into account the richer cut configurations that can occur.

4.1 Mesh data structure and preprocessing

The mesh data structure hinges on the features developed for fitted HHO methods (see [15, Section 3]). It is independent from the space dimension and from the shape of the elements. Compared with `DiSk++`, some changes in the mesh data structure were introduced so as to provide additional information concerning the cut cells. This information which is stored for each cell contains:

- The status of the cell (in $\mathcal{T}_h^1$, $\mathcal{T}_h^2$ or $\mathcal{T}_h^\Gamma$);
- A flag indicating whether or not the cell has a small cut and on which side of the interface the small cut occurs. The criterion for a small cut is (in a slight variation with respect to the requirement of Assumption 1)

$$|T^i| \leq \alpha |T|, \tag{30}$$

  with user-parameter $\alpha \in (0, \frac{1}{2})$ (we set $\alpha := 0.3$ in Section 5).
- The list of neighboring cells to be used in the agglomeration procedure;
- A list of points representing the interface to be used for numerical integration;
- A list of quadrature weights and points if the cell results from agglomeration.

In practice the mesh preprocessing follows the following steps:

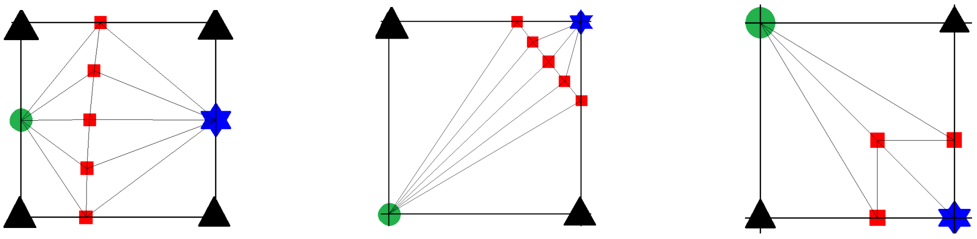
[1] see `https://github.com/wareHHOuse`

- We detect in which subdomain are the nodes, the faces, and the cells of the mesh and whether or not they are cut. Moreover we compute the intersection points between the faces and the interface. This initializes the representation of the interface inside the cut cells.
- We use the criterion (30) to select the cells that have to be agglomerated. We also store the knowledge of which subcell is small. This is used when we need to find suitable neighbors to these cells.
- For every small cut cell, we compute the list of its neighbors, that is
$$\Delta(T) := \Delta_1(T) \setminus \{T\}, \tag{31}$$
i.e., $\Delta(T)$ is the collection of the cells distinct from T that share at least a point with T. It is useful to store separately the neighbors sharing a face and the neighbors sharing only a node. Since we consider Cartesian meshes, this routine can be specialized for optimization.
- We refine the representation of the interface. For every cell, the segment representation that was defined on the first step above is divided into $2^{n_{\text{int}}}$ segments for a user-specified parameter $n_{\text{int}} \geq 1$. For instance, $n_{\text{int}} = 1$ in Fig. 2c and $n_{\text{int}} = 2$ in Fig. 2a and 2b. In practice we want to have n_{int} large enough to mitigate the geometric error induced from the fact that we consider a low-order representation of the interface in combination with a high-order discretization method (see Fig. 8 for an illustration). The position of the points on the interface is determined by a bisection method using the level-set representation of the interface.
- We finally agglomerate the small cut cells as described in Section 4.3.

4.2 Quadratures

During the assembly phase (and to compute errors when reporting convergence rates), integrals over the subcells T^i, $i \in \{1, 2\}$, and over the interface T^Γ need to be evaluated for every cut cell $T \in \mathcal{T}_h^\Gamma$. This operation is done by using the list of points representing the interface (the red squares in Fig. 2). The quadrature points and weights over the interface are obtained by gathering all the Gauss–Legendre quadrature points and weights over all the segments composing this list of points. Furthermore, each subcell is decomposed into several triangles, and the quadrature points and weights for the subcell are then obtained by gathering all the Dunavant [22] quadrature points and weights of the subtriangles. In order to build this subtriangulation, a point is fixed and triangles are formed such that one vertex is this fixed point and the two other vertices are vertices of the polygonal cut subcell T^i, $i \in \{1, 2\}$. Hence we use points located on $(\partial T)^i$ and on T^Γ, i.e., the black triangles and the red squares shown in Fig. 2. In practice, since we consider quadrangular meshes, the fixed point is either the midpoint of a face (if two opposite faces are cut, see Fig. 2a) or a vertex of the cut cell (if two adjacent faces are cut, see Fig. 2b and 2c). This choice of fixed points can reduce the number of negative weights if the mesh is fine enough.

Remark 4.2. *(Alternative)* One can also devise quadratures from an isoparametric representation for the interface, as, e.g., in [32].



(a) Cutting opposite faces. (b) Cutting adjacent faces. (c) Interface with corner.

Figure 2: Subtriangulation of a cut cell. The interface is represented by red squares. The center of subtriangulation of the subcells is indicated by a blue star and a green circle. The other vertices of the subcells are indicated by black triangles.

4.3 The cell agglomeration procedure

The starting point of the procedure is an initial shape-regular mesh fulfilling Assumption 2, but not necessarily Assumption 1. We consider the partition $\mathcal{T}_h = \mathcal{T}_h^1 \cup \mathcal{T}_h^\Gamma \cup \mathcal{T}_h^2$ as in (2). We further partition the set of the cut cells as

$$\mathcal{T}_h^\Gamma = \mathcal{T}_h^{\mathrm{OK}} \cup \mathcal{T}_h^{\mathrm{KO},1} \cup \mathcal{T}_h^{\mathrm{KO},2}, \tag{32}$$

where $\mathcal{T}_h^{\mathrm{OK}}$ contains all the cut cells without small cuts and $\mathcal{T}_h^{\mathrm{KO},i}$ contains all the cut cells with a small subcell in Ω_i, $i \in \{1,2\}$. Recall that the subsets $\mathcal{T}_h^{\mathrm{KO},1}$ and $\mathcal{T}_h^{\mathrm{KO},2}$ are disjoint owing to [8, Lemma 6.2] if the unfitted mesh is fine enough and the parameter δ in Assumption 1 is small enough. Here the small-cut criterion is the simpler condition (30). Nonetheless we observed in all our experiments that the subsets $\mathcal{T}_h^{\mathrm{KO},1}$ and $\mathcal{T}_h^{\mathrm{KO},2}$ were indeed disjoint.

The cell agglomeration procedure needs to fulfill two goals. First every agglomerated cell must be free of small cuts. Second, if two cells T and T' are agglomerated, they must have a common neighbor. Note that this second condition allows for diagonal agglomeration if needed (see for instance Fig. 4). The procedure is outlined in Algorithm 1. In stage 1, for every cell T in $\mathcal{T}_h^{\mathrm{KO},1}$ (having a small cut in Ω_1), we find a neighbor $\mathcal{N}_1(T) \in (\mathcal{T}_h^1 \cup \mathcal{T}_h^{\mathrm{OK}} \cup \mathcal{T}_h^{\mathrm{KO},2}) \cap \Delta(T)$. The existence of $\mathcal{N}_1(T)$ follows from [8, Lemma 6.3] (if the mesh is fine enough and the parameter δ small enough), and by construction the agglomerated cell $T \cup \mathcal{N}_1(T)$ has no small cuts. We denote $\widetilde{\mathcal{T}_h}^{\mathrm{KO},2}$ the subset composed of the cells in $\mathcal{T}_h^{\mathrm{KO},2}$ that have not been chosen as neighbors during stage 1. In stage 2, for every cell T in $\widetilde{\mathcal{T}_h}^{\mathrm{KO},2}$ (having a small cut in Ω_2 and that has not yet been agglomerated after stage 1), we find a neighbor $\mathcal{N}_2(T) \in (\mathcal{T}_h^2 \cup \mathcal{T}_h^{\mathrm{OK}} \cup \mathcal{T}_h^{\mathrm{KO},1}) \cap \Delta(T)$. The existence of $\mathcal{N}_2(T)$ again follows from [8, Lemma 6.3] under the same conditions, and by construction $T \cup \mathcal{N}_2(T)$ has no small cuts. However the actual agglomerated cell may be larger than $T \cup \mathcal{N}_2(T)$ (i.e., some chain of agglomerations can occur), and the goal of stage 3 is to modify $\mathcal{N}_1(T)$ for some cells in $\mathcal{T}_h^{\mathrm{KO},1}$ to avoid this. The idea is that if a cell in $\mathcal{T}_h^{\mathrm{KO},1}$ is at the same time pointing (by means of $\mathcal{N}_1$ in stage 1) to a cell in $\mathcal{T}_h^{\mathrm{KO},2}$ and is pointed to by a cell in $\widetilde{\mathcal{T}_h}^{\mathrm{KO},2}$ (by means of $\mathcal{N}_2$ in stage 2), then the pointer $\mathcal{N}_1(T)$ is changed unless the cell $\mathcal{N}_1(T)$ belongs to $\mathcal{T}_h^{\mathrm{KO},2}$ and is not pointed to by another cell. After completion of stage 3, all the agglomerated cells are collected in the subset $\mathcal{T}_h^{\text{agglo}}$, and the final mesh produced by the procedure is

$$\widetilde{\mathcal{T}_h} = ((\mathcal{T}_h \backslash \{T, \mathcal{N}_1(T) \mid T \in \mathcal{T}_h^{\text{KO},1}\}) \backslash \{T, \mathcal{N}_2(T) \mid T \in \widetilde{\mathcal{T}_h}^{\text{KO},2}\}) \cup \mathcal{T}_h^{\text{agglo}}. \tag{33}$$

Algorithm 1 Cell agglomeration procedure.

```
Require: Mesh $\mathcal{T}_h$ satisfying Assumption 2.
1. For every $T \in \mathcal{T}_h^{\text{KO},1}$, find a suitable neighbor $\mathcal{N}_1(T) \in (\mathcal{T}_h^1 \cup \mathcal{T}_h^{\text{OK}} \cup \mathcal{T}_h^{\text{KO},2}) \cap \Delta(T)$.
2. For every $T \in \widetilde{\mathcal{T}_h}^{\text{KO},2}$, find a suitable neighbor $\mathcal{N}_2(T) \in (\mathcal{T}_h^2 \cup \mathcal{T}_h^{\text{OK}} \cup \mathcal{T}_h^{\text{KO},1}) \cap \Delta(T)$.
3. For every $T \in \mathcal{T}_h^{\text{KO},1}$, if the following conditions are met:
    (3.a) there is $T' \in \widetilde{\mathcal{T}_h}^{\text{KO},2}$ such that $\mathcal{N}_2(T') = T$;
    (3.b) $\mathcal{N}_1(T) \notin \mathcal{T}_h^{\text{KO},2}$ or there is $T'' \in \mathcal{T}_h^{\text{KO},1} \backslash \{T\}$ such that $\mathcal{N}_1(T) = \mathcal{N}_1(T'')$;
    then set $\mathcal{N}_1(T) := T'$. Otherwise do not modify $\mathcal{N}_1(T)$.
4. Generate a new mesh $\widetilde{\mathcal{T}_h}$ as in (33).
```

Lemma 11 (Cell agglomeration)**.** *Every agglomerated cell is in $\Delta_1(T^\#)$ for some $T^\# \in \mathcal{T}_h$.*

Proof. Assume that a cell $T \in \mathcal{T}_h^{\text{KO},1}$ is chosen by a cell $T' \in \widetilde{\mathcal{T}_h}^{\text{KO},2}$, i.e., $T = \mathcal{N}_2(T')$, so that the condition (3.a) of Algorithm 1 is fulfilled. By definition of the subset $\widetilde{\mathcal{T}_h}^{\text{KO},2}$, T' has not been chosen by any cell in stage 1. Hence there is no propagation of the agglomeration in the direction of T'. Let us now consider the propagation in the direction of T. Recall that according to stage 1, $\mathcal{N}_1(T)$ is in $\mathcal{T}_h^1 \cup \mathcal{T}_h^{\text{OK}} \cup \mathcal{T}_h^{\text{KO},2}$. If actually $\mathcal{N}_1(T) \in \mathcal{T}_h^1 \cup \mathcal{T}_h^{\text{OK}}$, then condition (3.b) is fulfilled so that we set $\mathcal{N}_1(T) := T'$. Hence the agglomerated cell contains T and T', and potentially any other cell $\widetilde{T}_2 \in \widetilde{\mathcal{T}_h}^{\text{KO},2}$ such that $\mathcal{N}_2(\widetilde{T}_2) = T$. Hence all these cells are in $\Delta_1(T)$. Otherwise $\mathcal{N}_1(T) \in \mathcal{T}_h^{\text{KO},2}$. Then, either there is $T'' \in \mathcal{T}_h^{\text{KO},1}$ such that $\mathcal{N}_1(T) = \mathcal{N}_1(T'')$. In this case condition (3.b) is again fulfilled, so that we set $\mathcal{N}_1(T) := T'$, and the agglomerated cell is again in $\Delta_1(T)$. Otherwise, there is no such T'' so that condition (3.b) is not fulfilled, which means that the agglomerated cell is composed of T', $T = \mathcal{N}_2(T')$, $\mathcal{N}_1(T)$, and potentially any other cell $\widetilde{T}_2 \in \widetilde{\mathcal{T}_h}^{\text{KO},2}$ such that $\mathcal{N}_2(\widetilde{T}_2) = T$. Again all these cells are in $\Delta_1(T)$. Finally the reasoning is similar if one considers a cell $T \in \mathcal{T}_h^{\text{OK}} \cup \mathcal{T}_h^1 \cup \mathcal{T}_h^2$ which is pointed to by a cell from $\mathcal{T}_h^{\text{KO},1}$ and a cell from $\widetilde{\mathcal{T}_h}^{\text{KO},2}$. □

Remark 4.3. *(Stage 3)* Algorithm 1 without stage 3 has been proposed in [8, Section 6]. In this case one can show that every agglomerated cell is in $\Delta_2(T^\#)$ for some $T^\# \in \mathcal{T}_h$ (instead of $\Delta_1(T^\#)$). Indeed, a cell T in $\mathcal{T}_h^{\text{KO},2}$ can choose a cell T' in $\mathcal{T}_h^{\text{KO},1}$ that chooses a cell $T^\#$ in $\mathcal{T}_h^{\text{KO},2} \cup \mathcal{T}_h^1 \cup \mathcal{T}_h^{\text{OK}}$. And similarly, another cell $\tilde{T}$ in $\mathcal{T}_h^{\text{KO},2}$ can choose a cell $\tilde{T}'$ in $\mathcal{T}_h^{\text{KO},1}$ that chooses $T^\#$. The agglomeration cannot propagate further since if a cell in $\mathcal{T}_h^{\text{KO},1}$ chooses T as a neighbor, then T does not look for a neighbor and does not choose

T' (see the definition of $\widetilde{\mathcal{T}_h}^{\mathrm{KO},2}$). Furthermore, another benefit of stage 3 is the reduction whenever possible of the size of the agglomerated cells by promoting the agglomeration of pairs of cells, one from $\mathcal{T}_h^{\mathrm{KO},1}$ and the other from $\widetilde{\mathcal{T}_h}^{\mathrm{KO},2}$. See Figure 5 for an illustration of the three stages.

Remark 4.4. *(Choice of neighbor)* When looking for a neighbor in stages 1 and 2 in Algorithm 1, picking a neighbor that shares at least one common face is preferred. If there are no such neighbors, then diagonal agglomeration by means of a neighbor sharing only a point is performed (see Fig. 4). Moreover, in the case where several cells sharing at least one common face (resp., no common faces) with some $T \in \mathcal{T}_h^{\mathrm{KO},i}$ are available, the cell having the smallest (positive) area in $\Omega_{\bar{\imath}}$ (i.e. the smallest $|T^{\bar{\imath}}|$) is chosen. This way, the cells in $\mathcal{T}_h^{\mathrm{KO},1}$ tend (whenever possible) to be agglomerated with cells from $\mathcal{T}_h^{\mathrm{KO},2}$. This reduces the total number of agglomerations as well as the triggering of stages 2 and 3.

5 Numerical experiments

We present numerical experiments to illustrate the cell agglomeration procedure and the convergence rates of Theorem 10. We also study the condition number of the stiffness matrix (after static condensation).

In all the experiments we consider the unit square domain $\Omega := (0,1)^2$ with homogeneous Cartesian meshes of step size $h = 2^{-N}$ with $N \in \{3,4,5,6,7\}$. Note that here the mesh size h refers to the number of subdivisions of each side of Ω before the cell agglomeration procedure. The interface is represented by a level-set function Φ so that $\Gamma := \{(x,y) \in \Omega \mid \Phi(x,y) = 0\}$, $\Omega_1 := \{(x,y) \in \Omega \mid \Phi(x,y) < 0\}$, and $\Omega_2 := \{(x,y) \in \Omega \mid \Phi(x,y) > 0\}$. We consider a circular interface and a flower-like interface for which the level-set functions are, respectively,

$$\Phi_C(x,y) := (x-a)^2 + (y-b)^2 - R^2, \tag{34}$$

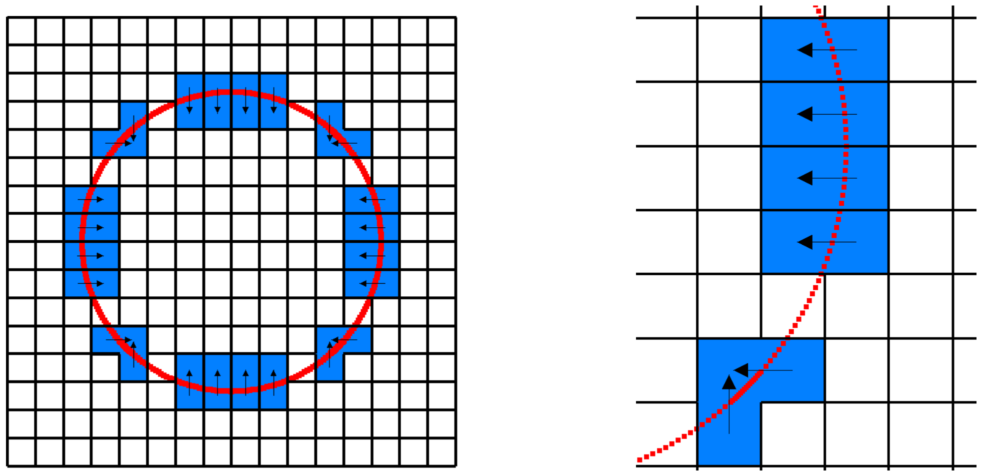
$$\Phi_F(x,y) := (x-a)^2 + (y-b)^2 - R^2 + c\cos(n\theta), \tag{35}$$

with $\theta := \arctan(\frac{y-b}{x-a})$ if $x \geq a$, $\theta := \pi + \arctan(\frac{y-b}{x-a})$ if $x < a$, where $a, b, c \in (0,1)$.

5.1 Agglomeration procedure

We test the agglomeration procedure detailed in Section 4.3. The circular interface is first considered with $R := 1/3$ and $a = b := 0.5$ in (34). We plot the interface and the mesh obtained after the cell agglomeration procedure in Fig. 3. As expected each agglomerated cell is in the neighborhood of one of the cells of the original mesh, i.e., there is no propagation of the agglomeration. We also remark that all the agglomerations have been done by using cells sharing one face (see Remark 4.4).

We give more details on the cell agglomeration procedure in Table 1. We note that for meshes that are fine enough, there are almost as many cells in $\mathcal{T}_h^{\mathrm{KO},1}$ as in $\mathcal{T}_h^{\mathrm{KO},2}$, each subset being composed of about 35% of the cells in $\mathcal{T}_h^{\Gamma}$. However, since the cells in $\mathcal{T}_h^{\mathrm{KO},1}$ look in priority for neighbors in $\mathcal{T}_h^{\mathrm{KO},2}$, there are much fewer cells in $\widetilde{\mathcal{T}_h}^{\mathrm{KO},2}$ than in $\mathcal{T}_h^{\mathrm{KO},2}$.



(a) The full mesh.

(b) A zoom.

Figure 3: Circular interface ($h = 1/16$). The interface is represented by red squares. Agglomerated cells are highlighted in blue. The arrows indicate how the agglomerations have been performed.

h^{-1}	$\mathcal{T}_h$	$\mathcal{T}_h^{\Gamma}$	$\mathcal{T}_h^{\mathrm{OK}}$	$\mathcal{T}_h^{\mathrm{KO},1}$	$\mathcal{T}_h^{\mathrm{KO},2}$	$\widetilde{\mathcal{T}}_h^{\mathrm{KO},2}$	$\widetilde{\mathcal{T}}_h^{\mathrm{KO},1}$
8	64	20	8	8	4	0	0
16	256	44	8	24	12	0	0
32	1024	84	44	24	16	0	0
64	4096	172	56	68	48	8	0
128	16384	340	120	108	112	24	0
256	65536	684	184	260	240	48	0

Table 1: Circular interface. Details on the cell agglomeration process for the various unfitted meshes: number of mesh cells, cut cells, and cells in the subsets $\mathcal{T}_h^{\mathrm{OK}}$, $\mathcal{T}_h^{\mathrm{KO},1}$, $\mathcal{T}_h^{\mathrm{KO},2}$, $\widetilde{\mathcal{T}}_h^{\mathrm{KO},2}$ and $\widetilde{\mathcal{T}}_h^{\mathrm{KO},1}$.

We also notice that stage 3 of Algorithm 1 is never active here, i.e., $\#(\widetilde{\mathcal{T}_h}^{\mathrm{KO},1}) = 0$, where $\widetilde{\mathcal{T}}_h^{\mathrm{KO},1}$ is the collection of the cells $T \in \mathcal{T}_h^{\mathrm{KO},1}$ such that $\mathcal{N}_1(T)$ is modified during stage 3.

We now test a flower-like interface with the parameters $R := 1/3$, $a := 0.47$, $b := 0.46$, $n := 12$ and $c := 0.015$ in (35). We present in Fig. 4 the mesh after the agglomeration process. As expected the agglomerated cells are contained in the neighborhood of one cell of the original mesh. Interestingly, we observe that one agglomeration has been done with a diagonal neighbor, i.e., a neighbor that shares a node and no faces. Indeed, considering the right panel of Fig. 4, the cell in the bottom left corner is in $\mathcal{T}_h^{\mathrm{KO},2}$ and all its neighbors sharing at least one face are either in $\mathcal{T}_h^{\mathrm{KO},2}$ or in $\mathcal{T}_h^{1}$. A diagonal agglomeration is then considered.

We give more details on the cell agglomeration procedure in Table 2. The results are very similar to those reported in Table 1. The main difference is that stage 3 of Algorithm 1

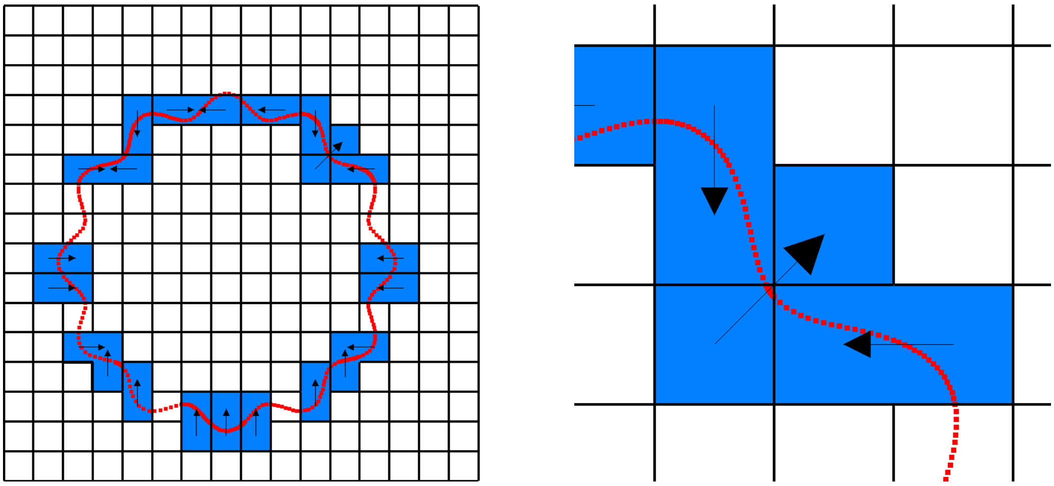
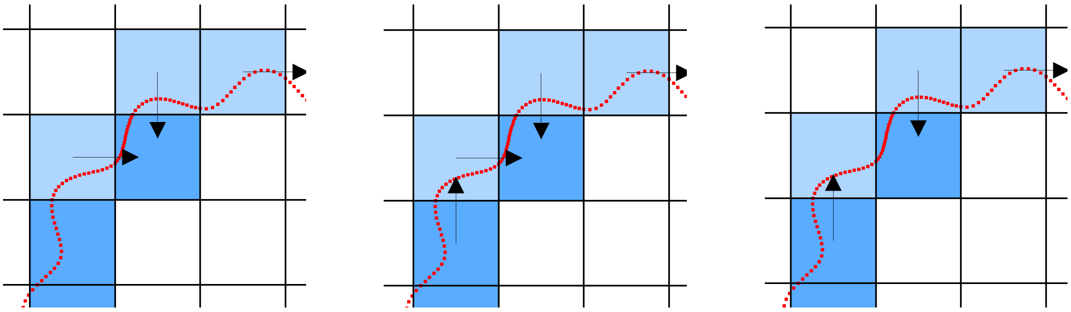


Figure 4: Flower-like interface ($h = 1/16$). The interface is represented by red squares. Agglomerated cells are highlighted in blue. The arrows indicate how the agglomerations have been performed. Left: global view. Right: zoom near the cell where a diagonal agglomeration has occurred.

is active in some cases. A zoom on the cells where this stage is used is provided in Fig. 5 (for $h = 1/8$). The light blue cells are the ones in $\mathcal{T}_h^{\text{KO},1}$, the dark blue cells are the ones in $\mathcal{T}_h^{\text{KO},2}$. A black arrow pointing from T to T' means that T chooses T' as a neighbor for the agglomeration. The interface is indicated by red points. We can see on the left of the picture that a cell in $\mathcal{T}_h^{\text{KO},1}$ is chosen as a neighbor in stage 2. This cell had already chosen another neighbor during stage 1. Since this other neighbor is also chosen by another cell in $\mathcal{T}_h^{\text{KO},1}$ (the one at the top of the picture), stage 3 is active and the cell at the left does not agglomerate any more with the cell in the center. This reduces the size of the agglomerated cells.

h^{-1}	$\mathcal{T}_h$	$\mathcal{T}_h^{\Gamma}$	$\mathcal{T}_h^{\text{OK}}$	$\mathcal{T}_h^{\text{KO},1}$	$\mathcal{T}_h^{\text{KO},2}$	$\widetilde{\mathcal{T}}_h^{\text{KO},2}$	$\widetilde{\mathcal{T}}_h^{\text{KO},1}$
8	64	22	4	11	7	2	1
16	256	46	13	19	14	3	0
32	1024	100	24	40	36	14	3
64	4096	194	42	79	73	27	3
128	16384	388	119	139	130	48	1
256	65536	784	221	286	277	95	0

Table 2: Flower-like interface. Details on the cell agglomeration process for the various unfitted meshes: number of mesh cells, cut cells and cells in the subsets $\mathcal{T}_h^{\text{OK}}$, $\mathcal{T}_h^{\text{KO},1}$, $\mathcal{T}_h^{\text{KO},2}$, $\widetilde{\mathcal{T}}_h^{\text{KO},2}$ and $\widetilde{\mathcal{T}}_h^{\text{KO},1}$.



(a) Stage 1. (b) Stage 2. (c) Stage 3.

Figure 5: Flower-like interface ($h = 1/8$). Illustration of the first three stages of the cell agglomeration procedure.

5.2 Convergence rates

We now study the convergence of the method with respect to h. A static condensation procedure is used to decrease the total number of degrees of freedom. For the circular interface, we report in Table 3 the total number of degrees of freedom (after static condensation) and the ratio of the system size before and after static condensation. This highlights the benefits of performing static condensation. In the results presented hereafter, the linear systems are solved by a conjugate gradient method. The tolerance is fixed to 10^{-14}, and we consider a diagonal preconditioner. Moreover the gradients are reconstructed in the full vector-valued polynomial space $\mathbb{G}^k(T^i;\mathbb{R}^d) = \mathbb{P}^k(T^i;\mathbb{R}^d)$ for $k \in \{0,1,2,3\}$.

	$k=0$		$k=1$		$k=2$		$k=3$	
h^{-1}	dofs	ratio	dofs	ratio	dofs	ratio	dofs	ratio
8	116	2.76	232	2.76	348	2.95	464	3.20
16	484	2.61	968	2.61	1452	2.79	1936	3.01
32	2020	2.57	4040	2.57	6060	2.74	8080	2.97
64	8092	2.53	16184	2.53	24276	2.70	32368	2.91
128	32588	2.52	65176	2.52	97764	2.68	130352	2.89

Table 3: Circular interface. Number of degrees of freedom (dofs) of the condensed stiffness matrix and ratio of dofs between non-condensed and condensed matrices after agglomeration of the mesh (using $n_{\text{int}} = 4$ for the interface discretization).

We first consider the circular interface with the exact solution:

$$u_1(r) := \frac{r^6}{\kappa_1}, \qquad u_2(r) := \frac{r^6}{\kappa_2} + R^6\left(\frac{1}{\kappa_1} - \frac{1}{\kappa_2}\right), \tag{36}$$

with $a = b := 0.5$, $r^2 := (x-0.5)^2 + (y-0.5)^2$ and $R := 1/3$. Note that there are no jumps across the interface ($g_D = 0$ and $g_N = 0$), but the diffusion coefficient can be highly contrasted (i.e., $\kappa_1 \ll \kappa_2$). A similar test case is proposed in [12].

We first test the convergence of the error with respect to the mesh size h with $\kappa_1 = 1$ and $\kappa_2 = 10^4$. The numerical results are reported in Fig. 6. We recover the expected

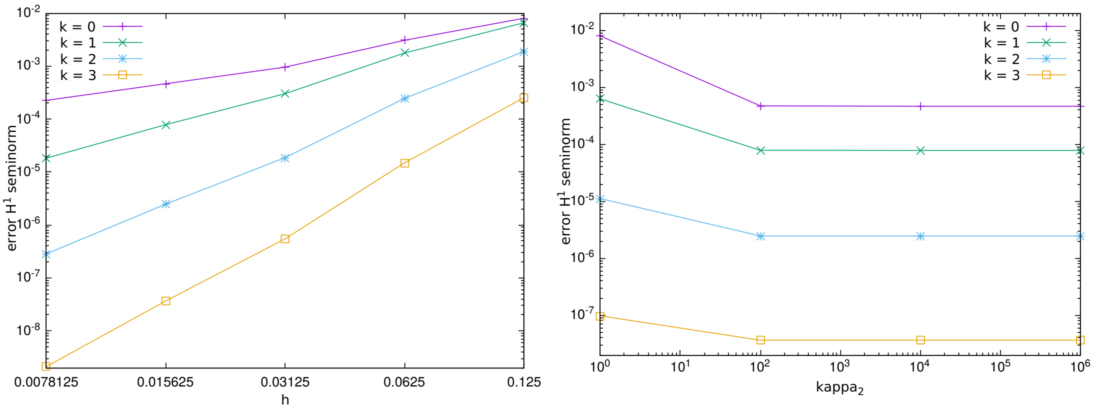


Figure 6: Circular interface. Left: Convergence for $n_{\text{int}} = 10$, $\kappa_1 = 1$ and $\kappa_2 = 10^4$. Right: Robustness with respect to contrast ($\kappa_1 = 1$, $h = 1/64$, $n_{\text{int}} = 10$).

convergence rates for the cell error H^1-seminorm. Moreover the error measured in the L^2-norm also fulfills optimal convergence rates (not shown for brevity).

We then evaluate the robustness of the method with respect to the contrast of the coefficients κ_1 and κ_2. We consider $\kappa_1 = 1$ and $\kappa_2 = 10^\ell$ with $\ell \in \{0, 2, 4, 6\}$. The results are reported in Fig. 6. We observe that the method is robust with respect to the contrast between κ_1 and κ_2 since the cell error H^1-seminorm remains bounded when the contrast increases.

Let us now consider the flower-like interface with parameters $R := 0.31$, $a = b := 0.5$, $n := 4$, $c := 0.04$ and the following discontinuous solution:

$$u_1(x, y) := \cos(\pi x)\cos(\pi y), \qquad u_2(x, y) := \sin(\pi x)\sin(\pi y). \tag{37}$$

For this test case, we set $\kappa_1 = \kappa_2 := 1$. The shape of the interface, the agglomerated mesh (for $h = 1/16$), and the cell component of the numerical solution at the quadrature nodes are presented in Fig. 7.

We test the convergence of the cell error in the H^1-seminorm for the polynomial orders $k \in \{0, 1, 2, 3\}$. Owing to the lack of mesh resolution, we do not consider the case with $h = 1/8$. The results are reported in the left panel of Fig. 8 for $n_{\text{int}} = 10$. We observe that we recover the expected optimal convergence rates. The value of n_{int} is an important parameter when using high-order polynomials since a low order representation of the interface can slow the convergence of the method when jumps are considered [28]. To illustrate this point, we compare different values of n_{int} in the right panel of Fig. 8 for $k = 3$. We see that for n_{int} small, we do not have optimal convergence of the error. Indeed some geometric error is introduced by the linear piecewise representation of the interface. This error is lowered when choosing n_{int} larger. A more efficient technique could be the use of a higher order representation of the immersed interface like in [32].

5.3 Square interface problem: condition number

As a final numerical experiment, we evaluate the condition number $\text{cond}(K) := \frac{\lambda_{\max}}{\lambda_{\min}}$ of the stiffness matrix K, where $\lambda_{\max}$ and $\lambda_{\min}$ are respectively the maximum and minimum

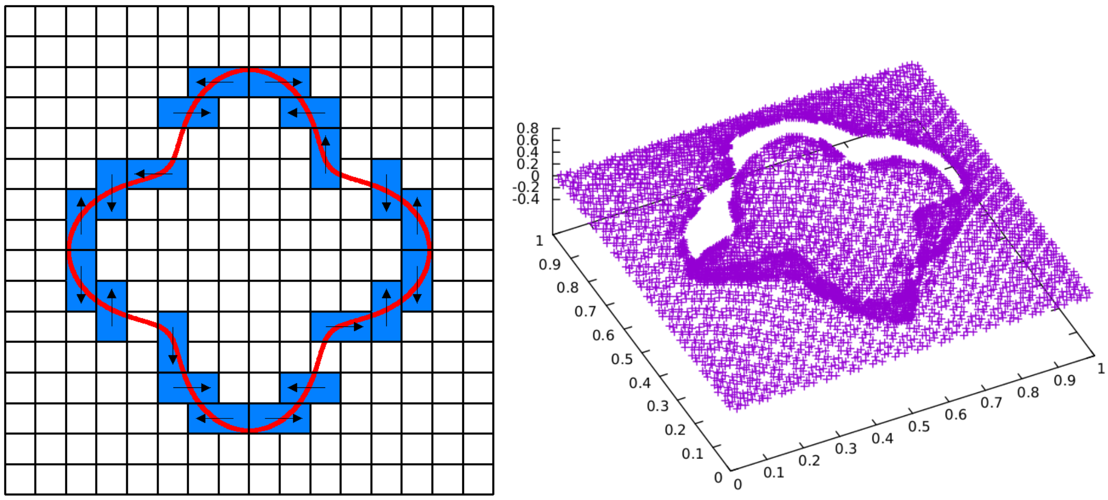


Figure 7: Flower-like interface. Left: mesh and shape of the interface ($h = 1/16$). Right: Cell component of the discrete solution at the quadrature nodes ($h = 1/16$, $k = 0$).

eigenvalues of K which are computed with the use of the Spectra library[2]. The stiffness matrix considered is the one that we use to solve the linear system, i.e. we use static condensation. We consider $\kappa_1 = \kappa_2 = 1$.

We consider the square interface defined by the level-set function $\Phi_S(x, y) := \max(x - 0.5, y - 0.5) - (0.25 + a)$ with parameter $a \in (0, 0.1)$, see Fig. 9. When $a \to 0$, this configuration maximizes the number of badly cut cells (and the number of small subfaces). We consider a fixed unfitted 64×64 square mesh. Even if we agglomerate some mesh cells, there are small cut faces since there is no agglomeration of faces (the faces are unchanged during the agglomeration procedure, except for the ones that are withdrawn). For instance, the large square around the corner is an octagon (and not a square) since it has eight faces (see Fig. 9). In a similar way, the agglomerated cells along the interface are hexagons, not rectangles. In this configuration, for a small enough, a corresponds to the length of the smallest subface, and our goal is to explore the behavior of the condition number of the stiffness matrix when $a \to 0$.

The results are reported in Fig. 10. The right panel shows that the condition number does not diverge when $a \to 0$, even if some faces are cut with very tiny subsets. We think that the scaling of the face basis functions associated with a subcell F^i, $i \in \{1, 2\}$, by $|F^i|^{-1/2}$ tames the potential ill-conditioning of the matrix due to the small-cut faces. The left panel in Fig. 10 presents the evolution of the condition number as a function of h. When $h \to 0$, the condition number grows like $O(h^{-2})$ which is the usual rate for second-order elliptic differential operators. Moreover, the condition number remains reasonable when k increases. This result is different from the one reported for cutFEM in [37], where strong growth of the condition number with increasing polynomial degree was observed.

[2] see `https://spectralib.org/`

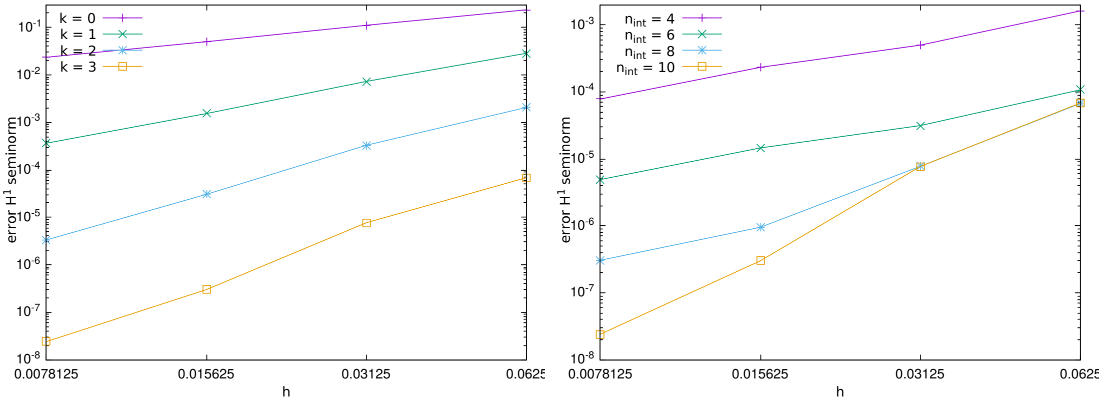


Figure 8: Flower-like interface. Left: Convergence for different values of k ($n_{\text{int}} = 10$). Right: Convergence for different values of n_{int} ($k = 3$).

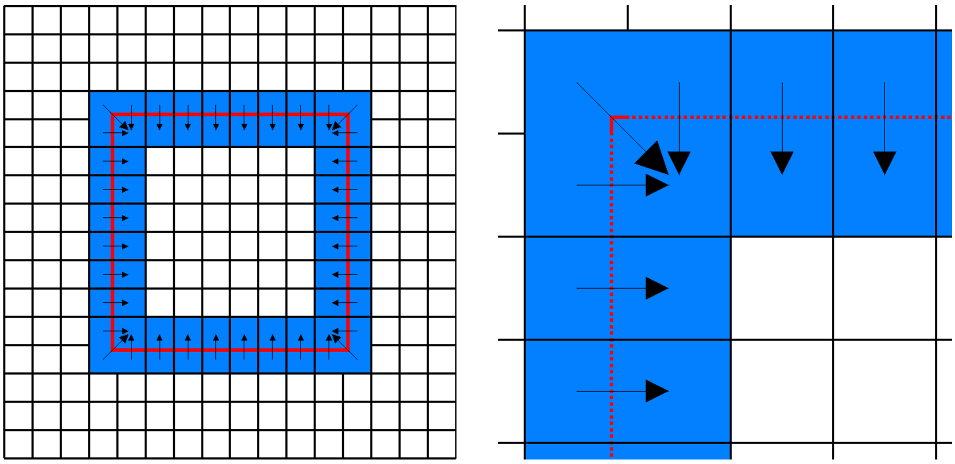


Figure 9: Square interface ($h = 1/16$). Left: full mesh. Right: zoom around the left top corner.

References

[1] M. Abbas, A. Ern, and N. Pignet. Hybrid High-Order methods for finite deformations of hyperelastic materials. Comput. Mech., 62(4):909–928, 2018.

[2] S. Badia, F. Verdugo, and A. F. Martín. The aggregated unfitted finite element method for elliptic problems. Comput. Methods Appl. Mech. Engrg., 336:533–553, 2018.

[3] J. W. Barrett and C. M. Elliott. A finite-element method for solving elliptic equations with Neumann data on a curved boundary using unfitted meshes. IMA J. Numer. Anal., 4(3):309–325, 1984.

[4] J. W. Barrett and C. M. Elliott. Fitted and unfitted finite-element methods for elliptic equations with smooth interfaces. IMA J. Numer. Anal., 7(3):283–300, 1987.

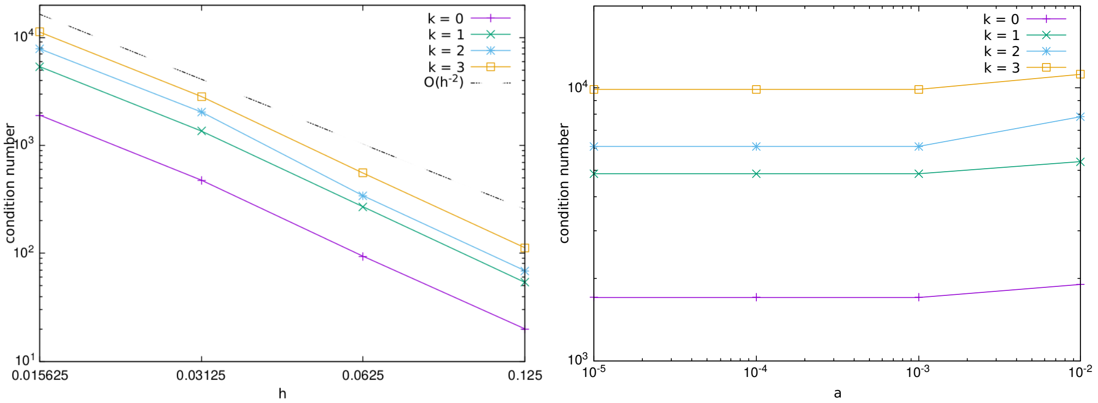


Figure 10: Square interface. Condition number of the stiffness matrix. Left: as a function of h ($a = 0.01$). Right: as a function of the cut parameter a ($h = 1/64$).

[5] L. Botti and D. A. Di Pietro. Assessment of hybrid high-order methods on curved meshes and comparison with discontinuous Galerkin methods. J. Comput. Phys., 370:58–84, 2018.

[6] M. Botti, D. A. Di Pietro, and P. Sochala. A Hybrid High-Order method for nonlinear elasticity. SIAM J. Numer. Anal., 55(6):2687–2717, 2017.

[7] E. Burman. Ghost penalty. C. R. Math. Acad. Sci. Paris, 348(21-22):1217–1220, 2010.

[8] E. Burman and A. Ern. An unfitted hybrid high-order method for elliptic interface problems. SIAM J. Numer. Anal., 56(3):1525–1546, 2018.

[9] E. Burman and P. Hansbo. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. Appl. Numer. Math., 62(4):328–341, 2012.

[10] E. Burman and P. Zunino. A domain decomposition method based on weighted interior penalties for advection-diffusion-reaction problems. SIAM J. Numer. Anal., 44(4):1612–1638, 2006.

[11] E. Burman, S. Claus, P. Hansbo, M. G. Larson, and A. Massing. CutFEM: discretizing geometry and partial differential equations. Internat. J. Numer. Methods Engrg., 104 (7):472–501, 2015.

[12] E. Burman, J. Guzmán, M. A. Sánchez, and M. Sarkis. Robust flux error estimation of an unfitted Nitsche method for high-contrast interface problems. IMA J. Numer. Anal., 38(2):646–668, 2018.

[13] A. Cangiani, E. H. Georgoulis, and Y. A. Sabawi. Adaptive discontinuous Galerkin methods for elliptic interface problems. Math. Comp., 87(314):2675–2707, 2018.

[14] A. Cangiani, Z. Dong, and E. H. Georgoulis. *hp*-version discontinuous Galerkin methods on essentially arbitrarily-shaped elements. arXiv preprint 1906.01715, 2020.

[15] M. Cicuttin, D. A. Di Pietro, and A. Ern. Implementation of discontinuous skeletal methods on arbitrary-dimensional, polytopal meshes using generic programming. J. Comput. Appl. Math., 344:852–874, 2018.

[16] B. Cockburn, W. Qiu, and M. Solano. A priori error analysis for HDG methods using extensions from subdomains to achieve boundary conformity. Math. Comp., 83(286): 665–699, 2014.

[17] B. Cockburn, D. A. Di Pietro, and A. Ern. Bridging the Hybrid High-Order and Hybridizable Discontinuous Galerkin methods. ESAIM: Math. Model Numer. Anal. (M2AN), 50(3):635–650, 2016.

[18] F. de Prenter, C. Lehrenfeld, and A. Massing. A note on the stability parameter in Nitsche's method for unfitted boundary value problems. Comput. Math. Appl., 75 (12):4322–4336, 2018.

[19] D. A. Di Pietro and A. Ern. Mathematical Aspects of Discontinuous Galerkin Methods, volume 69 of Mathématiques & Applications. Springer-Verlag, Berlin, 2012.

[20] 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., 283:1–21, 2015.

[21] 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.

[22] D. A. Dunavant. High degree efficient symmetrical Gaussian quadrature rules for the triangle. Internat. J. Numer. Methods Engrg., 21(6):1129–1148, 1985.

[23] A. Ern, A. F. Stephansen, and P. Zunino. A discontinuous Galerkin method with weighted averages for advection-diffusion equations with locally small and anisotropic diffusivity. IMA J. Numer. Anal., 29(2):235–256, 2009.

[24] C. Gürkan, E. Sala-Lardies, M. Kronbichler, and S. Fernández-Méndez. Extended hybridizable discontinous Galerkin (X-HDG) for void problems. J. Sci. Comput., 66 (3):1313–1333, 2016.

[25] C. Gürkan, M. Kronbichler, and S. Fernández-Méndez. eXtended hybridizable discontinuous Galerkin with Heaviside enrichment for heat bimaterial problems. J. Sci. Comput., 72(2):542–567, 2017.

[26] A. Hansbo and P. Hansbo. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. Comput. Methods Appl. Mech. Engrg., 191 (47-48):5537–5552, 2002.

[27] P. Huang, H. Wu, and Y. Xiao. An unfitted interface penalty finite element method for elliptic interface problems. Comput. Methods Appl. Mech. Engrg., 323:439–460, 2017.

[28] L. N. T. Huynh, N. C. Nguyen, J. Peraire, and B. C. Khoo. A high-order hybridizable discontinuous Galerkin method for elliptic interface problems. Internat. J. Numer. Methods Engrg., 93(2):183–200, 2013.

[29] A. Johansson and M. G. Larson. A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary. Numer. Math., 123(4):607–628, 2013.

[30] C. Lehrenfeld. Hybrid Discontinuous Galerkin methods for solving incompressible flow problems. PhD thesis, Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen, 2010.

[31] C. Lehrenfeld. Removing the stabilization parameter in fitted and unfitted symmetric Nitsche formulations. 2016. arXiv:1603.00617.

[32] C. Lehrenfeld and A. Reusken. Analysis of a high-order unfitted finite element method for elliptic interface problems. IMA J. Numer. Anal., 38(3):1351–1387, 2018.

[33] C. Lehrenfeld and J. Schöberl. High order exactly divergence-free hybrid discontinuous Galerkin methods for unsteady incompressible flows. Comput. Methods Appl. Mech. Engrg., 307:339–361, 2016.

[34] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. Abh. Math. Sem. Univ. Hamburg, 36:9–15, 1971. Collection of articles dedicated to Lothar Collatz on his sixtieth birthday.

[35] W. Qiu, M. Solano, and P. Vega. A high order HDG method for curved-interface problems via approximations from straight triangulations. J. Sci. Comput., 69(3): 1384–1407, 2016.

[36] W. E. H. Sollie, O. Bokhove, and J. J. W. van der Vegt. Space-time discontinuous Galerkin finite element method for two-fluid flows. J. Comput. Phys., 230(3):789–817, 2011.

[37] S. Sticko, G. Ludvigsson, and G. Kreiss. High order cut finite elements for the elastic wave equation. https://arxiv.org/pdf/1804.00332.pdf, 2019.