UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Extended hybridizable discontinuous Galerkin method

by

Ceren Gürkan

ADVERTIMENT La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del repositori institucional UPCommons (http://upcommons.upc.edu/tesis) i el repositori cooperatiu TDX (http://www.tdx.cat/) ha estat autoritzada pels titulars dels drets de propietat intel·lectual **únicament per a usos privats** emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei UPCommons o TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a UPCommons (*framing*). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del repositorio institucional UPCommons (http://upcommons.upc.edu/tesis) y el repositorio cooperativo TDR (http://www.tdx.cat/?locale-attribute=es) ha sido autorizada por los titulares de los derechos de propiedad intelectual **únicamente para usos privados enmarcados** en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio UPCommons No se autoriza la presentación de su contenido en una ventana o marco ajeno a UPCommons (*framing)*. Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the institutional repository UPCommons (http://upcommons.upc.edu/tesis) and the cooperative repository TDX (http://www.tdx.cat/?locale-attribute=en) has been authorized by the titular of the intellectual property rights **only for private uses** placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading nor availability from a site foreign to the UPCommons service. Introducing its content in a window or frame foreign to the UPCommons service is not authorized (framing). These rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

Universitat Politècnica de Catalunya
Erasmus Mundus PhD program in
SIMULATION IN ENGINEERING AND
ENTREPRENEURSHIP DEVELOPMENT (SEED)

Departament d'Enginyeria Civil i Ambiental

EXTENDED HYBRIDIZABLE DISCONTINUOUS GALERKIN METHOD

by

CEREN GÜRKAN

Doctoral Thesis
Advisor: Assoc. Prof. Sonia Fernández-Méndez
Co-Advisor: Dr. Martin Kronbichler

Barcelona, March 2017

Her şeyimi borçlu olduğum anne ve babama...

To my mum and dad to whom I owe everything...

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Sonia Fernández-Méndez, for her endless support, patience, guidance and encouragement at all stages of this work, more importantly, thank you Sonia, too very much, for being just in the way that you are.

Many thanks to all my colleagues, from all around the world. I am afraid to write down the names, to not to miss any; but thank you all, for showing me that does not matter our origins, we can still laugh together and remember; the world spins for the sake of you good people. Thanks, for making me believe yet again that *a different, a better world is possible*.

Thanks to my friends, whom understands me the most, the ones I speak the same language. Thanks all of you for just being there and make me feel confident that you all will be there when I need you.

Very special thanks to Dr. Erhan Karaesmen for making my dreams came true.

Finally, endless thanks to David Villamizar Duque, for... simply everything...

Contents

List of Figures

Chapter 1

Introduction

This work proposes a new numerical technique: the eXtended Hybridizable Discontinuous Galerkin (X-HDG) Method, to solve efficiently problems including moving boundaries and interfaces. It aims to outperform available methods and improve the results by inheriting favored properties of HDG together with an explicit interface definition.

Problems where distinct materials or fluids are involved are named *interface problems* and they are widely encountered in both academical and practical applications. From crack modeling to the Stefan problem or any multi-field implementation, scientists have been searching for better ways to tackle interface problems for decades now. X-HDG combines the Hybridizable Discontinuous Galerkin (HDG) Method with an eXtended Finite Element (X-FEM) philosophy together with a level set description of the interface, to form an *hp* convergent, high order unfitted numerical method. HDG outperforms other DG methods for problems involving self-adjoint operators, due to its hybridization and superconvergence properties. The hybridization process drastically reduces the number of degrees of freedom in the discrete problem, similarly to static condensation in the context of high-order Continuous Galerkin (CG), see for instance Giorgiani et al. [2013b]. On other hand, HDG is based on a mixed formulation that, differently to CG or other Discontinuous Galerkin (DG) methods, is stable even when all variables (primal unknowns and derivatives) are approximated with polynomials of the same degree k. As a result, convergence of order $k+1$ in the $\mathcal{L}_2$ norm is proved not only for the primal unknown, but also for its derivatives. Therefore, a simple element-by-element postprocess of the derivatives leads to a su-

perconvergent approximation of the primal variables, with convergence of order $k+2$ in the $\mathcal{L}_2$ norm. X-HDG inherits these favored properties of HDG in front of CG and DG methods; moreover, thanks to the level set description of interfaces, costly re meshing is avoided dealing with moving interfaces. This work demonstrates that X-HDG keeps the optimal and superconvergence of HDG with no need of mesh fitting to the interface.

In the next section, a historical overview regarding the methods used to solve interface problems is presented. The main objectives and contributions of this work are listed in Section 1.2 and some preliminaries are added in Section 1.3 to ease the understanding of the rest of the document.

1.1 Interface Problems, a historical overview

Today, approximately 2100 years after the day that Archimedes run naked through the streets of Syracuse, we are already using *numerical methods* to analyze any natural phenomenon with computer power moreover this relatively new discipline called *computational mechanics* is advancing at a very high pace.

In computational mechanics, almost any phenomena can be analyzed by following some "simple" steps. First, a mathematical model—which usually is a *partial differential equation* (PDE)—needs to be derived. Then, *discretization* converts the PDE into a system of algebraic equations. Different strategies used in discretization process are studied in the field of *numerical methods*. Finally, with the aid of computational tools the discretized equations are solved and the results are verified. Next we will see how the interface problems have been treated through time using different numerical methods.

1.1.1 Interface Problems with Continuous Methods

Let us start this overview with the most commonly used Continuous Galerkin Method (CGM): the Finite Element Method (FEM). Early history of FEM starts with Turner et al. [1956] who was the pioneer engineer to invent the method —which by the time was called "Direct Stiffness Method"— to discretize the delta wing working at Boeing over the period 1950–1962. Turner's summer school intern Clough [1960] was the first who called the method as "Finite Element Method" (FEM). After that, in late 60s and early 70s, the FE Method was generalized and used for the numerical modeling of

physical systems in a wide variety of engineering disciplines such as electromagnetism, heat transfer and fluid dynamics.

Being among the early contributors of FEM, Ergatoudis et al. [1968] introduced the novel *isoparametric element* concept while Zienkiewicz and Cheung [1965] extended the method to field problems. After that, countless contributions have been made in the context of the Finite Element Method; for a complete overview, see e.g Zienkiewicz et al. [2005].

Soon after, Babuška [1970] turned the attention to the elliptic *interface problem* with discontinuous coefficients and made a first attempt to solve it using an *unfitted method*. The finite element approximation was based on a mesh that is independent of the interface. The idea was to penalize the jump at the interface and approximating this penalized problem. Even though only $\mathcal{O}(h^{0.5})$ convergence rate in $\mathcal{H}^1$ norm for all $k \geq 2$ was achieved, it was an important attempt to open the path to handle interface problems with continuous methods.

Later, various methods were proposed to treat interface problems over unfitted meshes, among them penalty methods, immersed boundary methods or the fictitious domain methods can be counted. Unfitted methods are often combined with the popular level set method (Osher and Sethian [1988]) which allows implicit interface definition by means of the zero level set of a function of one higher dimension.

Barrett and Elliott [1987] further studied the penalty method proposed by Babuška [1970], and for a properly chosen penalty parameter, optimal convergence in $\mathcal{H}^1$ norm was proved. Influenced by Peskin and Mcqueen [1989], the immersed interface method proposed by Leveque and Li [1994] implemented a finite difference type discretization to cope with elliptic interface problems. The method was also applied to other interface problems such as Stokes flow see Leveque and Li [1997], or the one-dimensional moving interface problem see Li [1997] and Hele-Shaw flow by Hou et al. [1997]. However, the final linear systems obtained with the immersed interface method were non-symmetric and indefinite. Moreover, the method was proved to converge only up to second order. An important contribution was proposed by Burman and Hansbo [2012] with the fictitious domain method. Level-set defined interface cuts through the computational mesh and optimal a priori error estimates in $\mathcal{H}^1$ and $\mathcal{L}_2$ norms are proved for linear elements.

Among unfitted methods in the FE framework, the eXtended Finite Element Method (X-FEM) is surely the one that is most widely implemented. The method

was proposed by Moës et al. [1999] to model the crack growth without remeshing. An enriched approximation is utilized near the crack to incorporate the discontinuity through the partition of unity method of Melenk and Babuška [1996]. The results obtained were pretty convincing and X-FEM was soon widely implemented.

Sukumar et al. [2001] used X-FEM to solve void and inclusion problems, where optimal convergence in the energy norm was reached only for void problems using linear elements. On the other hand, no optimal convergence was reached for inclusion problems, not even with linear elements. Hansbo and Hansbo [2002] proposed an interface position free, stable FE method for second order elliptic problems and optimum order of convergence was demonstrated in the $\mathcal{L}_2$ norm for linear elements. Further in Becker et al. [2009] the same strategy was applied to the Stokes interface problem and yet again optimum convergence was shown using linear elements. Annavarapu et al. [2012] extended Hansbo's work for special cases and proposed a robust Nitsche formulation for interface problems. Furthermore Ji et al. [2002] successfully applied X-FEM to a phase change problem and a good match with the analytical solution was observed. Sala Lardies et al. [2012] on the other hand, implemented X-FEM using high order elements and enrichment functions for inclusion and bimaterial problems where optimum convergence is demonstrated in the $\mathcal{L}_2$ norm. For a complete overview on X-FEM and its applications see the excellent work by Fries and Belytschko [2010].

In conclusion, to the author's best knowledge and as they are cited above, various numerical methods in the Continuous Galerkin framework successfully tackle interface problems; both high and low order and over fitted and unfitted meshes. How the interface problems are treated using *discontinuous methods* is presented next.

1.1.2 Interface Problems with Discontinuous Methods

Reed and Hill [1973] introduced the "Discontinuous Galerkin" (DG) method to the community where it was proved that a discontinuous flux definition across elemental boundaries leads to better results than using a continuous flux definition for a neutron transport problem. The function space used in the DG method was the same for the continuous method, but with relaxed continuity at interelement boundaries. The basis functions in DG can be chosen so that either the field variable, its derivatives, or generally both, are considered discontinuous across the element boundaries, while the overall continuity in the domain is maintained. DG inherits the advantages

of both FE and Finite Difference (FD) methods. After its advantages against continuous methods had been revealed, countless contributions were made and various discontinuous Galerkin methods were derived through the time. Arnold et al. [2001] compared different discontinuous Galerkin methods proposed over time. For curious reader this work is highly suggested.

Guyomarch et al. [2009] studied a class of discontinuous Galerkin methods for elliptic interface problems which was shown to be optimally convergent in the $\mathcal{L}_2$ norm with fitted meshes.

Fidkowski and Darmofal [2007] introduced DG with the immersed boundary method (IBM) Peskin [1972], analyzing steady compressible flow with a computational cost comparison against the fitted boundary case, when the error tolerance is fixed. DG, together with IBM, proved to achieve a desired accuracy with fewer degrees of freedom. Later, Bastian and Engwer [2009] considered high order unfitted DG to solve Poisson type porous flow problem where optimum convergence rates were achieved for high order of approximation. Massjung [2012] proposed an unfitted DG method for the second order elliptic problem, a generalization of the idea proposed by Hansbo and Hansbo [2002], where optimal h and suboptimal p convergence was demonstrated in the $\mathcal{L}_2$ norm. More recently Wang and Chen [2014] implemented an unfitted DG where optimal error estimates in the energy norm for the primal unknown and for its derivative were obtained, however, only with linear elements.

Even though DG methods have several advantages compared to continuous methods —such as intrinsic stabilization, suitability for code vectorization, parallel computation and adaptivity—, and different proposals successfully handled interface problems even using high order approximations, it is well known that for a fixed mesh and fixed approximation degree, the total number of degrees of freedom in the final system obtained using DG methods is higher than with CG methods.

However, this major drawback is overcome with the Hybridized Discontinuous Galerkin (HDG) method, where in next section a closer look to the method is presented.

1.1.3 Hybridized Discontinuous Galerkin Method and Its Application to Interface Problems

The Hybridized Discontinuous Galerkin (HDG) Method was proposed by Cockburn et al. [2009] where the authors achieved to *hybridize* DG method, with an idea similar

to static condensation in CGM see Giorgiani et al. [2013b]. In the resulting linear system, the number of degrees of freedom is reduced dramatically, while all the advantageous properties of DG are inherited see Peraire and Persson [2008], Montlaur et al. [2008]. Moreover with HDG, convergence of order $k+1$ in the $\mathcal{L}_2$ norm is proved not only for the primal unknown, but also for its derivative see Cockburn et al. [2008], Cockburn et al. [2012] leading to a super convergent —order $k+2$— post processed solution, where k is the polynomial degree of approximation. HDG became very popular in the community very fast and was applied in all kinds of problems, for its application to the Stokes and Navier-Stokes equations see for example Cockburn et al. [2011], Nguyen et al. [2010], Nguyen et al. [2011] or for an efficiency study against CG methods see Kirby et al. [2011], Giorgiani et al. [2013b], Huerta et al. [2013].

Even tough HDG is very popular in the CFD community there is not much of work done for the solution of interface problems with HDG. In Wang and Khoo [2013] HDG was applied to solve Stokes interface problem with a fitted mesh, requiring a proper interface description using computational mesh. Similarly in Huynh et al. [2013] HDG was used to solve a Poisson interface problem with a fitted mesh. In Dong et al. [2016] the authors proposed an HDG method for unfitted meshes, based on the use of ansatz functions that are defined in the whole computational domain and represent the discontinuities over the interface. However, convergence was proved up to second order only, and numerical examples show that higher order convergence may not be reached in general. In Cockburn et al. [2014] and Cockburn and Solano [2014] a methodology based on extrapolation of the solution to the boundary from nodal values of the mesh using unfitted meshes was proposed. However, restrictions on the distance from the computational mesh to the boundary to ensure optimal convergence limit the applicability of the method.

This work proposes an alternative strategy for the HDG solution of interface problems, based on an eXtended Finite Element (X-FEM) philosophy, the Extended Hybridizable Discontinuous Galerkin (X-HDG) method. A level set definition is used for the geometrical representation of the interface leading to a single computational mesh. As the interface cuts through some of the elements and faces, a proper numerical integration at those elements and faces is implemented see Sala Lardies et al. [2012] and section 1.3.3. In case of material interfaces, a proper enrichment is introduced. In case of moving interfaces, constant re-meshing is avoided. Overall, X-HDG

aims to inherit all the desired properties of HDG while removing the fitted mesh restriction to treat interface problems in a better, alternative way.

1.2 Objectives and document structure

This work proposes a new numerical technique to treat interface problems: the eXtended Hybridizable Discontinuous Galerkin (X-HDG) Method. The aim is to outperform available methods by inheriting favored properties of HDG such as its hybrid nature, element by element fashion, stabilization or suitability to code vectorization but most importantly its optimum and super convergence. In the context of second order elliptic and Stokes interfaces; as well as with moving interfaces, the goal is to demonstrate a high order, unfitted, optimal and super convergent method where high order accurate meshing, or re-meshing costs are avoided. By inheriting the convergence properties of HDG, X-HDG as well aims to demonstrate its strong point, a $k+1$ convergent solution gradient, leading to improved results solving moving interface problems compared to available methods.

The objectives being set, next, the structure of this document is presented.

- **X-HDG for Second Order Elliptic Void Problems:** The X-HDG method is derived and implemented to solve the steady-state Laplace equation on a domain where the interface separates a single material from the void. The modified weak form for cut elements is presented in Sections 2.1.2 and 2.2 where a Dirichlet or Neumann boundary condition is imposed on a complex boundary that is not captured by the computational mesh. The accuracy and the convergence of X-HDG is tested over examples with manufactured solutions in section 2.3. The results are compared with fitted mesh and the standard HDG method. The work presented in Chapter 2 can also be found in Gürkan et al. [2016].

- **X-HDG for Second Order Elliptic Bimaterial Problems:** The X-HDG for second order elliptic bimaterial problems is studied in Chapter 3. The modified local problem for cut elements and faces is presented in section 3.1.2. For the discretization, enriched spaces are used for both elemental and facial variables to capture the discontinuities within the element. The formulation is verified through numerical examples in section 3.2. An example including multi

materials is studied in 3.2.5 where the results are compared with standard X-FEM.

- **X-HDG for Stokes Problem:** In Chapters 4 and 5, the X-HDG method is derived and implemented to solve Stokes interface problem. Modified local problem is derived for cut elements and faces in sections 4.1, 4.3 and 5.1 for Neumann voids, Dirichlet voids and for bimaterial interfaces, respectively. The applicability of X-HDG is proved through several examples and documented in 4.5 and 5.3.

- **X-HDG for moving interfaces:** X-HDG for moving interfaces is studied in Chapter 6. A transient Laplace problem is considered, where the time dependent term is discretized using the backward Euler method. The local problem is presented in Sections 6.2 and 6.3 for standard and cut elements, respectively. A constant speed collapsing circle example together with two phase Stefan problem are analyzed in section 6.5, numerical examples.

1.3 Preliminaries

For a better understanding of upcoming chapters, here, the standard HDG discretization is recalled for second order elliptic equations in Section 1.3.1 and Stokes equations in Section 1.3.2. This section is relevant since, in the X-HDG setting, the elements in the domain which are not cut by the interface are discretized following standard HDG setting i.e., as it is explained next.

1.3.1 Hybridizable Discontinuous Galerkin Method for Second Order Elliptic Equation

Let $\Omega \subset \mathbb{R}^d$ be a domain with an exterior boundary $\partial\Omega$. The following problem is considered,

$$\begin{aligned} -\boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} u) &= f \quad \text{in } \Omega \\ u &= u_D \quad \text{on } \Gamma_D \\ -\nu \boldsymbol{\nabla} u \cdot \boldsymbol{n} &= g_N \quad \text{on } \Gamma_N, \end{aligned} \tag{1.1}$$

where u is the solution, ν is a material coefficient, f is a given source term, u_D are prescribed values on the Dirichlet boundary and g is the prescribed flux on the Neumann boundary with $\partial\Omega = \Gamma_D \cup \Gamma_N$.

The domain Ω is now assumed to be partitioned by a finite element mesh with $\mathtt{n_{el}}$ disjoint elements K_i, such that

$$\overline{\Omega} = \bigcup_{i=1}^{\mathtt{n_{el}}} \overline{K}_i, \quad K_i \cap K_j = \emptyset \text{ for } i \neq j. \tag{1.2}$$

The union of all $\mathtt{n_{fc}}$ faces Γ_i (sides for 2D) is denoted as

$$\Gamma := \bigcup_{i=1}^{\mathtt{n_{el}}} \partial K_i = \bigcup_{f=1}^{\mathtt{n_{fc}}} \Gamma_f. \tag{1.3}$$

The discontinuous setting induces a new problem equivalent to (1.1), with some element-by-element equations and some global ones. The local element-by-element problems correspond to the statement of the PDE in (1.1) with essential boundary conditions at each element K_i, that is,

$$\begin{aligned} \boldsymbol{\nabla} \cdot \boldsymbol{q} &= f \quad \text{in } K_i \\ \boldsymbol{q} + \nu \boldsymbol{\nabla} u &= 0 \quad \text{in } K_i \\ u &= \widehat{u} \quad \text{on } \partial K_i. \end{aligned} \tag{1.4}$$

Two new variables are introduced: $\boldsymbol{q}$ corresponding to the flux of u, allowing the splitting of the PDE into two first order PDEs, and $\widehat{u}$ corresponding to the trace of u at the mesh faces.

Note that, given the trace $\widehat{u}$, which is single valued at the mesh skeleton Γ, the local problem (1.4) can be solved on each element to determine the solution u and the flux $\boldsymbol{q}$. Thus, the problem now reduces to determine the trace $\widehat{u}$ by solving the so-called *conservativity condition* (also known as *global equations*), that is, the continuity of the flux across element boundaries

$$[\![\boldsymbol{q} \cdot \boldsymbol{n}]\!] = 0 \quad \text{on } \Gamma \backslash\, \partial\Omega, \tag{1.5}$$

together with imposing the exterior boundary condition, as stated in (1.1)

$$\widehat{u} = \mathbb{P}_2(u_D) \quad \text{on } \Gamma_D \quad \text{and} \quad -\nu \boldsymbol{\nabla} u \cdot \boldsymbol{n} = g_N \quad \text{on } \Gamma_N \tag{1.6}$$

where $\mathbb{P}_2(u_D)$ is the $\mathcal{L}_2$ projection of the data u_D on the faces, i.e., a least-squares fitting. The *jump* $[\![\cdot]\!]$ operator is defined at an interior face Γ_f as

$$[\![\circledcirc]\!] = \circledcirc_{L(f)} + \circledcirc_{R(f)} \quad \text{on } \Gamma_f, \tag{1.7}$$

where $R(f)$ and $L(f)$ are the indices indicating the left and right elements sharing the face, that is, $\Gamma_f = \overline{K_{L(f)}} \cap \overline{K_{R(f)}}$, and the subindex $\circledcirc_i$ denotes the value of function $\circledcirc$ from element K_i. In particular, $[\![\boldsymbol{q} \cdot \boldsymbol{n}]\!] = \boldsymbol{q}_{L(f)} \cdot \boldsymbol{n}_{L(f)} + \boldsymbol{q}_{R(f)} \cdot \boldsymbol{n}_{R(f)} = (\boldsymbol{q}_{L(f)} - \boldsymbol{q}_{R(f)}) \cdot \boldsymbol{n}_{L(f)}$.

It is important noting that the continuity of the solution u across Γ is imposed by the Dirichlet boundary condition in the local problems (1.4) and the fact that $\widehat{u}$ is single valued on Γ.

The discretization of the conservativity condition (1.5) and the local problem (1.4), with the boundary condition (1.6), leads to the HDG formulation. The following discrete spaces for elemental variables, u and $\boldsymbol{q}$, and for the trace variable, $\widehat{u}$, are considered

$$\begin{aligned} \mathcal{V}^h &:= \left\{ v \in \mathcal{L}_2(\Omega) \ : v|_{K_i} \in \mathcal{P}_k(K_i) \ \text{ for } \ i = 1, \ldots, \mathtt{n_{el}} \right\} \\ \Lambda^h &:= \left\{ \hat{v} \in \mathcal{L}_2(\Gamma) \ : \hat{v}|_{\Gamma_f} \in \mathcal{P}_k(\Gamma_f) \ \text{ for } \ f = 1, \ldots, \mathtt{n_{fc}} \right\}, \end{aligned} \tag{1.8}$$

where $\mathcal{P}_k$ denotes the space of polynomials of degree less or equal to k.

Next the discretizations of the local and global problems are presented, respectively. To simplify the presentation, in an abuse of notation, the same notation is used for the numerical approximation, belonging to the finite dimensional spaces (1.8), and the exact solution, that is u, $\boldsymbol{q}$ and $\widehat{u}$.

The discretization of (1.4), leads to

$$\begin{aligned} \int_{K_i} v \boldsymbol{\nabla} \cdot \boldsymbol{q} \, dV + \int_{\partial K_i} \nu\tau v(u - \widehat{u}) \, dS &= \int_{K_i} v f \, dV \qquad \forall v \in \mathcal{P}_k(K_i) \\ \int_{K_i} \boldsymbol{q} \cdot \boldsymbol{w} \, dV - \int_{K_i} \nu u \boldsymbol{\nabla} \cdot \boldsymbol{w} \, dV + \int_{\partial K_i} \nu \widehat{u} \boldsymbol{w} \cdot \boldsymbol{n} \, dS &= 0 \quad \forall \boldsymbol{w} \in [\mathcal{P}_k(K_i)]^d \end{aligned} . \tag{1.9}$$

The first equation in (1.9) can be derived from the first equation in (1.4) by applying integration by parts, replacing the flux by the numerical flux

$$\widehat{\boldsymbol{q}} := \boldsymbol{q} + \nu\tau(u - \widehat{u})\boldsymbol{n}, \tag{1.10}$$

and undoing the integration by parts. The second equation is obtained from the weak form of the second equation in (1.4), applying integration by parts and replacing the boundary condition $u = \widehat{u}$ on the element boundary.

Remark 1. The parameter τ, appearing in the definition of the numerical flux (1.10), is a non-negative stabilization parameter usually taken of order $\mathcal{O}(1)$. For each element, it may be taken as a positive constant on all faces, or positive on one arbitrary

face and zero at the rest (single face). Both options lead to stable and optimally convergent solutions, with superconvergent post-processed solutions. See for instance Giorgiani et al. [2013a], Cockburn et al. [2008] for details on the influence of this parameter on the solution behaviour.

The discretization of the local problem (1.9) leads to a system of equations of the form

$$\begin{cases} \mathbf{A}_{uu}^{K_i}\mathbf{u}^i + \mathbf{A}_{uq}^{K_i}\mathbf{q}^i + \mathbf{A}_{u\widehat{u}}^{K_i}\mathbf{\Lambda}^i = \mathbf{f}_u^{K_i} \\ \mathbf{A}_{qu}^{K_i}\mathbf{u}^i + \mathbf{A}_{qq}^{K_i}\mathbf{q}^i + \mathbf{A}_{q\widehat{u}}^{K_i}\mathbf{\Lambda}^i = \mathbf{0} \end{cases} \tag{1.11}$$

where $\mathbf{u}^i$ and $\mathbf{q}^i$ are the vectors of nodal values of u and $\boldsymbol{q}$ in element K_i, and Λ^i is the vector of nodal values of $\widehat{u}$ on the n faces of the element ($n = 3$ for triangles and $n = 4$ for tetrahedra). That is,

$$\mathbf{\Lambda}^i := \begin{bmatrix} \widehat{\mathbf{u}}^{\mathbf{F}_{i1}} \\ \vdots \\ \widehat{\mathbf{u}}^{\mathbf{F}_{in}} \end{bmatrix}, \tag{1.12}$$

where $\widehat{\mathbf{u}}^f$ denotes the nodal values of $\widehat{u}$ on face Γ_f, and $\mathbf{F}_{ij}$ is the number of the j-th face of element K_i, see an example in Figure 1.1.

Note that the subindices in the $\mathbf{A}$ matrices refer to the space for the weighting function and the test function, respectively.

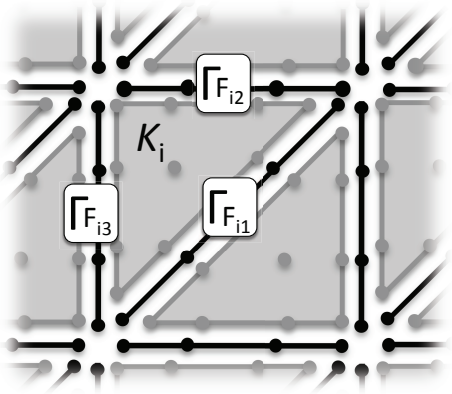


Figure 1.1: Example of a third degree HDG discretization. Nodal approximation at elements (gray nodes) for u and $\boldsymbol{q}$, and nodal approximation at faces (black nodes) for the trace $\widehat{u}$. The three faces for element K_i correspond to faces number F_{i1}, F_{i2} and F_{i3}.

System (1.11) can be solved for $\mathbf{u}^i$ and $\mathbf{q}^i$ in each element, obtaining the so-called *local solver* in the element K_i

$$\mathbf{u}^i = \mathbf{U}^{K_i}\boldsymbol{\Lambda}^i + \mathbf{f}_U^{K_i}, \quad \mathbf{q}^i = \mathbf{Q}^{K_i}\boldsymbol{\Lambda}^i + \mathbf{f}_Q^{K_i}, \tag{1.13}$$

with

$$\begin{bmatrix} \mathbf{U}^{K_i} \\ \mathbf{Q}^{K_i} \end{bmatrix} = -\mathbb{A}^{-1} \begin{bmatrix} \mathbf{A}_{u\widehat{u}}^{K_i} \\ \mathbf{A}_{q\widehat{u}}^{K_i} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{f}_U^{K_i} \\ \mathbf{f}_Q^{K_i} \end{bmatrix} = \mathbb{A}^{-1} \begin{bmatrix} \mathbf{f}_u^{K_i} \\ \mathbf{0} \end{bmatrix} \tag{1.14}$$

and

$$\mathbb{A} = \begin{bmatrix} \mathbf{A}_{uu}^{K_i} & \mathbf{A}_{uq}^{K_i} \\ \mathbf{A}_{qu}^{K_i} & \mathbf{A}_{qq}^{K_i} \end{bmatrix}$$

That is, for each element, the nodal values of the solution, $\mathbf{u}^i$ and $\mathbf{q}^i$, can be explicitly expressed in terms of the trace on its faces, $\boldsymbol{\Lambda}^i$.

Thus, now the problem is reduced to determine the trace nodal values $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{nfc}}$ on the mesh skeleton Γ. For this purpose the so-called *global problem* is stated, which corresponds to the discretization of the conservativity condition on Γ (1.5).

Replacing $\boldsymbol{q}$ by the numerical flux (1.10), the weak form for the trace variable is: find $\widehat{u} \in \Lambda^h$ such that $\widehat{u} = \mathbb{P}_2(u_D)$ on Γ_D and

$$\int_{\Gamma\setminus\partial\Omega} \widehat{v}[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!]\, dS + 2\int_{\Gamma\setminus\partial\Omega} \widehat{v}\left(\{\nu\tau u\} - \{\nu\tau\}\widehat{u}\right) dS = \int_{\Gamma_N} \widehat{v} g_N$$

for all $\widehat{v} \in \Lambda^h$ such that $\widehat{v} = 0$ on Γ_D. The $\{\cdot\}$ is the mean operator on the faces,

$$\{\circledcirc\} = \frac{1}{2}\left(\circledcirc_{L(f)} + \circledcirc_{R(f)}\right) \quad \text{on } \Gamma_f. \tag{1.15}$$

On the other hand the jump operator $[\![\cdot]\!]$ is defined in (1.7). The discretization of this equation for every face Γ_f leads to an equation of the form

$$\mathbf{A}_{\widehat{u}u}^{f,L}\mathbf{u}^{L(f)} + \mathbf{A}_{\widehat{u}q}^{f,L}\mathbf{q}^{L(f)} \;+\; \mathbf{A}_{\widehat{u}u}^{f,R}\mathbf{u}^{R(f)}(f) + \mathbf{A}_{\widehat{u}q}^{f,R}\mathbf{q}^{R(f)} \;+\; \mathbf{A}_{\widehat{u}\widehat{u}}^{f}\widehat{\mathbf{u}}^f = 0. \tag{1.16}$$

Replacing the local solver (1.13), for the elements $K_{L(f)}$ and $K_{R(f)}$, in (1.16) for every face Γ_f, leads to a system of equations involving only the trace variables $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{nfc}}$.

The implementation of the method involves a loop over elements. For each element, the matrices and vectors for the local solver (1.13) are computed, and the contribution to the equation (1.16) is assembled for each one of the faces of the

element. Once the system is assembled for all elements, and Dirichlet boundary conditions (1.6) are imposed, the system can be solved. Then, given the trace variables $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{n_{fc}}}$, the solution, $\mathbf{u}^i$ and $\boldsymbol{q}^i$, can be computed for each element using (1.13).

Remark 2. A second element-by-element postprocessing can be done to obtain an HDG superconvergent solution in every element K_i as the solution of: find $u^* \in \mathcal{P}_{k+1}(K_i)$ such that

$$\int_{K_i} \nu \boldsymbol{\nabla} u^* \cdot \boldsymbol{\nabla} v \, dV = -\int_{K_i} \boldsymbol{q} \cdot \boldsymbol{\nabla} v \, dV \quad \forall v \in \mathcal{P}_{k+1}(K_i),$$
$$\int_{K_i} u^* \, dV = \int_{K_i} u \, dV$$

for all $v \in \mathcal{P}_{k+1}(K_i)$. The solution of this element-by-element computation, u^*, converges with order $k+2$ in the $\mathcal{L}_2$ norm. See Cockburn et al. [2008, 2012] for details.

Remark 3. The Dirichlet boundary conditions (1.6) can be implemented by setting the nodal values for $\widehat{u}$ on the faces on the Dirichlet boundary by interpolation. However, an $\mathcal{L}_2$ projection of the prescribed values, i.e $\widehat{u} = \mathbb{P}_2(u_D)$, has to be considered to ensure a superconvergent solution u^*, see Remark 2. Otherwise, the average $\int_{K_i} u \, dV$ may not converge with order $k+2$, and u^* may not reach the full rate of convergence $k+2$.

1.3.2 Hybridizable Discontinuous Galerkin Method for Stokes Problem

The following Stokes problem is considered,

$$\begin{aligned} -\boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} \boldsymbol{u}) + \boldsymbol{\nabla} p &= \boldsymbol{f} \quad \text{in } \Omega \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} &= \mathbf{0} \quad \text{in } \Omega \\ \boldsymbol{u} &= \boldsymbol{u}_D \quad \text{on } \partial\Omega \\ \frac{1}{|\Omega|} \int_\Omega p \, dV &= \rho_\Omega, \end{aligned} \tag{1.17}$$

where $\boldsymbol{u}$ is the velocity, p is the pressure, ν is the material viscosity, $\mathbf{I}$ is the second-order identity tensor, $\boldsymbol{f}$ is an external force, $\boldsymbol{u}_D$ are prescribed values on the exterior boundary and ρ_Ω is the mean of the pressure in Ω. On $\partial\Omega$, Dirichlet boundary conditions are considered; however, any other boundary condition could as well be implemented at $\partial\Omega$ with no difficulty see for instance Nguyen et al. [2010].

The domain Ω is assumed to be covered by a finite element mesh with elements $\{K_i\}_{i=1}^{\mathtt{n_{el}}}$ and faces $\{\Gamma_i\}_{i=1}^{\mathtt{n_{fc}}}$ satisfying the conditions in (1.2) and (1.3). Now the problem (1.17) can be tackled by first solving the so-called local problem, and then closing the problem with some global equations imposed at the inter-element boundaries as well as on the outer boundary that is,

$$\begin{aligned} \boldsymbol{L} - \boldsymbol{\nabla} \boldsymbol{u} &= \boldsymbol{0} \quad \text{in } K_i \\ \boldsymbol{\nabla} \cdot (-\nu \boldsymbol{L} + p\mathbf{I}) &= \boldsymbol{f} \quad \text{in } K_i \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} &= \boldsymbol{0}, \quad \text{in } K_i \\ \boldsymbol{u} &= \widehat{\boldsymbol{u}} \quad \text{on } \partial K_i \\ \frac{1}{|K_i|} \int_{K_i} p \, dV &= \rho_i, \end{aligned} \tag{1.18}$$

for $i = 1, \ldots, \mathtt{n_{el}}$ and,

$$[\![(-\nu \boldsymbol{L} + p\mathbf{I}) \cdot \boldsymbol{n}]\!] = 0 \quad \text{on } \Gamma, \tag{1.19a}$$

$$\int_{\partial K_i} \widehat{\boldsymbol{u}} \cdot \boldsymbol{n} \, dS = 0 \text{ for } i = 1, \ldots, \mathtt{n_{el}}, \tag{1.19b}$$

$$\widehat{\boldsymbol{u}} = \mathbb{P}_2(\boldsymbol{u}_D) \quad \text{on } \partial\Omega \text{ and } \sum \frac{|K_i|}{|\Omega|} \rho_i = \rho_\Omega. \tag{1.19c}$$

In the above equations the variable $\boldsymbol{L}$ corresponds to the gradient of $\boldsymbol{u}$, allowing the splitting of the PDE in two first order PDEs, $\widehat{\boldsymbol{u}}$ corresponds to the trace of $\boldsymbol{u}$ at the mesh faces Γ, and ρ_i corresponds to the mean of pressure on the element, which is a single scalar for each element. Equations (1.18) correspond to the Stokes problem with Dirichlet boundary conditions, $\boldsymbol{u} = \widehat{\boldsymbol{u}}$. Note that the Stokes problem with only Dirichlet data does not have a unique solution, so the last equation in (1.18) is necessary for well posedness of the problem, by setting the mean of the pressure in the element to the new variable ρ_i. The local problem (1.18) can be solved element-by-element given $\widehat{\boldsymbol{u}}$ and ρ_i. Thus, now the problem reduces to determine $\widehat{\boldsymbol{u}}$ and $\{\rho_i\}_{i=1}^{\mathtt{n_{el}}}$ with the global equations (1.19). Equation (1.19a) is the conservativity condition which imposes continuity of the normal component of the traction across internal faces, where the jump operator is defined as in equation (1.7). With (1.19b), the incompressibility condition on the boundary of the elements is imposed and finally with (1.19c) Dirichlet boundary conditions on the external boundary is set and the pressure constant is fixed to the global mean value. Discretization spaces, $\mathcal{V}^h$ and Λ^h, defined in (1.8), are considered now for elemental variables $\boldsymbol{u}$, $\boldsymbol{L}$ and p, and for the trace variable $\widehat{\boldsymbol{u}}$, respectively. Then, the discretization of the local problem

(1.18) and the global equations (1.19) leads to the HDG formulation: find $\boldsymbol{u} \in [\mathcal{V}^h]^d$, $\boldsymbol{L} \in [\mathcal{V}^h]^{d\times d}$, $p \in \mathcal{V}^h$, $\widehat{\boldsymbol{u}} \in \Lambda^h$ and $\{\rho_i\}_{i=1}^{\mathtt{n_{el}}}$ such that

$$\begin{aligned}
\int_{K_i} \boldsymbol{\nabla}\cdot(-\nu\boldsymbol{L})\cdot\boldsymbol{v}\,dV + \int_{K_i}\boldsymbol{\nabla}p\cdot\boldsymbol{v}\,dV + \int_{\partial K_i}\tau\nu(\boldsymbol{u}-\widehat{\boldsymbol{u}})\cdot\boldsymbol{v}\,dS &= \int_{K_i}\boldsymbol{f}\cdot\boldsymbol{v}\,dV \\
\int_{K_i}\boldsymbol{L}:\boldsymbol{Q}\,dV + \int_{K_i}(\boldsymbol{\nabla}\cdot\boldsymbol{Q})\cdot\boldsymbol{u}\,dV - \int_{\partial K_i}(\boldsymbol{Q}\cdot\boldsymbol{n})\cdot\widehat{\boldsymbol{u}}\,dS &= 0 \\
\int_{K_i}\boldsymbol{u}\cdot\boldsymbol{\nabla}q\,dV - \int_{\partial K_i}(\widehat{\boldsymbol{u}}\cdot\boldsymbol{n})q\,dS &= 0 \\
\frac{1}{|K_i|}\int_{K_i} p\,dV &= \rho_i
\end{aligned} \tag{1.20a}$$

for $i = 1,\ldots,\mathtt{n_{el}}$ together with,

$$\begin{aligned}
\int_\Gamma \widehat{\boldsymbol{v}}\cdot[\![(-\nu\boldsymbol{L}+p\mathbf{I})\cdot\boldsymbol{n}]\!]\,dS + 2\int_\Gamma \widehat{\boldsymbol{v}}\cdot(\{\nu\tau\boldsymbol{u}\} - \{\nu\tau\}\widehat{\boldsymbol{u}})\,dS &= 0 \\
\int_{\partial K_i}\widehat{\boldsymbol{u}}\cdot\boldsymbol{n}\,dS = 0 \text{ for } i = 1,\ldots,\mathtt{n_{el}} & \\
\widehat{\boldsymbol{u}} = \mathbb{P}_2(\boldsymbol{u}_D) \text{ on } \partial\Omega \text{ and } \sum\frac{|K_i|}{|\Omega|}\rho_i = \rho_\Omega, &
\end{aligned} \tag{1.20b}$$

for all $\boldsymbol{v} \in [\mathcal{V}^h]^d$, $\boldsymbol{Q} \in [\mathcal{V}^h]^{d\times d}$, $q \in \mathcal{V}^h$, and for all $\widehat{\boldsymbol{v}} \in \Lambda^h$ such that $\widehat{\boldsymbol{v}} = 0$ on $\partial\Omega$. The first equation in (1.20a) can be derived from the first equation in (1.18) by applying integration by parts, replacing the velocity gradient by the numerical velocity gradient,

$$\widehat{\boldsymbol{L}} := \boldsymbol{L} + \tau(\widehat{\boldsymbol{u}} - \boldsymbol{u})\otimes\boldsymbol{n}, \tag{1.21}$$

and undoing the integration by parts. The second and third equations are obtained from the weak form of the second and third equations in (1.18) by simply applying integration by parts and inserting the boundary condition $\boldsymbol{u} = \widehat{\boldsymbol{u}}$ on the element boundary. The weak form of the global equation (1.19a) is presented in the first equation of (1.20b) where the mean $\{\cdot\}$ and jump $[\![\cdot]\!]$ operators are defined in (1.15) and (1.7), respectively. The flux $\boldsymbol{L}$ is replaced by the numerical flux (1.21) as usual. The discretization of the first three equations in (1.20) leads to a system of equations of the form

$$\begin{bmatrix} \mathbf{A}_{uu}^{K_i} & \mathbf{A}_{uL}^{K_i} & \mathbf{A}_{up}^{K_i} \\ \mathbf{A}_{Lu}^{K_i} & \mathbf{A}_{LL}^{K_i} & \mathbf{0} \\ \mathbf{A}_{pu}^{K_i} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^i \\ \mathbf{L}^i \\ \mathbf{p}^i \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{A}_{u\widehat{u}}^{K_i} \\ \mathbf{A}_{L\widehat{u}}^{K_i} \\ \mathbf{A}_{p\widehat{u}}^{K_i} \end{bmatrix} \boldsymbol{\Lambda}^i \tag{1.22}$$

where $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$ are the vectors of nodal values of $\boldsymbol{u}$, $\boldsymbol{L}$ and p in element K_i, and $\boldsymbol{\Lambda}^i$ is the vector of nodal values of $\widehat{\boldsymbol{u}}$ on the n faces of the element as defined in (1.12).

The system (1.22) is determined up to a constant, hence imposing the discrete form of the last equation in (1.20a),

$$\mathbf{A}_{\rho p}^{K_i}\mathbf{p}^i = \rho_i, \tag{1.23}$$

to the the final system of the local problem with Lagrange multipliers (1.20a) leads to

$$\left[\begin{array}{ccc:c} \mathbf{A}_{uu}^{K_i} & \mathbf{A}_{uL}^{K_i} & \mathbf{A}_{up}^{K_i} & \mathbf{0} \\ \mathbf{A}_{Lu}^{K_i} & \mathbf{A}_{LL}^{K_i} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{pu}^{K_i} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^{K_i T} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^{K_i} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \mathbf{u}^i \\ \mathbf{L}^i \\ \mathbf{p}^i \\ \hdashline \lambda \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f} \\ \mathbf{0} \\ \mathbf{0} \\ \hdashline \mathbf{0} \end{array}\right] + \left[\begin{array}{c} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \hdashline \mathbf{1} \end{array}\right] \rho_i - \left[\begin{array}{c} \mathbf{A}_{u\widehat{u}}^{K_i} \\ \mathbf{A}_{L\widehat{u}}^{K_i} \\ \mathbf{A}_{p\widehat{u}}^{K_i} \\ \hdashline \mathbf{0} \end{array}\right] \boldsymbol{\Lambda}^i \tag{1.24}$$

System (1.24) can be solved for $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$ in each element, leading to the so-called *local solver* in the element K_i that is,

$$\begin{gathered} \mathbf{u}^i = \mathbf{S}_u^{K_i}\boldsymbol{\Lambda}^i + \mathbf{R}_u^{K_i}\rho_i + \mathbf{f}_U^{K_i}, \qquad \mathbf{L}^i = \mathbf{S}_L^{K_i}\boldsymbol{\Lambda}^i + \mathbf{R}_L^{K_i}\rho_i + \mathbf{f}_L^{K_i} \\ \mathbf{p}^i = \mathbf{S}_p^{K_i}\boldsymbol{\Lambda}^i + \mathbf{R}_p^{K_i}\rho_i + \mathbf{f}_p^{K_i}, \end{gathered} \tag{1.25}$$

with,

$$\left[\begin{array}{c} \mathbf{S}_u^{K_i} \\ \mathbf{S}_L^{K_i} \\ \mathbf{S}_p^{K_i} \\ \mathbf{S}_\lambda^{K_i} \end{array}\right] = -\mathbb{A}^{-1}\left[\begin{array}{c} \mathbf{A}_{u\widehat{u}}^{K_i} \\ \mathbf{A}_{L\widehat{u}}^{K_i} \\ \mathbf{A}_{p\widehat{u}}^{K_i} \\ \mathbf{0} \end{array}\right], \quad \left[\begin{array}{c} \mathbf{R}_u^{K_i} \\ \mathbf{R}_L^{K_i} \\ \mathbf{R}_p^{K_i} \\ \mathbf{R}_\lambda^{K_i} \end{array}\right] = \mathbb{A}^{-1}\left[\begin{array}{c} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{array}\right], \quad \left[\begin{array}{c} \mathbf{f}_U^{K_i} \\ \mathbf{f}_L^{K_i} \\ \mathbf{f}_P^{K_i} \\ \mathbf{f}_\lambda^{K_i} \end{array}\right] = \mathbb{A}^{-1}\left[\begin{array}{c} \mathbf{f} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{array}\right] \tag{1.26}$$

and

$$\mathbb{A} = \left[\begin{array}{cccc} \mathbf{A}_{uu}^{K_i} & \mathbf{A}_{uL}^{K_i} & \mathbf{A}_{up}^{K_i} & \mathbf{0} \\ \mathbf{A}_{Lu}^{K_i} & \mathbf{A}_{LL}^{K_i} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{pu}^{K_i} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^{K_i T} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^{K_i} & \mathbf{0} \end{array}\right]$$

It is important to note that in (1.25) elemental variables $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$ are expressed in terms of only $\boldsymbol{\Lambda}^i$ and ρ_i which is what makes this method *hybridizable*. Discretization of the global equations (1.19a) and (1.19b) leads to the following set of equations

$$\begin{aligned} \mathbf{A}_{\widehat{u}u}^{f,L}\mathbf{u}^{L(f)} + \mathbf{A}_{\widehat{u}\boldsymbol{L}}^{f,L}\mathbf{L}^{L(f)} + \mathbf{A}_{\widehat{u}p}^{f,L}\mathbf{p}^{L(f)} \; + \; \mathbf{A}_{\widehat{u}u}^{f,R}\mathbf{u}^{R(f)}(f) + \mathbf{A}_{\widehat{u}\boldsymbol{L}}^{f,R}\mathbf{L}^{R(f)} \\ + \mathbf{A}_{\widehat{u}p}^{f,R}\mathbf{p}^{R(f)} \; + \; \mathbf{A}_{\widehat{u}\widehat{u}}^{f}\widehat{\mathbf{u}}^f = 0 \end{aligned} \tag{1.27}$$

for $f = 1, \ldots, \mathtt{n_{fc}}$ and

$$\mathbf{A}^{K_i}_{\rho\widehat{u}} \mathbf{\Lambda}^i = 0$$

for $i = 1, \ldots, \mathtt{n_{el}}$.

Replacing the local solver (1.25), for the elements $K_{L(f)}$ and $K_{R(f)}$, in (1.27) for every face Γ_f, leads to a system of equations involving only the trace variables $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{n_{fc}}}$ and $\{\rho\}_i^{\mathtt{n_{el}}}$ as unknowns, and can be uniquely solved after setting Dirichlet boundary conditions and pressure constraint (1.19c). The elemental unknowns are then determined by simple put-back using the local solver (1.25). As in the case of the second order elliptic problems; with HDG, a second element by element post processing leads to superconvergent solution of both of the velocity components here as well. The application is identical as it is explained in Remark 2.

1.3.3 Numerical integration on cut elements

The usual X-FEM strategy for the geometrical description of interfaces and numerical integration is adopted. An interface $\mathcal{I}$ is represented as the zero isoline of a level set function φ, that is $\mathcal{I} = \{\boldsymbol{x} \mid \varphi(\boldsymbol{x}) = 0\}$. The level set function φ is assumed to be given by its nodal values in the computational mesh, see the left panel of Figure 1.2. The nodal values of the level set function may correspond, for instance, to the signed distance of the nodes to the interface $\mathcal{I}$.

A proper representation of the interface inside cut elements is crucial. Those elements are split by the interface in two subregions, corresponding to two disctinct materials or a material and the void, and a numerical quadrature has to be defined to integrate over each of the material domains or only over the material domain. The usual practice for first order computations is considering a linear interface representation in each cut element. This strategy provides optimal convergence rates for linear approximations, but it is clearly not suitable for high-order computations. The geometrical error due to the low resolution representation of the interface leads to poor accuracy and convergence rates limited to order $\mathcal{O}(h^{3/2})$ in $\mathcal{H}^1$ norm, see for instance Dréau et al. [2010]. Two main strategies have been proposed in the literature to properly represent an interface for integration purposes in k-th order computations: (i) a fine enough piecewise linear representation of the interface in each cut element Dréau et al. [2010], Gross and Reusken [2007] or (ii) a r-th degree parametrization to approximate the interface Cheng and Fries [2010] .

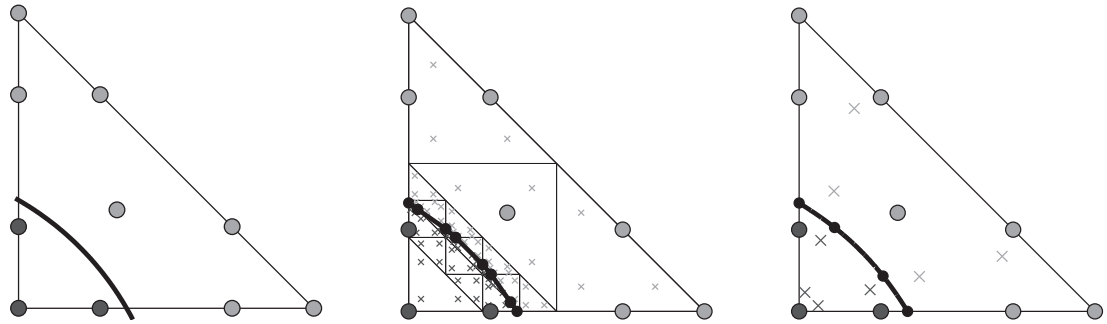


Figure 1.2: 3-th order triangle cut by an interface (left). Interface representation and numerical quadrature with (center) piecewise linear approximation based on an oct-tree partition, and (right) 3-th degree parametrization. The color on the nodes refer to the sign of the level set function. Crosses are integration points.

In Dréau et al. [2010], an octree-like partition of the element in integration cells is recursively defined to get a piecewise linear representation of the interface, with segments of the desired size $\tilde{h}$. Cells intersected by the interface are divided in two subregions for integration, see for example Figure 1.2 center. Special care has to be taken to the level of refinement in order to get accurate results and optimal convergence rates. In fact, optimal asymptotic convergence can not be obtained with a constant ratio $\tilde{h}/h$ and, as noted in Dréau et al. [2010], further refinement of the integration cells is necessary as the computational mesh is refined. In fact, in practice, for a given problem and computational mesh, the accuracy of the solution may show strong dependence on the integration cell size $\tilde{h}$, and the selection of a small enough cell size may be not straightforward.

Remark 4. In Figures 1.2, 1.3 and 1.4, illustrating strategies for the approximation of the interface and numerical integration, the representation is done in the reference element, where all computations are performed based on a transformation as usual in finite element methods.

Alternatively a k-th degree parametrization for the representation of the interface in each element is proposed in Cheng and Fries [2010], see for example Figure 1.2 right, leading to accurate results and optimal convergence rates with fewer integration points (i.e. lower computational cost). However, the strategy proposed in Cheng and Fries [2010] requires a fine enough mesh such that the interface cuts the element boundary twice and not within one side, splitting the triangle in a triangle and a quadrilateral. Thus, this strategy may fail in complicated situations that can usually

appear with a high-order level set on a coarse mesh, such as a bubble inside one element, or an interface cutting more than two sides of an element, see Figure 1.3. To overcome this limitation, a robust and efficient strategy, based on the strategy in Cheng and Fries [2010] but capable to handle more complicated situations, is proposed in Sala Lardies et al. [2012] further implemented in Gürkan et al. [2016] and recalled next.

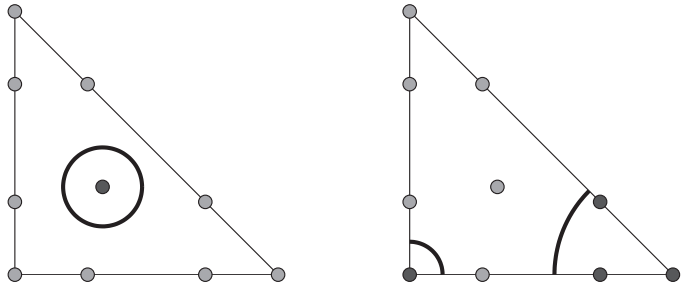


Figure 1.3: Some examples of interfaces described by a 4-th order level set. The color on the nodes refer to the sign of the level set function.

1.3.3.1 Numerical integration at cut elements with piecewise k-th interfaces

Cheng and Fries [2010] consider a k-th degree polynomial parametrization for the representation of the interface in triangular elements. In their work, the mesh is assumed to be fine enough so that all cut elements correspond to what here will be called *basic* element; that is, the interface splits the triangle in a triangular and quadrilateral region or, in other words, the interface cuts two sides of the triangle with just one intersection on each side. Under this assumption, the first $k+1$ points on the interface are to be found first, including the two intersections with the element boundary. The $k+1$ points are used as base points for a k-th degree polynomial parametrization ϕ approximating the interface, and dividing the element in two integration subdomains sharing one curved side, see Figure 1.2 right.

A transformation from a straight-sided element can now be used to define the numerical quadrature in each subdomain, or just in the region in the domain Ω in our case. To avoid generating all the nodes necessary for the use of the standard isoparametric transformation, which would require a proper location of interior nodes—see Ciarlet and Raviart [1972]—in each subregion, the use of specially designed transformations for elements with only one curved side is suggested here, see for instance

Sevilla et al. [2011]. In particular, in a *basic* element cut by an interface parametrized by $\boldsymbol{\phi}(s)$ for $s \in (-1,1)$, the following transformations can be used:

$$(\xi(s,t),\eta(s,t)) = \frac{1-t}{2}\mathbf{C} + \frac{1+t}{2}\boldsymbol{\phi}(s), \qquad s,t \in (-1,1)$$

for a triangular subdomain corresponding to the convex hull of the curved side and node $\mathbf{C}$, and

$$(\xi(s,t),\eta(s,t)) = \frac{1-t}{2}\left[\frac{1-s}{2}\mathbf{C} + \frac{1+s}{2}\mathbf{D}\right] + \frac{1+t}{2}\boldsymbol{\phi}(s), \qquad s,t \in (-1,1)$$

for a curved quadrilateral corresponding to the convex hull of the curved size and nodes $\mathbf{C}$ and $\mathbf{D}$ (for a quadrilateral $\{\boldsymbol{\phi}(-1),\ \boldsymbol{\phi}(1),\ \mathbf{D}\ ,\ \mathbf{C}\}$ being properly oriented, or switching $\mathbf{C}$ and $\mathbf{D}$ otherwise).

Numerical examples in Cheng and Fries [2010] illustrate how the approximation of the interface with a piecewise k-th degree parametrization in each element leads to optimal convergence rates. Nevertheless, in practical applications, the assumption of a fine enough mesh, such that all cut elements are *basic* elements, is too restrictive for high-order computations and hampers the robustness of the method. High-order level sets may lead to very complex interfaces, see some examples in Figure 1.3, that are not contemplated in Cheng and Fries [2010]. In a more general context, an interface can split an element into more than two regions. Moreover, even if it is split in only two regions, casuistic (interior bubble, two cuts within one side, etc) may impede the implementation of a unique k-th degree polynomial parametrization.

In Gürkan et al. [2016] a simple idea is proposed: divide and conquer. That is, any element intersected by the interface in a complex manner (i.e. not a *basic* element) is recursively divided in integration cells until all cells can be considered either *basic* cells—i.e. cells whose boundary is cut by the interface twice not within one side—or not intersected by the interface. Nodal values of the level-set function are interpolated from the original element to get nodal values in the integration cells. Then, the strategy considering a k-th degree parametrization of the interface is applied for all *basic cells*, leading to a piecewise k-th degree interface representation in the element. Note that if the recursive division in cells is done using the same type of element, the resulting level set is exactly the same polynomial level set, and no information is lost during the process.

For an easy implementation, the decision of splitting an element or cell can be based on the changes of sign of the level-set nodal values, first looking at nodal values at sides and then looking to interior nodal values. An element or cell is to be split if:

(i) one of its sides is cut more than once by the interface (there is more than one changes of sign of the nodal values of the level set in the side), or

(ii) it has more than two sides cut by the interface, or

(iii) its boundary is not intersected by the interface, but it has one or more interior nodes with sign of the level set different to the boundary sign (interior bubble or void).

Figure 1.4 shows an example of recursive division and the corresponding numerical quadrature for a 3rd order finite element. Nodal values with different sign are marked with different colors.

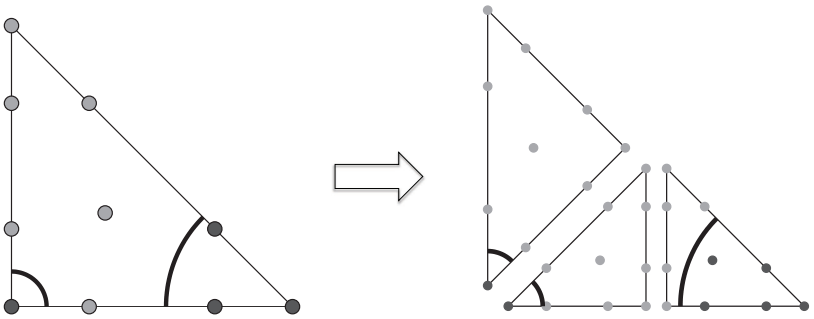


Figure 1.4: Example of recursive division for an element cut by the interface in a complex manner.

The strategy considering a piecewise k-th degree parametrization to approximate the interface provides optimal convergence rates for any order, in a robust manner. Moreover, in practical applications most of the elements cut by the interface will be in the *basic* situation, and paying little attention to the casuistic for the recursive division in the complex elements, they will usually be divided in few integration cells, with little increase in computational cost.

Chapter 2

X-HDG for Second Order Elliptic Void Problems

Next, the X-HDG formulation is detailed for problems including void interfaces. In section 2.1, X-HDG formulation for second order elliptic void problems is introduced, where Neumann boundary conditions are considered at the interface. Next in section 2.2, the case where Dirichlet boundary condition is imposed at the interface is studied. The formulation is validated in section 2.3 and, finally, the chapter is closed with some general conclusions and final remarks at section 2.4. The developments presented in this chapter can also be found in Gürkan et al. [2016].

2.1 X-HDG for problems with Neumann voids

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with an interior boundary $\mathcal{I}$ (also referred as interface) and an exterior boundary $\partial\Omega^{ext} := \partial\Omega \backslash \mathcal{I}$. The following problem is considered,

$$\begin{aligned} -\boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} u) &= f \quad \text{in } \Omega \\ \nu \boldsymbol{\nabla} u \cdot \boldsymbol{n} &= g \quad \text{on } \mathcal{I} \\ u &= u_D \quad \text{on } \partial\Omega^{ext} \end{aligned} \tag{2.1}$$

where u is the solution, ν is a material coefficient, f is a given source term, u_D are prescribed values on the exterior boundary, and g is a prescribed flux on the interior boundary, i.e. the voids boundary. Neumann boundary conditions are considered in the interior boundary $\mathcal{I}$; the implementation of Dirichlet boundary conditions on $\mathcal{I}$

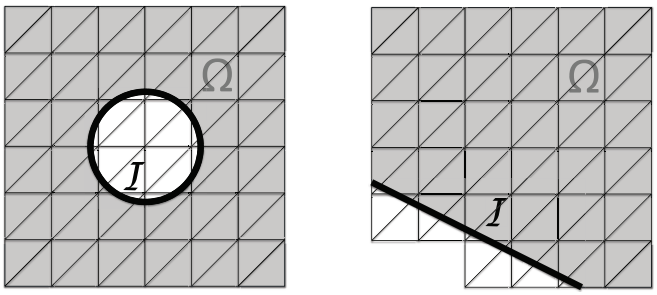


Figure 2.1: Two examples of domain with a void: a circular void boundary and a straight interface, $\mathcal{I}$ in black. The mesh covers the domain Ω (in gray) and fits the exterior boundary $\partial\Omega^{ext} = \partial\Omega \backslash \mathcal{I}$.

is developed in Section 2.2. Here, for simplicity, Dirichlet boundary conditions are considered on the exterior boundary $\partial\Omega^{ext}$; other boundary conditions at $\partial\Omega^{ext}$ do not add any difficulty, since they are implemented as in standard HDG.

The domain Ω is now assumed to be covered by a finite element mesh with $\mathtt{n_{el}}$ disjoint elements K_i, such that

$$\overline{\Omega} \subset \bigcup_{i=1}^{\mathtt{n_{el}}} \overline{K}_i, \quad K_i \cap K_j = \emptyset \text{ for } i \neq j, \quad \partial\Omega^{ext} \subset \partial\left[\bigcup_{i=1}^{\mathtt{n_{el}}} \overline{K}_i\right] \tag{2.2}$$

Note that the mesh fits the exterior boundary $\partial\Omega^{ext}$, but some elements may be cut by the interior boundary $\mathcal{I}$, see Figure 2.1. The union of all $\mathtt{n_{fc}}$ faces Γ_i (sides for 2D) intersecting the domain Ω is denoted as

$$\Gamma := \bigcup_{i=1}^{\mathtt{n_{el}}} \left[\partial K_i \cap \overline{\Omega}\right] = \bigcup_{f=1}^{\mathtt{n_{fc}}} \left[\Gamma_f \cap \overline{\Omega}\right]. \tag{2.3}$$

Now, as for standard HDG, the problem is rewritten with some element-by-element equations and some global ones. The local element-by-element problems correspond to the statement of the PDE in (2.1) with essential boundary conditions at each element K_i, that is,

$$\left.\begin{aligned} \boldsymbol{\nabla}\cdot\boldsymbol{q} &= f \quad \text{in } K_i \\ \boldsymbol{q} + \nu\boldsymbol{\nabla}u &= 0 \quad \text{in } K_i \\ u &= \widehat{u} \quad \text{on } \partial K_i \end{aligned}\right\} \text{ if } K_i \subset \Omega \tag{2.4a}$$

$$\left.\begin{aligned} \boldsymbol{\nabla}\cdot\boldsymbol{q} &= f \quad \text{in } \Omega_i \\ \boldsymbol{q} + \nu\boldsymbol{\nabla}u &= 0 \quad \text{in } \Omega_i \\ \boldsymbol{q}\cdot\boldsymbol{n} &= g \quad \text{on } \mathcal{I}_i \\ u &= \widehat{u} \quad \text{on } \partial\Omega_i \backslash \mathcal{I}_i \end{aligned}\right\} \text{ if } \mathcal{I} \cap K_i \neq \emptyset \tag{2.4b}$$

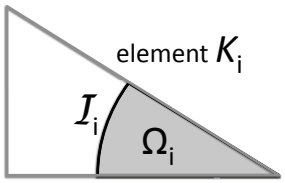


Figure 2.2: Element cut by the interior boundary $\mathcal{I}$: the intersection of the element K_i with the interface and the domain (in gray) are denoted as $\mathcal{I}_i$ and Ω_i.

for $i = 1, \ldots, \mathtt{n_{el}}$, where, for cut elements,

$$\Omega_i := \Omega \cap K_i, \quad \mathcal{I}_i := \mathcal{I} \cap K_i \tag{2.5}$$

The local problems have been particularized for elements cut by the interior boundary (2.4b) and standard elements (2.4a).

As usual, then the problem is closed using the *conservativity conditions* (i.e. *global equations*), that is, the continuity of the flux across element boundaries

$$[\![\boldsymbol{q} \cdot \boldsymbol{n}]\!] = 0 \quad \text{on } \Gamma \backslash \partial\Omega^{ext}, \tag{2.6}$$

and the boundary condition, equivalent to the exterior boundary condition in (2.1),

$$\widehat{u} = \mathbb{P}_2(u_D) \quad \text{on } \partial\Omega^{ext}, \tag{2.7}$$

where the *jump* $[\![\cdot]\!]$ operator is defined as in (1.7).

The discretization of the conservativity condition (2.6) and the local problems (2.4), with the boundary condition (2.7), leads to the X-HDG formulation. The following discrete spaces for elemental variables, u and $\boldsymbol{q}$, and for the trace variable, $\widehat{u}$, are considered

$$\begin{aligned} \mathcal{V}^h &:= \left\{ v \in \mathcal{L}_2(\Omega) \ : v|_{K_i \cap \Omega} \in \mathcal{P}_k(K_i \cap \Omega) \ \text{ for } \ i = 1, \ldots, \mathtt{n_{el}} \right\} \\ \Lambda^h &:= \left\{ \hat{v} \in \mathcal{L}_2(\Gamma) \ : \hat{v}|_{\Gamma_f \cap \overline{\Omega}} \in \mathcal{P}_k(\Gamma_f \cap \overline{\Omega}) \ \text{ for } \ f = 1, \ldots, \mathtt{n_{fc}} \right\}, \end{aligned} \tag{2.8}$$

where $\mathcal{P}_k$ denotes the space of polynomials of degree less or equal to k.

The next sections present the details of the X-HDG formulation, stating the discretization of the local problems for standard and cut elements and the discretization of the conservativity condition (2.6). The local problem at elements not cut by the interface (2.4a) and the global problem are discretized as usual in HDG , see Cockburn et al. [2009, 2008], as recalled in section 1.3.1. The discretization of the local problem for cut elements (2.4b) is developed in section 2.1.2.

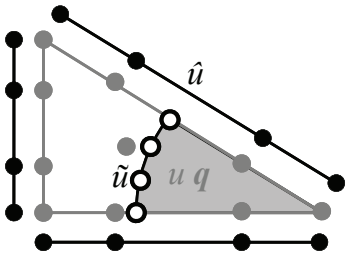


Figure 2.3: X-HDG discretization in an element cut by the interior Neumann boundary: gray for elemental variables, u and $\boldsymbol{q}$, black nodes for trace variable $\widehat{u}$ and white nodes for the trace on the Neuman boundary $\widetilde{u}$.

2.1.1 Local problem for standard elements

Local problem of a non-cut element K_i in X-HDG setting simply leads to the discretization of equations (2.4a). The details of the discretization and how to obtain the *local solver* for a standard element is presented in detail in Section 1.3.1 which is basically obtained by following standard HDG formulation.

2.1.2 Local problem for a cut element

The X-HDG local problem at an element K_i cut by the interior boundary corresponds to the discretization of (2.4b), that is: given $\widehat{u} \in \Lambda^h$, find $u \in \mathcal{P}_k(\Omega_i)$, $\boldsymbol{q} \in [\mathcal{P}_k(\Omega_i)]^d$ such that

$$\begin{aligned} &\int_{\Omega_i} v\boldsymbol{\nabla}\cdot\boldsymbol{q}\,dV + \int_{\partial\Omega_i\setminus\mathcal{I}_i} \nu\tau v(u-\widehat{u})\,dS + \int_{\mathcal{I}_i} \nu\tau v(u-\widetilde{u})\,dS = \int_{\Omega_i} vf\,dV \\ &\int_{\Omega_i} \boldsymbol{q}\cdot\boldsymbol{w}\,dV - \int_{\Omega_i} \nu u\boldsymbol{\nabla}\cdot\boldsymbol{w}\,dV + \int_{\partial\Omega_i\setminus\mathcal{I}_i} \nu\widehat{u}\boldsymbol{w}\cdot\boldsymbol{n}\,dS + \int_{\mathcal{I}_i} \nu\widetilde{u}\boldsymbol{w}\cdot\boldsymbol{n}\,dS = 0 \end{aligned} \tag{2.9}$$

for all $v \in \mathcal{P}_k(\Omega_i)$ and $\boldsymbol{w} \in [\mathcal{P}_k(\Omega_i)]^d$, where $\Omega_i = \Omega \cap K_i$, and $\widetilde{u}$ is a new trace variable approximating the trace of the solution on the interface $\mathcal{I}_i = \mathcal{I} \cap K_i$, see Figure 2.3. Compared to the weak form for standard elements (1.9), the X-HDG weak form for a cut element has two additional terms corresponding to integrals along the interface $\mathcal{I}_i$, involving the new trace variable $\widetilde{u}$.

The discretization of the local problem (2.9) leads to a system of equations of the form

$$\begin{cases} \left[\mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i}\right]\mathbf{u}^i & +\mathbf{A}_{uq}^{\Omega_i}\mathbf{q}^i + \mathbf{A}_{u\widehat{u}}^{\Omega_i}\mathbf{\Lambda}^i + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\widetilde{\boldsymbol{u}} = \mathbf{f}_u^{\Omega_i} \\ \mathbf{A}_{qu}^{\Omega_i}\mathbf{u}^i & +\mathbf{A}_{qq}^{\Omega_i}\mathbf{q}^i + \mathbf{A}_{q\widehat{u}}^{\Omega_i}\mathbf{\Lambda}^i + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\widetilde{\boldsymbol{u}} = \mathbf{0} \end{cases}, \tag{2.10}$$

similar to (1.11), but with three new matrices corresponding to integrals on the Neumann boundary (marked with the superindex $\mathcal{I}_i$), and the nodal values for the new trace variable, $\widetilde{\boldsymbol{u}}$.

The local problem is now closed by imposing the Neumann boundary condition on $\mathcal{I}_i$, i.e.

$$\widehat{\boldsymbol{q}} \cdot \boldsymbol{n} = g \quad \text{on } \mathcal{I}_i.$$

Replacing the expression of the numerical flux $\widehat{\boldsymbol{q}}$ defined in (1.10), i.e. $\widehat{\boldsymbol{q}} \cdot \boldsymbol{n} = \boldsymbol{q} \cdot \boldsymbol{n} + \nu\tau(u - \widetilde{u})$, the weak form of the Neumann condition in $\mathcal{I}^i$ is: given $u \in \mathcal{P}_k(\Omega_i)$, $\boldsymbol{q} \in [\mathcal{P}_k(\Omega_i)]^d$, find $\widetilde{u} \in \mathcal{P}_k(\mathcal{I}_i)$ such that

$$\int_{\mathcal{I}_i} \widetilde{v}(\boldsymbol{q} \cdot \boldsymbol{n})\, dS + \nu\tau \int_{\mathcal{I}_i} \widetilde{v}\,(u - \widetilde{u})\; dS = \int_{\mathcal{I}_i} \widetilde{v} g\, dS \quad \forall \widetilde{v} \in \mathcal{P}_k(\mathcal{I}_i). \tag{2.11}$$

Thus, the discretization of the Neumann boundary condition leads to a discrete equation of the form

$$\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}u}\mathbf{u}^i + \mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}q}\mathbf{q}^i + \mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}\widetilde{u}}\widetilde{\boldsymbol{u}} = \mathbf{g},$$

that allows expressing the new trace values $\widetilde{\boldsymbol{u}}$ in terms of elemental values

$$\widetilde{\boldsymbol{u}} = \mathbf{T}^i_u\mathbf{u}^i + \mathbf{T}^i_q\mathbf{q}^i - \mathbf{t}^i. \tag{2.12}$$

with

$$\mathbf{T}^i_u = -[\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}\widetilde{u}}]^{-1}\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}u}, \quad \mathbf{T}^i_q = -[\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}\widetilde{u}}]^{-1}\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}q}, \quad \mathbf{t}^i = -[\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}\widetilde{u}}]^{-1}\mathbf{g}.$$

Replacing (2.12) in (2.10) leads to the final discrete local problem

$$\begin{cases} \left[\mathbf{A}^{\Omega_i}_{uu} + \mathbf{A}^{\mathcal{I}_i}_{uu} + \mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}}\mathbf{T}^i_u\right]\mathbf{u}^i & + \left[\mathbf{A}^{\Omega_i}_{uq} + \mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}}\mathbf{T}^i_q\right]\mathbf{q}^i + \mathbf{A}^{\Omega_i}_{u\widehat{u}}\boldsymbol{\Lambda}^i = \mathbf{f}^{\Omega_i}_u + \mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}}\mathbf{t}^i \\ \left[\mathbf{A}^{\Omega_i}_{qu} + \mathbf{A}^{\mathcal{I}_i}_{q\widetilde{u}}\mathbf{T}^i_u\right]\mathbf{u}^i & + \left[\mathbf{A}^{\Omega_i}_{qq} + \mathbf{A}^{\mathcal{I}_i}_{q\widetilde{u}}\mathbf{T}^i_q\right]\mathbf{q}^i + \mathbf{A}^{\Omega_i}_{q\widehat{u}}\boldsymbol{\Lambda}^i = \mathbf{A}^{\mathcal{I}_i}_{q\widetilde{u}}\mathbf{t}^i \end{cases}$$

Now, similarly to (1.11) for standard elements, this system can be solved for $\mathbf{u}^i$ and $\mathbf{q}^i$, obtaining the *local solver* in the cut element K_i,

$$\mathbf{u}^i = \mathbf{U}^{K_i}\boldsymbol{\Lambda}^i + \mathbf{f}^{K_i}_U, \quad \mathbf{q}^i = \mathbf{Q}^{K_i}\boldsymbol{\Lambda}^i + \mathbf{f}^{K_i}_Q, \tag{2.13}$$

with

$$\begin{bmatrix} \mathbf{U}^{K_i} \\ \mathbf{Q}^{K_i} \end{bmatrix} = -\mathbb{A}^{-1}\begin{bmatrix} \mathbf{A}^{\Omega_i}_{u\widehat{u}} \\ \mathbf{A}^{\Omega_i}_{q\widehat{u}} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{f}^{K_i}_U \\ \mathbf{f}^{K_i}_Q \end{bmatrix} = \mathbb{A}^{-1}\begin{bmatrix} \mathbf{f}^{\Omega_i}_u + \mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}}\mathbf{t}^i \\ \mathbf{A}^{\mathcal{I}_i}_{q\widetilde{u}}\mathbf{t}^i \end{bmatrix} \tag{2.14}$$

and

$$\mathbb{A} = \begin{bmatrix} \left[\mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{uq}^{\Omega_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_q^i\right] \\ \left[\mathbf{A}_{qu}^{\Omega_i} + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{qq}^{\Omega_i} + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_q^i\right] \end{bmatrix}$$

It is important noting that the structure of the local solver is exactly the same as for non-cut elements see equation (1.13), thanks to the fact that the internal trace variable $\widetilde{u}$ has been isolated and it is not an unknown of the problem anymore.

Remark 5. Although the approximation in a cut element, or on a cut face, is defined as polynomial functions in $\Omega_i = K_i \cap \Omega$, or on $\Gamma_f \cap \Omega$, respectively, standard nodal basis functions in the whole element K_i, or face Γ_f, are considered, see Figure 2.3. Thus, as usual in X-FEM, the reference element is the standard one but with a modified numerical quadrature to integrate only in the domain Ω, see Section 1.3.3.

Remark 6. It is worth noting that X-HDG has an additional cost compared to HDG, mainly due to the modification of the numerical quadrature for cut elements, and the evaluation of the nodal basis functions and derivatives at the new integration points as recalled in Remark 5. That is, elemental computations for cut elements are substantially more expensive than for standard elements, for which the reference element information and all associated pre-computations can be used. However, in practical applications the ratio of cut elements to standard elements is small so that the extra cost becomes negligible compared to the cost of mesh adaptation or re-meshing in case of moving boundaries.

Remark 7. The traces vector $\mathbf{\Lambda}^i$ can be defined to include the nodal values only for faces intersecting the domain. However, the implementation is simpler if all faces of the element are considered. In this case, the blocks in matrices $\mathbf{U}^{K_i}$ and $\mathbf{Q}^{K_i}$ and in vectors $\mathbf{f}_U^{K_i}$ and $\mathbf{f}_Q^{K_i}$, corresponding to faces not intersecting the domain, are zero blocks; and, therefore, the trace values for these faces do not contribute to the final system.

2.1.3 Global problem

As usual, both in standard and in cut elements local solver of the form(2.13), is obtained such that u and $\boldsymbol{q}$ is expressed only in terms of the trace values at its boundary, $\widehat{u}$. To determine the trace nodal values $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{n_{fc}}}$ on the mesh skeleton Γ

the conservativity condition (2.6) is discretized, $\boldsymbol{q}$ is replaced by the numerical flux (1.10) and the weak form is obtained to find $\widehat{u} \in \Lambda^h$ such that $\widehat{u} = u_D$ on $\partial\Omega^{ext}$

$$\int_\Gamma \widehat{v}[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!]\,dS + 2\nu\tau\int_\Gamma \widehat{v}\,(\{u\}-\widehat{u})\,dS = 0 \quad \forall \widehat{v}\in\Lambda^h,$$

where $\{\cdot\}$ is defined in (1.15). The discrete form of (2.6) reads

$$\mathbf{A}^{f,L}_{\widehat{u}u}\mathbf{u}^{L(f)} + \mathbf{A}^{f,L}_{\widehat{u}q}\mathbf{q}^{L(f)} \;+\; \mathbf{A}^{f,R}_{\widehat{u}u}\mathbf{u}^{R(f)}(f) + \mathbf{A}^{f,R}_{\widehat{u}q}\mathbf{q}^{R(f)} \;+\; \mathbf{A}^{f}_{\widehat{u}\widehat{u}}\widehat{\mathbf{u}}^f = 0. \tag{2.15}$$

Replacing the local solver (1.13) for standard elements and (2.13) for cut elements for $K_{L(f)}$ and $K_{R(f)}$, in (2.15) for every face Γ_f, a final system of equations involving only the trace variables $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{nfc}}$ is obtained.

As usual in a standard HDG code, the implementation of the method involves a loop over elements. For each element, the matrices and vectors for the local solver (2.14) are computed, and the contribution to the equation (2.15) is assembled for each one of the faces of the element. Once the system is assembled for all elements, and Dirichlet boundary conditions (2.7) are imposed, the system can be solved. Then, given the trace variables $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{nfc}}$, the solution, $\mathbf{u}^i$ and $\boldsymbol{q}^i$, can be computed for each cut element using (2.14) and for each standard element using (1.13). It is important noting that X-HDG keeps the structure of a standard HDG code. The only difference is the modified local problem at cut elements (2.10), and the corresponding matrices in the local solver (2.14). After obtaining the solution $\mathbf{u}^i$ and $\boldsymbol{q}^i$ a superconvergent solution u^* can be obtained in X-HDG as in HDG. For the details of how to obtain the superconvergent solution u^* please refer to Remark 2.

Remark 8. As commented in Remark 7, the simplest implementation assembles for all faces, even if they do not intersect the domain. The matrices assembled for faces not intersecting the domain are null matrices and, therefore, the corresponding rows and columns in global system are null. Those rows and columns are to be removed from the system, reducing its size and rendering the system solvable with unique solution, after imposing the Dirichlet boundary condition (2.7).

Remark 9. Boundary conditions on the exterior boundary $\partial\Omega^{ext} = \partial\Omega\backslash\mathcal{I}$ are implemented as in standard HDG. However, special care has to be taken in the presence of exterior faces cut by the interface $\mathcal{I}$, see Figure 2.1 right. If a face Γ_f in the exterior Neumann boundary (i.e. $\Gamma_f \cap \overline{\Omega} \subset \partial\Omega^{ext}$ with Neumann boundary conditions) is cut by the interface , its contribution to the r.h.s. of the system is integrated only in

the domain, that is on $\Gamma_f \cap \overline{\Omega}$, with a suitable numerical quadrature. For a face Γ_f intersecting the exterior boundary, with Dirichlet boundary conditions and cut by the interface, nodal values approximating the prescribed value on $\Gamma_f \cap \overline{\Omega}$ are set for all nodes of the face.

2.2 X-HDG for problems with Dirichlet voids

Let us consider now the problem with Dirichlet boundary conditions at the interior boundary $\mathcal{I}$,

$$\begin{aligned} -\boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} u) &= f \quad \text{in } \Omega \\ u &= u_{\mathcal{I}} \quad \text{on } \mathcal{I} \\ u &= u_D \quad \text{on } \partial\Omega \backslash \mathcal{I} \end{aligned} \tag{2.16}$$

where $u_{\mathcal{I}}$ is the prescribed value.

Similarly to the Neumann case, the problem is split element-by-element into local problems that are linked through the conservativity condition. The local problem for elements not cut by $\mathcal{I}$ is again the standard HDG local problem, leading to the same HDG local solver (1.13) with (1.14). For elements K_i cut by $\mathcal{I}$, the strong form of the local problem is now

$$\begin{aligned} \boldsymbol{\nabla} \cdot \boldsymbol{q} &= f \quad \text{in } \Omega_i \\ \boldsymbol{q} + \nu \boldsymbol{\nabla} u &= 0 \quad \text{in } \Omega_i \\ u &= u_{\mathcal{I}} \quad \text{on } \mathcal{I}_i \\ u &= \widehat{u} \quad \text{on } \partial\Omega_i \backslash \mathcal{I}_i, \end{aligned}$$

where the trace on the interior boundary $u_{\mathcal{I}}$ is a given data in this case.

The weak form of the local problem is then: given $\widehat{u} \in \Lambda^h$, find $u \in \mathcal{P}_k(\Omega_i)$, $\boldsymbol{q} \in [\mathcal{P}_k(\Omega_i)]^d$ such that

$$\begin{aligned} \int_{\Omega_i} v \boldsymbol{\nabla} \cdot \boldsymbol{q} \, dV + \int_{\partial\Omega_i \backslash \mathcal{I}_i} \tau v (u - \widehat{u}) \, dS + \int_{\mathcal{I}_i} \tau v u \, dS &= \int_{\Omega_i} v f \, dV + \int_{\mathcal{I}_i} \tau v u_{\mathcal{I}} \, dS \\ \int_{\Omega_i} \boldsymbol{q} \cdot \boldsymbol{w} \, dV - \int_{\Omega_i} \nu u \boldsymbol{\nabla} \cdot \boldsymbol{w} \, dV + \int_{\partial\Omega_i \backslash \mathcal{I}_i} \nu \widehat{u} \boldsymbol{w} \cdot \boldsymbol{n} \, dS &= - \int_{\mathcal{I}_i} \nu u_{\mathcal{I}} \boldsymbol{w} \cdot \boldsymbol{n} \, dS \end{aligned}$$

for all $v \in \mathcal{P}_k(\Omega_i)$ and $\boldsymbol{w} \in [\mathcal{P}_k(\Omega_i)]^d$. Compared to the weak form for standard elements (1.9), the X-HDG weak form for a cut element has three additional terms (one on the l.h.s., and two on the r.h.s. involving the data $u_{\mathcal{I}}$) corresponding to integrals along the interior boundary $\mathcal{I}_i$.

The discretization of the local problem on a cut element leads now to a system of equations of the form

$$\begin{cases} \left[\mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i}\right]\mathbf{u}^i & + \mathbf{A}_{uq}^{\Omega_i}\mathbf{q}^i + \mathbf{A}_{u\widehat{u}}^{\Omega_i}\mathbf{\Lambda}^i = \mathbf{f}^{\Omega_i} + \mathbf{f}_u^{\mathcal{I}_i} \\ \mathbf{A}_{qu}^{\Omega_i}\mathbf{u}^i & + \mathbf{A}_{qq}^{\Omega_i}\mathbf{q}^i + \mathbf{A}_{q\widehat{u}}^{\Omega_i}\mathbf{\Lambda}^i = \mathbf{f}_q^{\mathcal{I}_i} \end{cases},$$

similar to (1.11), but with an additional matrix $\mathbf{A}_{uu}^{\mathcal{I}_i}$ and two additional force vectors.

Now, this system can be solved for $\mathbf{u}^i$ and $\mathbf{q}^i$, obtaining the *local solver* in the cut element K_i, i.e. equation (1.13) with

$$\begin{bmatrix} \mathbf{U}^{K_i} \\ \mathbf{Q}^{K_i} \end{bmatrix} = -\mathbb{A}^{-1}\begin{bmatrix} \mathbf{A}_{u\widehat{u}}^{\Omega_i} \\ \mathbf{A}_{q\widehat{u}}^{\Omega_i} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{f}_U^{K_i} \\ \mathbf{f}_Q^{K_i} \end{bmatrix} = -\mathbb{A}^{-1}\begin{bmatrix} \mathbf{f}_u^{\Omega_i} + \mathbf{f}_u^{\mathcal{I}_i} \\ \mathbf{f}_q^{\mathcal{I}_i} \end{bmatrix}$$

and

$$\mathbb{A} = \begin{bmatrix} \left[\mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i}\right] & \mathbf{A}_{uq}^{\Omega_i} \\ \mathbf{A}_{qu}^{\Omega_i} & \mathbf{A}_{qq}^{\Omega_i} \end{bmatrix}.$$

Finally the global HDG problem is formed completely and analogously to Section 2.1.3.

Remark 10. The extension of X-HDG for problems with both Neumann and Dirichlet interfaces is straightforward, just considering the local problem stated in section 2.1.2 for elements cut by a Neumann interface, and the local problem in this section for elements cut by a Dirichlet interface.

2.3 Numerical tests

The performance of the novel X-HDG method is tested on three numerical examples. A Laplace equation with known analytical solution is solved over a square domain with a circular void. Boundary condition on the circular boundary are of Neumann type in the first example, and of Dirichlet type in the second example. The accuracy and the convergence of X-HDG is tested and compared to HDG with a mesh adapted to the void boundary.

Eventually, a potential flow, or flow through a porous medium, is modeled using the proposed X-HDG method. Impermeable stones are modeled as voids in the computational domain, with homogeneous Neumann boundaries. In this case, the

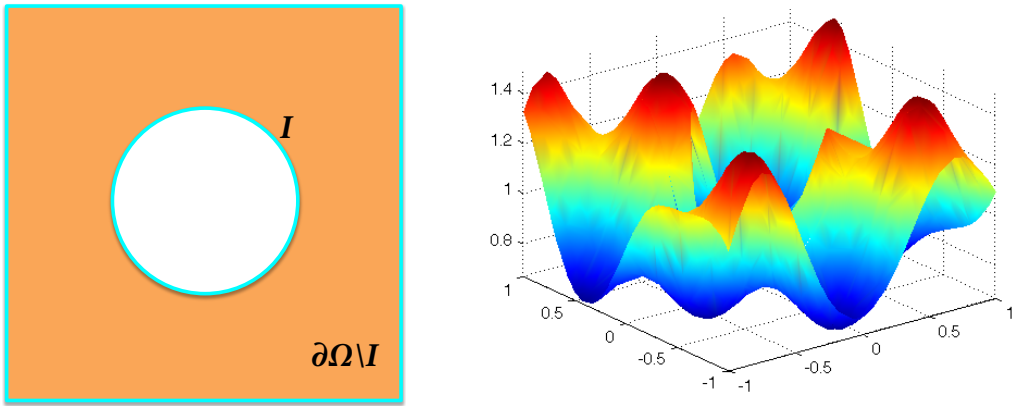


Figure 2.4: Circular void example: domain Ω and void boundary $\mathcal{I}$, and analytical solution.

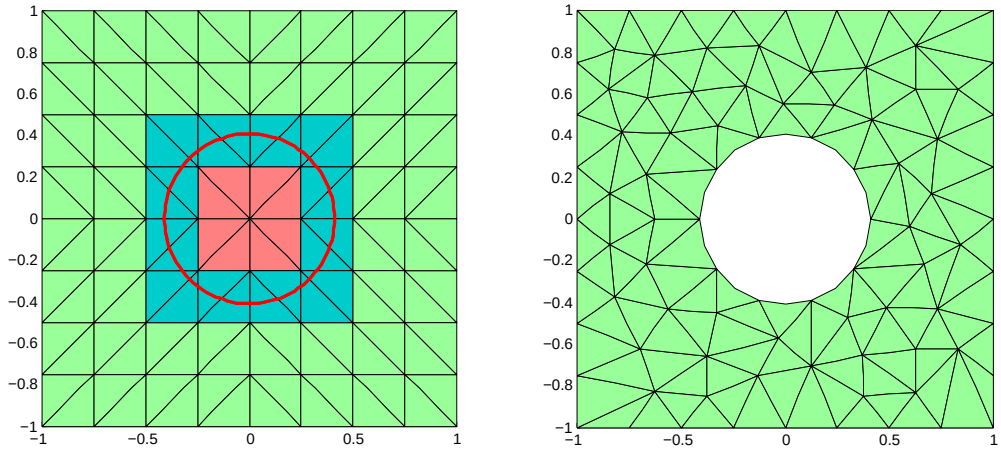


Figure 2.5: Circular void example: computational mesh for X-HDG and for standard HDG. The X-HDG mesh is not adapted to the void boundary. Elements in the interior of the domain are colored in dark gray. Elements in softer gray are elements cut by the interface $\mathcal{I}$. Elements in white are inside the void, and are not considered in the computation.

analytical solution is unknown and comparison with standard HDG is done with respect to a highly resolved reference solution on a fitted mesh.

In all numerical tests, the stabilization parameter is $\tau = 1$ on all faces.

2.3.1 Neumann void on a square domain

The first numerical example is designed to test the performance of X-HDG with Neumann boundary conditions imposed at the void interface. The Laplace equation (2.1), with $\nu = 1$, is solved over a square domain with a centred circular void with radius 0.41, $\Omega = (-1, 1)^2 \backslash B((0, 0), 0.41)$. Neumann boundary conditions are imposed

on the void boundary $\mathcal{I} = \partial B((0,0), 0.41)$, and Dirichlet boundary conditions are imposed on the exterior boundary $\partial\Omega\backslash\mathcal{I} = \partial\left((0,1)^2\right)$. Dirichlet and Neumann values and the source term f are set so that the analytical solution is

$$u(x,y) = \exp\big(0.1\sin(5.1x - 6.2y) + 0.3\cos(4.3x + 3.4y)\big).$$

Figure 2.4 shows the domain and the analytical solution.

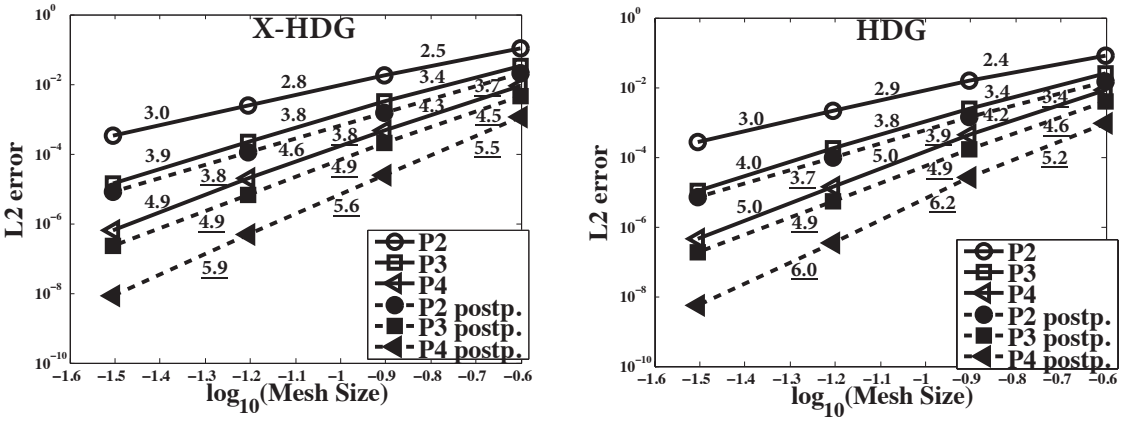


Figure 2.6: NEUMANN circular void example: convergence plots for X-HDG (left) and HDG (right). The numbers correspond to the slope of each segment, and they are underlined for postprocessed solution.

Figure 2.5 shows the computational mesh for X-HDG and for standard HDG. For the X-HDG computation, a regular triangular mesh in the square domain $(0,1)^2$ is considered, covering the domain Ω and fitting the exterior boundary $\partial\Omega\backslash\mathcal{I}$. A level set function is used to describe the boundary of the void, $\mathcal{I}$. Three kinds of elements appear in the computational mesh. Elements inside the domain (dark gray) are treated as standard HDG elements. For elements cut by the interior boundary $\mathcal{I}$ (light gray) the modified X-HDG local problem is considered, see section 2.1.2. The elements that are totally inside the void (white) have no contribution to the solution, so they are simply disregarded. The computational mesh for standard HDG is adapted to fit the void boundary, with similar uniform mesh size.

Figure 2.6 shows the evolution of the $\mathcal{L}_2$ error for decreasing uniform mesh size, for X-HDG and HDG, with degree $k = 2, 3, 4$, for both the solution and the postprocessed superconvergent solution, see Remark 18. X-HDG keeps HDG optimal converge with rates close to $k+1$ for the solution, and $k+2$ for the postprocessed solution, with similar levels of accuracy, while getting rid of adapting the mesh to the void boundary.

Remark 11. To ensure $k+2$ convergence rate of the postprocessed solution, a $k+1$ degree approximation of the interface should be considered. For standard HDG this is equivalent to requiring a $k+1$ degree mesh, properly fitting the description of the boundary, for the superconvengence postprocessing.

2.3.2 Dirichlet void on a square domain

In this second example, Dirichlet boundary conditions are imposed on the void boundary $\mathcal{I}$, testing the performance of X-HDG for Dirichlet interfaces. The domain, boundary conditions on exterior boundary and source term definitions are the same as for the first example, with same analytical solution.

Following the X-HDG methodology for Dirichlet voids explained in Section 2.2, the numerical solution is obtained for degree $k = 2, 3, 4$ for different mesh sizes. Like in the Neumann case, convergence plots for X-HDG and standard HDG (with mesh non-adapted and adapted to the circular boundary respectively, see Figure 2.5) are presented in Figure 2.7. Again, one to one resemblance is observed, keeping optimal HDG convergence properties with an unfitted computational mesh.

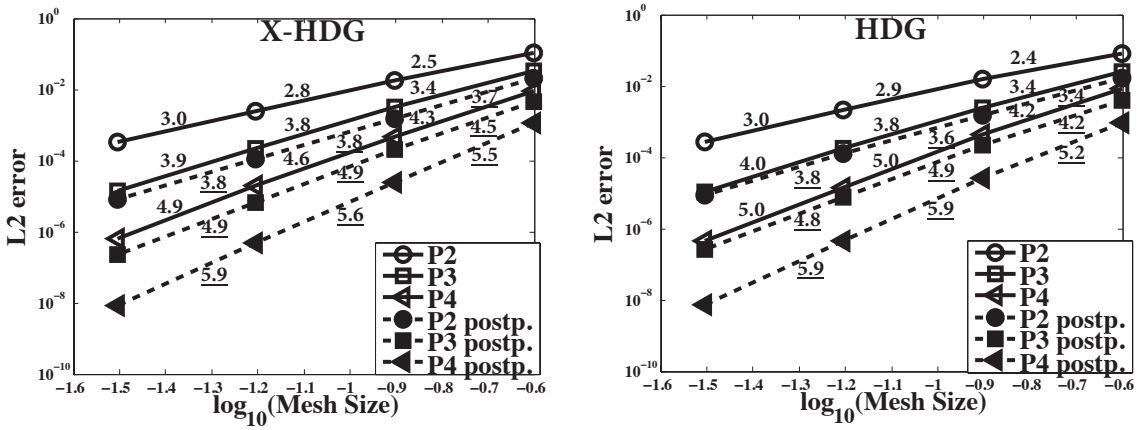


Figure 2.7: DIRICHLET circular void example: convergence plots for X-HDG (left) and HDG (right). The numbers correspond to the slope of each segment, and they are underlined for postprocessed solution.

2.3.3 Potential Flow example

A real-life example is considered to study the performance of X-HDG: a potential flow problem. The domain is the rectangle $(0, 10) \times (-3, 3)$ with several voids with boundaries $\mathcal{I}$, see Figure 2.8. The voids correspond to three circles with center and

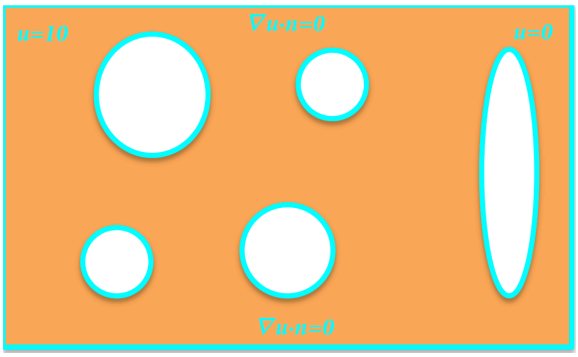


Figure 2.8: Domain for the potential flow example. Homogeneus Neumann boundary conditions are imposed on the voids interfaces $\mathcal{I}$ and on the top an bottom boundaries. Dirichlet boundary conditions driving the flow are imposed at left and right sides.

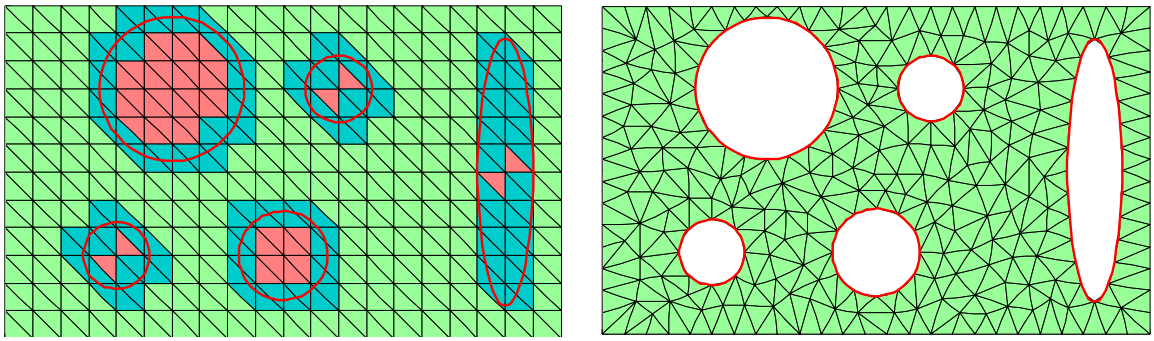


Figure 2.9: Flow example: computational mesh for X-HDG (left) and HDG (right).

radius $\{(2,-1.5),0.6\}$, $\{(3,1.5),1.3\}$ and $\{(5,-1.5),0.8\}$, and an ellipsoid centered at $(9,0)$ with x-radius 0.5 and y-radius 2.4. Homogeneous Neumann boundary conditions are set at the interior boundaries $\mathcal{I}$, modeling for instance impermeable rocks in the domain, and also at the top and bottom boundaries, $\{y=-3\}$ and $\{y=3\}$, to avoid leakage. The flow is driven by setting the potential to $u=10$ at the inflow boundary $\{x=0\}$ and $u=10$ at the outflow boundary $\{x=10\}$. The source term in this test is $f=0$, and the viscosity is set to $\nu=1$.

A structured triangle mesh in the rectangle $(0,10)\times(-3,3)$ is considered for the X-HDG solution, representing the voids boundaries $\mathcal{I}$ with a level-set function. For comparison, an adapted mesh with similar mesh size is considered for a standard HDG solution. Both computational meshes are shown in Figure 2.9.

In this example the output of interest is the velocity field $\boldsymbol{q}$. Figure 2.10 shows the velocity field and the streamlines for a computation with degree $k=3$ for HDG, with the mesh fitting to the boundary, and for X-HDG, with the structured unfitted mesh.

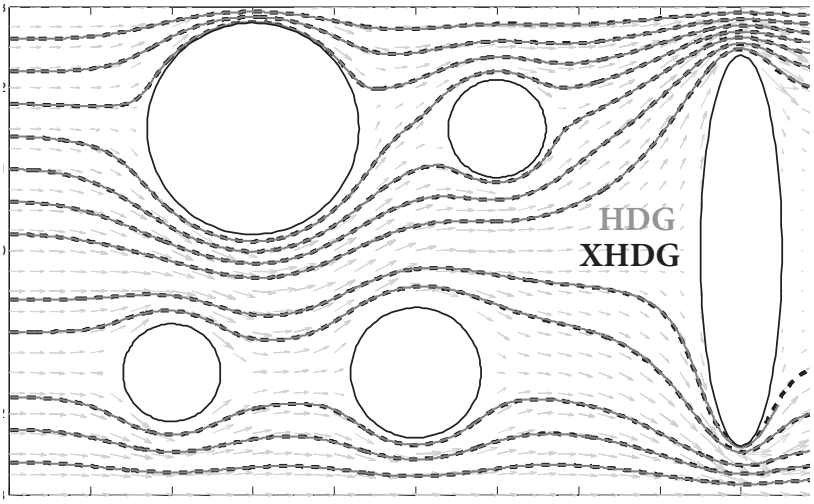


Figure 2.10: Potential flow example: streamlines for a degree $k = 3$ computation with HDG (solid gray line) and X-HDG (dashed black line).

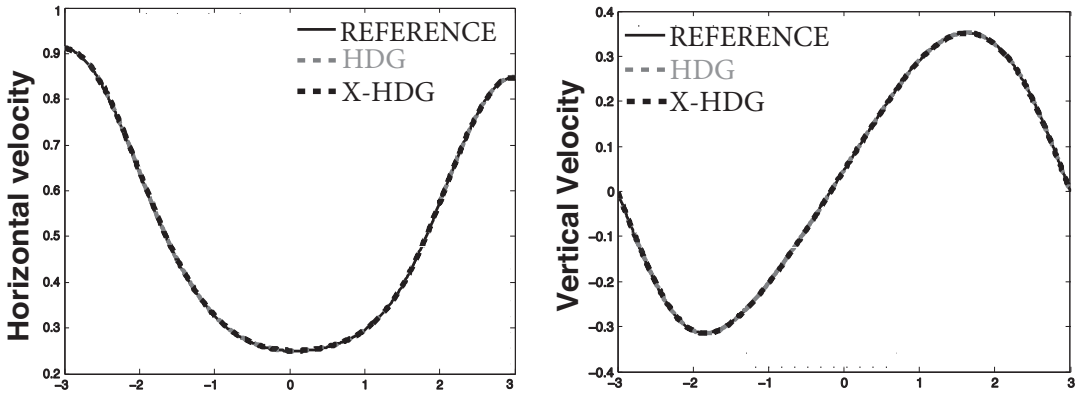


Figure 2.11: Potential flow example: horizontal (left) and vertical (right) components of the velocity along the vertical line $\{x = 7.7\}$, for HDG, X-HDG and a reference solution, with degree $k = 3$.

The streamlines for HDG and X-HDG overlap, demonstrating the applicability and good performance of the X-HDG method. To further compare the X-HDG and HDG solutions, the horizontal and vertical component of the velocity along the vertical line $\{x = 7.7\}$ are plotted in Figure 2.11. No analytical solution is available for this problem; thus, a standard HDG reference solution computed on an adapted highly resolved computational mesh is considered as reference solution. Again, both HDG and X-HDG provide accurate results. The error of the HDG and the X-HDG solution along $\{x = 7.7\}$ when compared to the reference solution is plotted in Figure 2.12. Similar accuracy, with errors of order 10^{-3}, are obtained with both standard HDG and X-HDG with similar mesh size, demonstrating again the reliability of X-HDG.

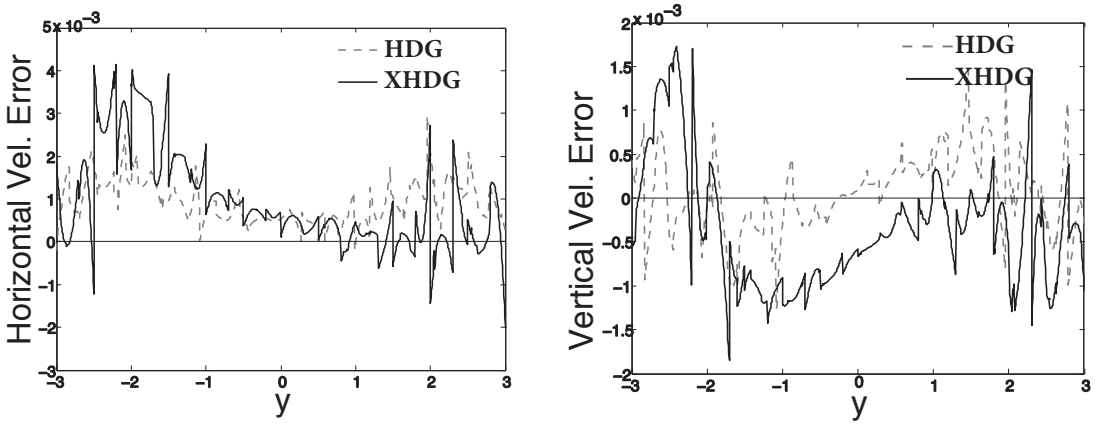


Figure 2.12: Potential flow example: error of the HDG and the X-HDG solution along $\{x = 7.7\}$ when compared to the reference solution, for the horizontal component (left) and vertical component (right) of the velocity.

These results assert that X-HDG keeps the accuracy of HDG without the need to adapt the mesh to the boundary.

2.4 Conclusions and final remarks

Detailed X-HDG formulation for problems including voids has been presented. With X-HDG, the computational mesh does not need to be adapted to the interface (i.e the boundary) simplifying and reducing the cost of mesh generation.

Differently to previous proposals for HDG solution with meshes non-fitting the boundary see Cockburn et al. [2014], Cockburn and Solano [2014], here the computational mesh covers the domain, avoiding extrapolations, and ensuring the robustness of the method. The local problem at elements not cut by the interface is discretized as usual in HDG see Section 1.3.1. A modified local problem on the other hand, is considered at elements cut by the interface. At every cut element, an auxiliary trace variable on the boundary is introduced, which is eliminated afterwards using the boundary conditions on the interface, keeping the original unknowns and the structure of the local problem solver. The local solver obtained from standard and cut elements then plugged into discretized global equations to end up with a global system that only has the trace variable $\widehat{u}$ as an unknown. For numerical integration in cut elements, in the context of high-order approximations, the methodology presented in Section 1.3.3 is used. Although this modified numerical integration in cut elements causes slight increase in CPU time, in practical applications the extra cost

becomes negligible compared with the cost of mesh adaptation or re-meshing. At the end X-HDG is compared to standard HDG in three numerical tests: the solution of a Laplace problem with known analytical solution in a square domain with a circular void, with Dirichlet and Neumann boundary conditions, and the computation of the velocity field for a potential flow problem in a rectangular domain with several voids. The HDG computations are done on a mesh fitting the voids boundaries, whereas with X-HDG a regular mesh covering the domain can be considered. Numerical tests assert that X-HDG keeps the HDG optimal convergence rates for the solution, the gradient and the post-processed super-convergent solution, without the need to adapt the mesh to the boundary. In all tests, similar accuracy is observed for similar mesh size.

The X-HDG method is fully developed in this chapter for the solution of steady void problems, describing the main ideas and fundamentals of the method. In the coming Chapter 3 the method is going to be further explored for bimaterial problems.

Chapter 3

X-HDG for Second Order Elliptic Bimaterial Problems

X-HDG for bimaterial problems is presented next. In section 3.1 the problem definition and weak forms are proposed. Numerical examples demonstrate the applicability of X-HDG, see section 3.2. The chapter is concluded with final remarks presented in section 3.3. The findings presented in this chapter can also be found in Gürkan et al. [2017]

3.1 X-HDG for bimaterial problems

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain divided in two disjoint subdomains

$$\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2, \quad \Omega_1 \cap \Omega_2 = \emptyset$$

with an interface

$$\mathcal{I} = \overline{\Omega}_1 \cap \overline{\Omega}_2.$$

The following bimaterial problem is considered,

$$\begin{aligned} -\boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} u) &= f \quad \text{in } \Omega_1 \cup \Omega_2, \\ [\![u\boldsymbol{n}]\!] &= \mathbf{0} \quad \text{on } \mathcal{I}, \\ [\![\nu \boldsymbol{n} \cdot \boldsymbol{\nabla} u]\!] &= 0 \quad \text{on } \mathcal{I}, \\ u &= u_D \quad \text{on } \Gamma_D, \\ -\nu \boldsymbol{n} \cdot \boldsymbol{\nabla} u &= g_N \quad \text{on } \Gamma_N \end{aligned} \tag{3.1}$$

where u is the solution, ν is a material coefficient with discontinuous definition across the interface (that is, $\nu = \nu_i$ in Ω_i for $i = 1, 2$), f is a given source term and u_D are prescribed values on the boundary. The *jump* $[\![\cdot]\!]$ operator is defined as in (1.7) that is, for instance, at the material interface $\mathcal{I}$ between the subdomains Ω_1 and Ω_2, the jump of a vector $\boldsymbol{w}$ is $[\![\boldsymbol{w} \cdot \boldsymbol{n}]\!] := \boldsymbol{w}_1 \cdot \boldsymbol{n}_1 + \boldsymbol{w}_2 \cdot \boldsymbol{n}_2 = (\boldsymbol{w}_1 - \boldsymbol{w}_2) \cdot \boldsymbol{n}_1$, where $\boldsymbol{n}_i$ is the normal vector exterior to Ω_i, and $\boldsymbol{w}_i$ denotes the restriction of $\boldsymbol{w}$ to Ω_i.

Here, for simplicity, Dirichlet boundary conditions are considered on the boundary $\partial\Omega^{ext}$; other boundary conditions on $\partial\Omega^{ext}$ do not add any difficulty.

The domain Ω is now assumed to be covered by a finite element mesh with $\mathtt{n_{el}}$ disjoint elements K_i, and a mesh skeleton Γ, satisfying (1.2) and (1.3). In the discontinuous setting, equation (3.1) is expressed as a first order system with some equations local to the elements and some global equations. The local element-by-element problems correspond to the statement of the PDE in (3.1), with Dirichlet boundary conditions, in each element K_i, that is,

$$\left.\begin{aligned} \boldsymbol{\nabla} \cdot \boldsymbol{q} &= f \quad \text{in } K_i \\ \boldsymbol{q} + \nu \boldsymbol{\nabla} u &= 0 \quad \text{in } K_i \\ u &= \widehat{u} \quad \text{on } \partial K_i \end{aligned}\right\} \text{ if } \mathcal{I} \cap K_i = \emptyset, \tag{3.2a}$$

$$\left.\begin{aligned} \boldsymbol{\nabla} \cdot \boldsymbol{q} &= f \quad \text{in } K_i \backslash \mathcal{I} \\ \boldsymbol{q} + \nu \boldsymbol{\nabla} u &= 0 \quad \text{in } K_i \backslash \mathcal{I} \\ [\![u\boldsymbol{n}]\!] &= 0 \quad \text{on } \mathcal{I} \cap K_i \\ [\![\boldsymbol{q} \cdot \boldsymbol{n}]\!] &= 0 \quad \text{on } \mathcal{I} \cap K_i \\ u &= \widehat{u} \quad \text{on } \partial K_i \end{aligned}\right\} \text{ if } \mathcal{I} \cap K_i \neq \emptyset, \tag{3.2b}$$

for $i = 1, \ldots, \mathtt{n_{el}}$. Two new variables are introduced: $\boldsymbol{q}$ corresponding to the flux of u, splitting the PDE into two first order PDEs, and $\widehat{u}$ corresponding to the trace of u at the mesh faces Γ. The trace $\widehat{u}$ is a single valued variable on each face, with the same value when seen from both sides of an interior face. Figure 3.1 shows an example of an HDG computational mesh, with elemental nodes and trace nodes. The local problems have been particularized for elements cut by the interior boundary (3.2b), including the interface conditions, and for standard elements (3.2a). Given the trace $\widehat{u}$, the local problems (3.2) can be solved in each element to determine the solution u and the flux $\boldsymbol{q}$. Thus, the problem now reduces to determine the trace $\widehat{u}$. This is done, as usual, by imposing the *conservativity conditions* (i.e. *global equations*), that

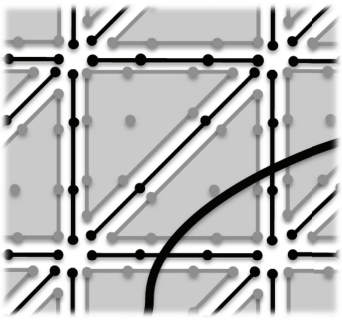


Figure 3.1: Example of a third degree HDG discretization. Nodal approximation at elements (gray nodes) for u and $\boldsymbol{q}$, and nodal approximation at sides (black nodes) for the trace $\widehat{u}$. Some elements and sides are cut by the interface.

is, the continuity of the flux across element boundaries

$$[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!]=0 \quad \text{on } \Gamma\backslash\partial\Omega, \tag{3.3}$$

and the boundary condition, equivalent to the boundary condition in (3.1),

$$\widehat{u}=\mathbb{P}_2(u_D) \text{ on } \Gamma_D, \quad \boldsymbol{q}\cdot\boldsymbol{n}=g_N \text{ on } \Gamma_N, \tag{3.4}$$

where $\mathbb{P}_2(u_D)$ is the $\mathcal{L}_2$ projection of the data u_D on the approximation space on the faces, i.e., a least-squares fitting.

The discretization of the conservativity condition (3.3) and the local problems (3.2), with the boundary condition (3.4), leads to the X-HDG formulation. The following discrete spaces for elemental variables, u and $\boldsymbol{q}$, and for the trace variable, $\widehat{u}$, are considered

$$\begin{aligned}
\mathcal{V}^h_{enr} := \big\{ v\in\mathcal{L}_2(\Omega) \;:\; & v|_{K_i}\in\mathcal{P}_k(K_i) && \text{if } K_i\cap\mathcal{I}=\emptyset,\\
& v|_{K_i}\in\mathcal{P}_k(K_i)\oplus H\mathcal{P}_k(K_i) && \text{if } K_i\cap\mathcal{I}\neq\emptyset\big\},\\
\Lambda^h_{enr} := \big\{ \hat{v}\in\mathcal{L}_2(\Gamma) \;:\; & \hat{v}|_{\Gamma_i}\in\mathcal{P}_k(\Gamma_i) && \text{if } \Gamma_i\cap\mathcal{I}=\emptyset,\\
& \hat{v}|_{\Gamma_i}\in\mathcal{P}_k(\Gamma_i)\oplus H\mathcal{P}_k(\Gamma_i) && \text{if } \Gamma_i\cap\mathcal{I}\neq\emptyset\big\},
\end{aligned} \tag{3.5}$$

where $\mathcal{P}_k$ denotes the space of polynomials of degree less or equal to k. H is a Heaviside function enriching the approximation in cut elements and on cut faces, which can be defined, for instance, as

$$H=\begin{cases} 1 & \text{in } \Omega_1\\ -1 & \text{in } \Omega_2 \end{cases}$$

to introduce discontinuities across the interface $\mathcal{I}$.

Remark 12. The X-HDG the approximation space is enriched with a Heaviside function and, therefore, the approximations are discontinuous across the interface, for both the solution u and the flux $\boldsymbol{q}$. It is a natural choice for the enrichment of the flux $\boldsymbol{q}$, whose tangential component is discontinuous across the interface. For the solution u, even though it is continuous, X-HDG considers a discontinuous approximation to keep the discontinuous setting of HDG also across the interface. Continuity of the solution u and the normal flux $\boldsymbol{q} \cdot \boldsymbol{n}$ across the interface is imposed in weak form, as usually done across interior faces in HDG. In fact, the X-HDG method proposed here is formally equivalent to a standard HDG method applied on a cut mesh combining triangular and quadrilateral elements with a $\mathcal{P}_k$ polynomial approximation (i.e., a complete k-th degree polynomial basis also for quadrilaterals), but organized and implemented in an alternative way to keep the original triangular computational mesh and the original unknown structure, as usual in X-FEM methods. Thus, X-HDG keeps the superconvergence and stability properties of standard HDG, but in accordance with an X-FEM philosophy.

Remark 13. For the sake of simplicity, all derivations in this section are done assuming that cut elements are split in two regions only, which corresponds to the usual situation. Thus, one Heaviside enrichment is considered, see (3.5). However, with high-order approximations more complicated situations, dividing elements or faces into more regions, may appear, see section 3.1.4 and the numerical example in section 3.2.5. If it is the case, an enrichment with multiple Heaviside functions is proposed in section 3.1.4 to ensure an independent approximation in each region of the element or face.

Remark 14. The description of the interface can be done with a level set function as usual in X-FEM methods, see for instance Gürkan et al. [2016]. The level set function is given by its nodal values on the computational mesh using an r-th degree approximation. In most numerical tests a k-th degree level set function is used for the X-HDG solution and an additional level set function with polynomial degree $k+1$ is used for the postprocessed solution. Nevertheless, for examples with evolving interfaces and more involved interface geometries an accurate representation of the interface is crucial to keep the X-HDG accuracy, see the kidney-shaped interface example in section 3.2.4. According to the analysis for standard HDG with meshes fitted to the interface in Huynh et al. [2013], a level set function with degree $r = 2k+1$ ensures optimal convergence of X-HDG in the more general situation.

The next sections present the details of the X-HDG formulation, stating the discretization of the local problems for standard and cut elements and the discretization of the conservativity condition or, in other words, the global equations.

3.1.1 Local problem for standard elements

The local problem at elements not cut by the interface (3.2a) is discretized as usual in HDG, see Cockburn et al. [2009, 2008], as recalled in section 1.3.1. The discretization of the local problem for cut elements (3.2b) is developed in coming section 3.1.2.

3.1.2 Local problem for cut elements

The X-HDG local problem on an element K_i cut by the interface corresponds to the discretization of (3.2b), that is: given $\widehat{u} \in \Lambda^h_{enr}$, find $u \in \mathcal{P}_k(K_i) \oplus H\mathcal{P}_k(K_i)$, $\boldsymbol{q} \in [\mathcal{P}_k(K_i) \oplus H\mathcal{P}_k(K_i)]^d$ such that

$$\begin{aligned} \int_{K_i} v\boldsymbol{\nabla}\cdot\boldsymbol{q}\,dV + \int_{\partial K_i} \tau\nu\, v(u-\widehat{u})\,dS + 2\int_{\mathcal{I}_i} \{\tau\nu v(u-\widetilde{u})\}\,dS = \int_{K_i} vf\,dV \\ \int_{K_i} \frac{1}{\nu}\boldsymbol{q}\cdot\boldsymbol{w}\,dV - \int_{K_i} u\boldsymbol{\nabla}\cdot\boldsymbol{w}\,dV + \int_{\partial K_i} \widehat{u}\boldsymbol{w}\cdot\boldsymbol{n}\,dS + \int_{\mathcal{I}_i} \widetilde{u}[\![\boldsymbol{w}\cdot\boldsymbol{n}]\!]\,dS = 0 \end{aligned} \tag{3.6}$$

for all $v \in \mathcal{P}_k(K_i) \oplus H\mathcal{P}_k(K_i)$ and $\boldsymbol{w} \in [\mathcal{P}_k(K_i) \oplus H\mathcal{P}_k(K_i)]^d$, where $\widetilde{u}$ is a new trace variable approximating the trace of the solution on the interface $\mathcal{I}_i := K_i \cap \mathcal{I}$. Note that the jump in ν, v and u across $\mathcal{I}_i$ is taken into account by a mean operation in the third term of the first equation. The X-HDG local problem (3.6) can be derived following the standard HDG rationale for each one of the domains, $K_i \cap \Omega_1$ and $K_i \cap \Omega_2$, and summing the obtained weak forms. Compared to the weak form for standard elements (1.9), the X-HDG weak form for a cut element has two additional terms corresponding to integrals along the interface $\mathcal{I}_i$. They involve the new trace variable $\widetilde{u}$ and take the discontinuous nature of the approximation space into account. These new terms weakly impose the condition $u = \widetilde{u}$ on $\mathcal{I}_i$, ensuring weak continuity of u across the interface.

Remark 15. The approximation space $\mathcal{P}_k(I_i)$ is represented by a one-dimensional polynomial nodal basis of degree k on the 1D reference element $(-1, 1)$, which is mapped to a nodal basis on an approximation of the interface $\mathcal{I}_i$. The mapping is done with an r-th degree parametrization given by $r+1$ points on the interface. The

r-th degree parametrization of the interface is also used for the definition of numerical quadrature schemes within the cut element, accounting for the discontinuous nature of the approximation across the interface, see Cheng and Fries [2010], Gürkan et al. [2016] for details, and an example with degree $r = 3$ of the level set function in Figure 3.2. With the proposed approach, the integrals on both sides of the interface are computed separately.

The discretization of the local problem (3.6) leads to a system of equations of the form

$$\begin{aligned} \left[\mathbf{A}_{uu}^{K_i} + \mathbf{A}_{uu}^{\mathcal{I}_i}\right] \mathbf{u}^i \quad &+\mathbf{A}_{uq}^{K_i}\mathbf{q}^i + \mathbf{A}_{u\widehat{u}}^{K_i}\mathbf{\Lambda}^i + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\widetilde{\boldsymbol{u}} = \mathbf{f}_u^{K_i} \\ \mathbf{A}_{qu}^{K_i}\mathbf{u}^i \quad &+\mathbf{A}_{qq}^{K_i}\mathbf{q}^i + \mathbf{A}_{q\widehat{u}}^{K_i}\mathbf{\Lambda}^i + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\widetilde{\boldsymbol{u}} = \mathbf{0} \end{aligned}, \tag{3.7}$$

similar to (1.11), but with three new matrices corresponding to integrals on the interface $\mathcal{I}_i$ (marked with the superindex $\mathcal{I}_i$), and the nodal values for the new trace variable, $\widetilde{\boldsymbol{u}}$.

The local problem is now closed by imposing the conservativity condition across the interface $\mathcal{I}_i$,

$$[\![\widehat{\boldsymbol{q}} \cdot \boldsymbol{n}]\!] = 0 \quad \text{on } \mathcal{I}_i. \tag{3.8}$$

Replacing the expression of the numerical flux $\widehat{\boldsymbol{q}}$ defined in (1.10), i.e., $\widehat{\boldsymbol{q}} \cdot \boldsymbol{n} = \boldsymbol{q} \cdot \boldsymbol{n} + \nu\tau(u - \widetilde{u})$, the weak form of the conservativity condition on $\mathcal{I}^i$ is: given $u \in \mathcal{P}_k(K_i) \oplus H\mathcal{P}_k(K_i)$, $\boldsymbol{q} \in [\mathcal{P}_k(K_i) \oplus H\mathcal{P}_k(K_i)]^d$, find $\widetilde{u} \in \mathcal{P}_k(\mathcal{I}_i)$ such that

$$\int_{\mathcal{I}_i} \widetilde{v}[\![\boldsymbol{q} \cdot \boldsymbol{n}]\!]\, dS + 2\int_{\mathcal{I}_i} \widetilde{v}\left(\{\nu\tau u\} - \{\nu\tau\}\widetilde{u}\right) dS = 0 \quad \forall \widetilde{v} \in \mathcal{P}_k(\mathcal{I}_i). \tag{3.9}$$

The discretization of the conservativity condition leads to a discrete equation of the form

$$\mathbf{A}_{\widetilde{u}u}^{\mathcal{I}_i}\mathbf{u}^i + \mathbf{A}_{\widetilde{u}q}^{\mathcal{I}_i}\mathbf{q}^i + \mathbf{A}_{\widetilde{u}\widetilde{u}}^{\mathcal{I}_i}\widetilde{\boldsymbol{u}} = 0.$$

This allows expressing the new trace values $\widetilde{\boldsymbol{u}}$ in terms of the elemental values

$$\widetilde{\boldsymbol{u}} = \mathbf{T}_u^i\mathbf{u}^i + \mathbf{T}_q^i\mathbf{q}^i \tag{3.10}$$

with $\mathbf{T}_u^i = -[\mathbf{A}_{\widetilde{u}\widetilde{u}}^{\mathcal{I}_i}]^{-1}\mathbf{A}_{\widetilde{u}u}^{\mathcal{I}_i}$, $\mathbf{T}_q^i = -[\mathbf{A}_{\widetilde{u}\widetilde{u}}^{\mathcal{I}_i}]^{-1}\mathbf{A}_{\widetilde{u}q}^{\mathcal{I}_i}$. This expression corresponds to a Schur complement for eliminating the trace solution $\widetilde{u}$. This elimination of a trace variable $\widetilde{u}$ in favor of the local variables, u and q, is logically the opposite elimination as the one expressed by the local solver. Substituting $\widetilde{\boldsymbol{u}}$ in (3.7) by (3.10) leads to the final

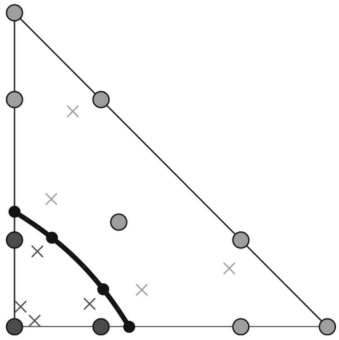


Figure 3.2: Interface representation and numerical quadrature with 3-rd degree parametrization of the interface. The color on the nodes refer to the sign of the level set function defining the interface. Crosses are exemplary integration points.

discrete local problem

$$\begin{cases} \left[\mathbf{A}_{uu}^{K_i} + \mathbf{A}_{uu}^{\mathcal{I}_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right]\mathbf{u}^i & + \left[\mathbf{A}_{uq}^{K_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_q^i\right]\mathbf{q}^i + \mathbf{A}_{u\widehat{u}}^{K_i}\mathbf{\Lambda}^i = \mathbf{f}_u^{K_i} \\ \left[\mathbf{A}_{qu}^{K_i} + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right]\mathbf{u}^i & + \left[\mathbf{A}_{qq}^{K_i} + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_q^i\right]\mathbf{q}^i + \mathbf{A}_{q\widehat{u}}^{K_i}\mathbf{\Lambda}^i = \mathbf{0}. \end{cases} \tag{3.11}$$

Now, system (3.11) can be solved for $\mathbf{u}^i$ and $\mathbf{q}^i$, obtaining the *local solver* in the cut element K_i, i.e., equation (1.13) with (1.14) and

$$\mathbb{A} = \begin{bmatrix} \left[\mathbf{A}_{uu}^{K_i} + \mathbf{A}_{uu}^{\mathcal{I}_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{uq}^{K_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_q^i\right] \\ \left[\mathbf{A}_{qu}^{K_i} + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{qq}^{K_i} + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_q^i\right] \end{bmatrix}.$$

The structure of the local solver is exactly the same as for non-cut elements (1.13), thanks to the fact that the internal trace variable $\widetilde{u}$ has been expressed in terms of the local variables. Note, however, that the size of the matrices involved in the local solver is different for a standard element and for a cut element due to the enriched approximation. In the simplest case (see Remark 13), there are twice as many degrees of freedom for cut elements and faces.

The discontinuous nature of the approximation across the interface necessitates a proper modified numerical quadrature for the integration in cut elements and cut faces. Adopting the strategy presented in 1.3.3, the integrals on both sides of the interface are computed separately, see Figure 3.2.

Remark 16. The extension of the formulation to other conditions on the interface is straight-forward. As an example, consider a problem with non-homogeneous conditions on the interface, $u_2 - u_1 = \alpha$ and $[\![-\nu\boldsymbol{n}\cdot\boldsymbol{\nabla}u]\!] = g$ on $\mathcal{I}_i$, where u_i denotes the

restriction of function u to Ω_i. Then, defining $\widetilde{u}^i$ as the trace of u_1 (i.e., $u_1 = \widetilde{u}^i$ on $\mathcal{I}_i$), we have $u_2 = \widetilde{u}^i + \alpha$ on $\mathcal{I}_i$, and therefore, new terms $\int_{\mathcal{I}_i} \tau\nu_2 v_2 \alpha$ and $-\int_{\mathcal{I}_i} \alpha \boldsymbol{w}_2 \cdot \boldsymbol{n}_2$ appear in the right hand side of the first and second equation in (3.6), respectively, and the right hand side of (3.9) becomes $\int_{\mathcal{I}_i} \widetilde{v} g \, dS + \int_{\mathcal{I}_i} \tau\nu_2 \widetilde{v}\alpha \, dS$.

3.1.3 Global problem

The local problem, both in a standard element or in a cut element, leads to the local solver (1.13) that expresses the solution in the element, u and $\boldsymbol{q}$, in terms of the trace values at its boundary, $\widehat{u}$. Thus, the problem is reduced to determine the trace nodal values $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{nfc}}$ on the mesh skeleton Γ. For this purpose the so-called *global problem* is stated, which corresponds to the discretization of the conservativity condition on Γ (3.3), replacing $\boldsymbol{q}$ by the numerical flux (1.10).

The weak form for the global problem is: find $\widehat{u} \in \Lambda^h$ such that $\widehat{u} = \mathbb{P}_2(u_D)$ on Γ_D and

$$\int_\Gamma \widehat{v} [\![\boldsymbol{q} \cdot \boldsymbol{n}]\!] \, dS + 2 \int_\Gamma \widehat{v} \left(\{\nu\tau u\} - \{\nu\tau\}\widehat{u} \right) \, dS = \int_{\Gamma_N} \widehat{v} g_N \, dS, \tag{3.12}$$

for all $\widehat{v} \in \Lambda^h$ with $\widehat{v} = 0$ on Γ_D. As usual in HDG, the discretization of (3.12) for every face Γ_f leads to an equation of the form

$$\mathbf{A}_{\widehat{u}u}^{f,L} \mathbf{u}^{L(f)} + \mathbf{A}_{\widehat{u}q}^{f,L} \mathbf{q}^{L(f)} \; + \; \mathbf{A}_{\widehat{u}u}^{f,R} \mathbf{u}^{R(f)}(f) + \mathbf{A}_{\widehat{u}q}^{f,R} \mathbf{q}^{R(f)} \; + \; \mathbf{A}_{\widehat{u}\widehat{u}}^{f} \widehat{\mathbf{u}}^f = \mathbf{g}^f. \tag{3.13}$$

Inserting the expressions of the local solver (1.13) for the elements $K_{L(f)}$ and $K_{R(f)}$ in (3.13) for every face Γ_f leads to a system of equations involving only the trace variables $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{nfc}}$.

As usual in an HDG code, the implementation of the method involves a loop over elements. For each element, the matrices and vectors for the local solver (1.13) are computed, and the contribution to the equation (3.13) is assembled for each face of the element. Once the system is assembled for all elements and Dirichlet boundary conditions (3.4) are imposed according to Remark 3, the system can be solved. Then, given the trace variables $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{nfc}}$, the solution, $\mathbf{u}^i$ and $\boldsymbol{q}^i$, can be computed for each element using (1.13).

It is important noting that X-HDG keeps the structure of a standard HDG code. The main differences are: (i) the modified local problem for cut elements (3.7), and the corresponding matrices in the local solver, (ii) the modified numerical integration in cut elements and on cut faces, (iii) the increased number of degrees of freedom for

the enriched approximation in cut elements and on cut faces, that has to be taken into account also for the assembly of the matrices involving cut faces.

Remark 17. Compared to standard HDG, the size of the system of equations to be solved for X-HDG is larger, because the number of degrees of freedom for cut faces doubles as compared to a standard case (in the simplest case, see Remark 13). However, in practical applications cut faces are usually a small portion of the whole set of faces.

Remark 18. Similarly to standard HDG, a second element-by-element postprocessing can be done to compute an X-HDG superconvergent solution. For a standard element, $K_i \cap \mathcal{I} = \emptyset$, the superconvergent approximation is computed as in standard HDG: find $u^* \in \mathcal{P}_{k+1}(K_i)$ such that

$$\int_{K_i} \nu \boldsymbol{\nabla} u^* \cdot \boldsymbol{\nabla} v \, dV = -\int_{K_i} \boldsymbol{q} \cdot \boldsymbol{\nabla} v \, dV \quad \forall v \in \mathcal{P}_{k+1}(K_i),$$
$$\int_{K_i} u^* \, dV = \int_{K_i} u \, dV.$$

For an element cut by the interface, $K_i \cap \mathcal{I} \neq \emptyset$, the superconvergent approximation is computed finding $u^* \in \mathcal{P}_{k+1}(K_i) \oplus H\mathcal{P}_{k+1}(K_i)$ such that

$$\int_{K_i} \nu \boldsymbol{\nabla} u^* \cdot \boldsymbol{\nabla} v \, dV = -\int_{K_i} \boldsymbol{q} \cdot \boldsymbol{\nabla} v \, dV \quad \forall v \in \mathcal{P}_{k+1}(K_i) \oplus H\mathcal{P}_{k+1}(K_i),$$
$$\int_{K_i} u^* \, dV = \int_{K_i} u \, dV \quad \text{and} \quad \int_{K_i} Hu^* \, dV = \int_{K_i} Hu \, dV.$$

The solution of this element-by-element computation, u^*, converges with order $k+2$ in the $\mathcal{L}_2$ norm.

3.1.4 Cut element split in more than two regions

As mentioned in Remark 13, in the most general situation some elements and faces may be split by the interface in more than two regions, requiring an enrichment with several Heaviside functions. For instance, for a triangle split in three regions, as shown in Figure 3.3, an approximation with two Heaviside functions should be considered to properly represent the discontinuity. The local problem in this case is: find $u \in \mathcal{P}_k(K_i) \oplus H^1\mathcal{P}_k(K_i) \oplus H^2\mathcal{P}_k(K_i)$, $\boldsymbol{q} \in [\mathcal{P}_k(K_i) \oplus H^1\mathcal{P}_k(K_i) \oplus H^2\mathcal{P}_k(K_i)]^d$

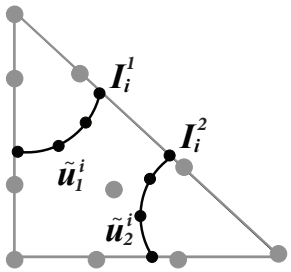


Figure 3.3: Triangle split by the interface in 3 regions, with two elemental interfaces $\mathcal{I}_i^1$ and $\mathcal{I}_i^2$, with corresponding auxiliary trace variables $\widetilde{u}_1$ and $\widetilde{u}_2$. The approximation is enriched with two Heaviside functions.

such that

$$\int_{K_i} v\boldsymbol{\nabla}\cdot\boldsymbol{q}\,dV + \int_{\partial K_i} \tau\nu\; v(u-\widehat{u})\,dS + 2\int_{\mathcal{I}_i^1}\{\tau\nu v(u-\widetilde{u}_1)\}\,dS + 2\int_{\mathcal{I}_i^2}\{\tau\nu v(u-\widetilde{u}_2)\}\,dS = \int_{K_i} vf\,dV$$

$$\int_{K_i}\frac{1}{\nu}\boldsymbol{q}\cdot\boldsymbol{w}\,dV - \int_{K_i} u\boldsymbol{\nabla}\cdot\boldsymbol{w}\,dV + \int_{\partial K_i}\widehat{u}\boldsymbol{w}\cdot\boldsymbol{n}\,dS + \int_{\mathcal{I}_i^1}\widetilde{u}_1[\![\boldsymbol{w}\cdot\boldsymbol{n}]\!]\,dS + \int_{\mathcal{I}_i^2}\widetilde{u}_2[\![\boldsymbol{w}\cdot\boldsymbol{n}]\!]\,dS = 0$$

for all $v \in \mathcal{P}_k(K_i)\oplus H^1\mathcal{P}_k(K_i)\oplus H^2\mathcal{P}_k(K_i)$ and $\boldsymbol{w} \in [\mathcal{P}_k(K_i)\oplus H^1\mathcal{P}_k(K_i)\oplus H^2\mathcal{P}_k(K_i)]^d$, where $\mathcal{I}_i^1$ and $\mathcal{I}_i^2$ are the interfaces splitting the element in three regions, H^1 and H^2 are Heaviside functions to represent the discontinuities on both interfaces, and $\widetilde{u}_1$ and $\widetilde{u}_2$ are new trace variables approximating the trace of the solution on $\mathcal{I}_i^1$ and $\mathcal{I}_i^2$, respectively. Following the rationale in section 3.1.2, the trace variables $\widetilde{u}_1$ and $\widetilde{u}_2$ are expressed in terms of u and $\boldsymbol{q}$ using the conservativity condition (3.8) with the corresponding weak form (3.9) on each interface. Inserting them into the local problem again leads to a local solver with the same structure as (1.13), expressing u and $\boldsymbol{q}$ in terms of the global trace variable $\widehat{u}$. For integrations, divide-and-conquer strategy presented in 1.3.3 is followed, which handles multiple cut situations easily.

Obviously, at faces that are split by the interface into more than two regions, an enrichment with several Heaviside functions must also be considered for $\widehat{u}$. A numerical example showing the performance of X-HDG when some elements are cut twice by the interface can be found in section 3.2.5.

We note that in a DG framework the use of different enrichment functions in neighboring elements does not imply any difficulty. In X-HDG the continuity between elements is imposed in a weak form, giving total freedom for a decoupled definition of the approximation space in every element. This is not the case in the

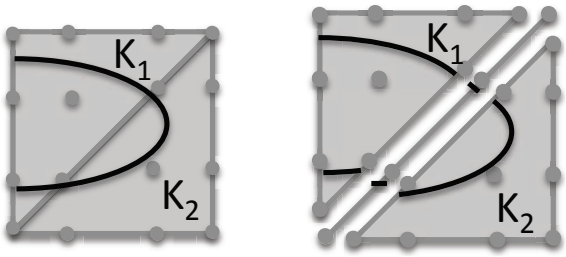


Figure 3.4: Representation of 2 triangles cut by one interface for X-FEM (continuous) and X-HDG. Element K_1 is split by the interface in 3 regions. The X-HDG discretization considers 2 Heaviside functions for enrichment in element K_1 and the cut side, and 1 Heaviside in element K_2.

context of continuous finite elements. In X-FEM, the usual practice is considering a unique definition of the level-set functions in the whole domain, to ensure continuity of the approximation in a simple way. For illustration purposes, Figure 3.4 shows a discretization example with 2 triangles cut by an interface, for a continuous approximation (left) and an HDG approximation (right). In this situation, the X-HDG discretization considers two Heaviside functions for enrichment in element K_1 and the cut side, and one Heaviside function for enrichment in element K_2. Continuity is imposed weakly by the X-HDG formulation in a natural and straight-forward way. However, a standard X-FEM approximation would consider a unique Heaviside function associated to the interface in both elements, in order to keep the continuity of the approximation on the side shared by the triangles. This is because in X-FEM all nodes have to be enriched with the same enrichment function. This would lead to an artificial link between the solution in the the two separated regions in K_1, with the consequent decrease of accuracy. An alternative in the context of continuous finite element is the so-called Virtual Node Algorithm (VNA) Assêncio and Teran [2013], which is based on a suitable duplication of nodes and elements, instead of the classical enrichment. However, the implementation of the VNA for complicated interfaces, as the one depicted in Figure 3.4, may be cumbersome.

3.2 Numerical examples

The performance of X-HDG is tested by several numerical examples in this section. Both straight and curved interfaces with discontinuous or continuous solutions across the interface are considered. For verification of the results obtained with X-HDG, results are compared against a standard HDG setting where the computational mesh

fits the interface. A last example with cut elements that are split by the interface in more than two regions is also considered, demonstrating the capability of X-HDG to handle this kind of problems.

3.2.1 Straight interface with zero jump conditions

In this first test case, the Poisson equation (3.1) is solved over a square domain $\Omega = (-1, 1)^2$ with a straight interface $\mathcal{I}$ at $x = 0.2031$. The material parameter ν is defined as $\nu_1 = 1$ in Ω_1 and $\nu_2 = 2.5$ in Ω_2. In Figure 3.5, the linear interface $\mathcal{I}$ separating the two material domains can be seen for both the X-HDG and the HDG settings. For X-HDG the interface is represented by the level set function $\phi = x - 0.2031$, and it is not fitted by the computational mesh. Elements cut by the interface are depicted in light gray. On other hand, for HDG the interface needs to coincide with element boundaries of the computational mesh. Note that generating a computational mesh fitting the interface is simple in this example, but may be cumbersome and costly for irregularly shaped curved interfaces, specially for evolving interfaces.

The source term and Dirichlet boundary conditions are set such that the analytical solution is

$$u(x) = \begin{cases} 5x^5 & \text{in } \Omega_1, \\ 2x^5 + A & \text{in } \Omega_2, \end{cases}$$

with $A = 3(0.2031)^5$. Zero jump conditions across the interface are imposed.

Figure 3.6 shows the convergence of X-HDG and HDG for the bimaterial problem with straight interface. Starting with an initial mesh using four elements per coordinate direction, four mesh refinement steps are considered—each refinement doubling the number of elements per direction—both for X-HDG and HDG. The meshes for HDG do not have uniform mesh size in order to fit the interface. The characteristic mesh size in the convergence plots is the mesh size of the uniform mesh with the same number of elements. The approximation degree is varied from $k = 1$ to $k = 4$. The results confirm that the X-HDG strategy retains the convergence rates and accuracy of standard HDG without the need of mesh adaptation to the interface. Optimal convergence of order $k + 1$ for u, and $k + 2$ for the postprocessed superconvergent solution u^*, is observed, with a slight effect of rounding errors in the solution with the finer mesh and degree $k = 4$.

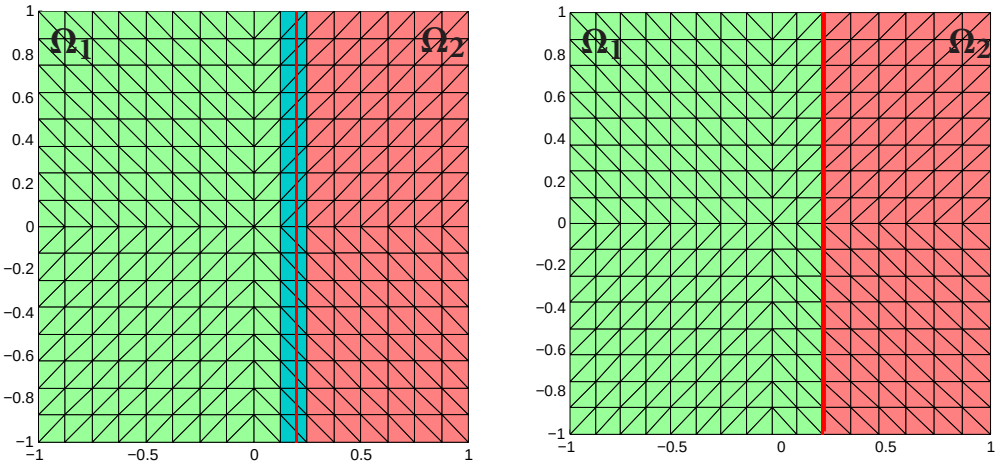


Figure 3.5: Linear interface with zero jump conditions: computational mesh for X-HDG and for standard HDG after two mesh refinements. On the left, the X-HDG mesh not adapted to the linear interface boundary $\mathcal{I}$ which shown in black. Elements in white are standard elements in domain one whereas elements in dark gray are standard elements in domain two. Elements cut by the interface $\mathcal{I}$ are shown in light gray shade. On the right, HDG mesh fitting to the linear interface is shown. Elements in domain one are shown in white whereas elements in domain two are shown in dark gray.

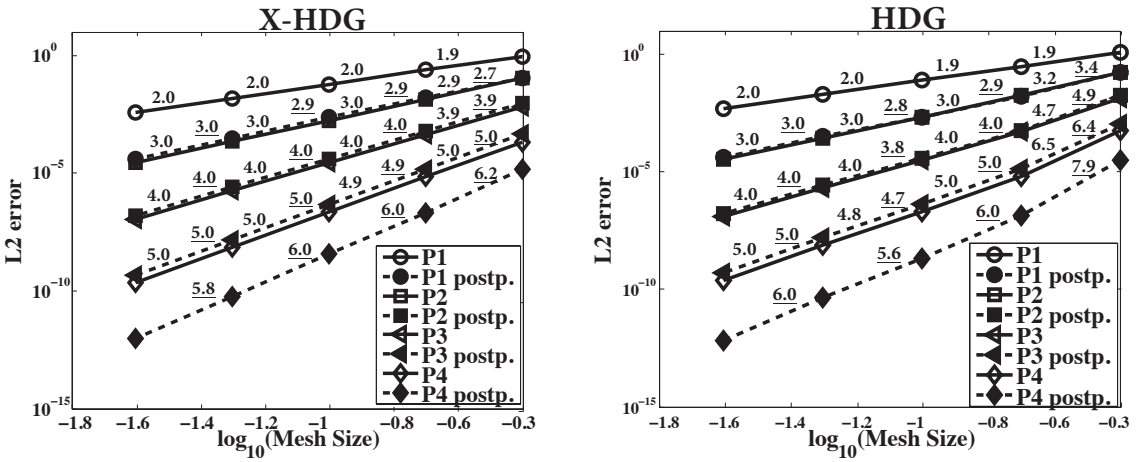


Figure 3.6: Linear interface with zero jump conditions: convergence history of X-HDG and HDG with solid line for the solution u (Pk) and dashed line for the post-processed superconvergent solution u^* (Pk postp). Slopes for each segment are shown.

3.2.2 Straight interface with non-zero jump conditions

As a second example, X-HDG is tested with a discontinuous solution over the interface, see *Remark 16*. The domain is again $\Omega = (0,1)^2$, the level-set function describing the interface is $\phi = x - 0.4$, the viscosity parameter is set to $\nu = 1$ in the entire do-

main, and homogeneous Dirichlet boundary conditions are imposed on the boundary. The jump conditions

$$u_2 - u_1 = 1, \quad [\![\boldsymbol{n} \cdot \boldsymbol{\nabla} u]\!] = 0 \quad \text{on } \mathcal{I},$$

are imposed on the interface, and the source term is chosen such that the analytical solution reads

$$u(x) = \begin{cases} \sin(\pi x)\sin(\pi y) & \text{in } \Omega_1, \\ \sin(\pi x)\sin(\pi y) + 1 & \text{in } \Omega_2. \end{cases}$$

Figure 3.7 shows the coarsest mesh used in our X-HDG calculations as well as the analytical solution. The mesh is refined three times—doubling the number of elements per direction in each refinement—with varying polynomial degrees between $k = 2$ and $k = 4$ for the convergence studies presented in Figure 3.8. Again, the results are compared against a conventional HDG setting where the interface is fitted by the computational mesh.

Figure 3.8 clearly shows that X-HDG performs equally well as the standard HDG method with a discontinuous solution over the interface without the need for a matching computational mesh.

3.2.3 Circular interface with zero jump conditions: heat distribution over a steady state bimaterial plate

In this example a circular interface $\mathcal{I}$ with radius $R = 0.5$ is considered to divide the square domain $\Omega = (-1, 1)^2$ into two regions, Ω_1 and Ω_2, as depicted in Figure 3.9. The heat distribution over a plate, which is made of two materials with different thermal conductivities, is computed. The analytical solution and the thermal conductivities are defined by

$$u(x) = \begin{cases} \frac{1}{\nu_1}(x^2 + y^2)^{5/2} & \text{in } \Omega_1, \\ \frac{1}{\nu_2}(x^2 + y^2)^{5/2} + (\frac{1}{\nu_1} - \frac{1}{\nu_2})R^5 & \text{in } \Omega_2, \end{cases} \quad \text{and} \quad \nu = \begin{cases} 1 & \text{in } \Omega_1, \\ 100 & \text{in } \Omega_2, \end{cases}$$

the corresponding source term is $f = -25(x^2 + y^2)^{3/2}$, Dirichlet boundary conditions are set on the boundary, and zero jump conditions are set on the interface.

Figure 3.9 shows the mesh used in X-HDG calculations after the first refinement step as well as the analytical solution. The mesh is isotropically refined three times and the degree of approximation is varied between $k = 1$ and $k = 3$. Convergence plots for the X-HDG solution are shown in Figure 3.10. Again optimal convergence

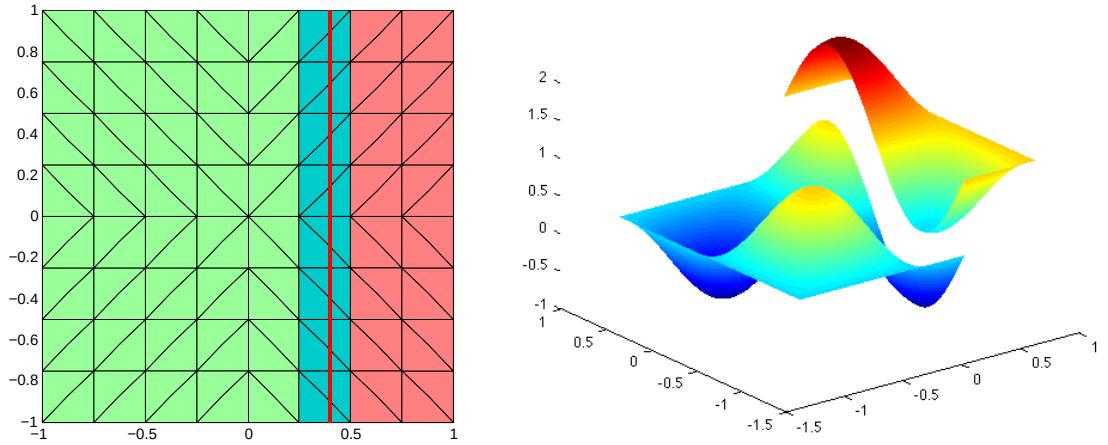


Figure 3.7: Straight Interface with non-zero jump conditions: X-HDG mesh after one mesh refinement and the interface $\mathcal{I}$ (left) and analytical solution (right).

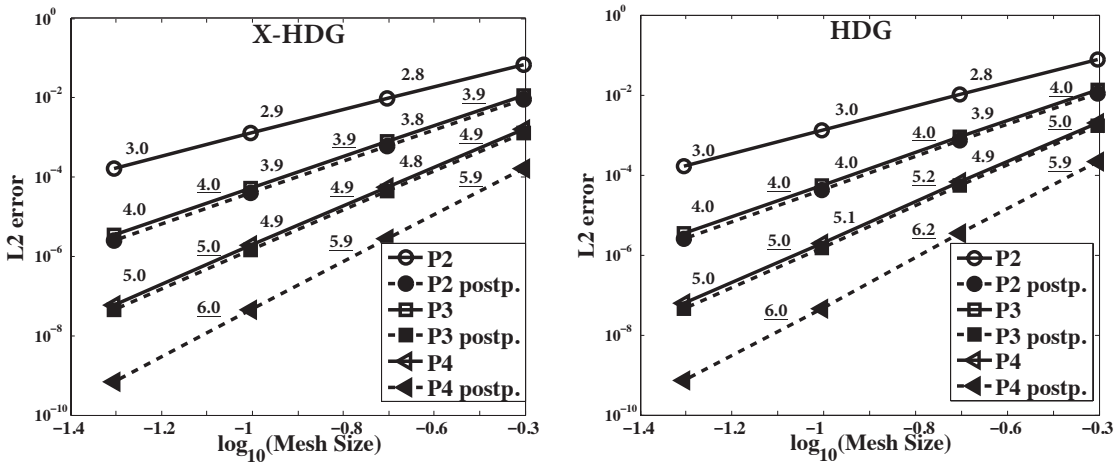


Figure 3.8: Straight Interface with non-zero jump conditions: convergence history of X-HDG and HDG with solid line for the solution u (Pk) and dashed line for the postprocessed superconvergent solution u^* (Pk postp). Slopes for each segment are shown.

rates are reached, for both the solution u and the superconvergence postprocess u^*, except for the last mesh with degree $k = 3$.

This convergence loss is the result of bad-cut situations (see Figures 3.11 and 3.12, left panel) causing an ill-conditioned matrix for the system of the global problem. By a bad-cut situation we mean a case where the area ratio of two material domains at the cut element is less than 0.1. This situation is familiar from X-FEM applications and there are plenty of strategies, complex or rather simple, to get rid of this ill-conditioning problem, see for instance Hansbo and Hansbo [2002]. Since the derivation of such stabilization strategies is not within the scope of this work,

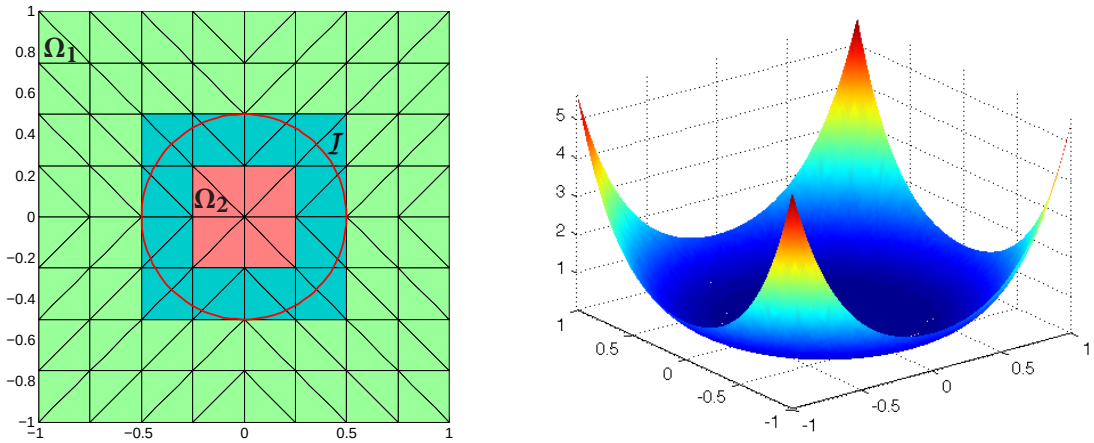


Figure 3.9: Circular interface with zero jump conditions: mesh used in X-HDG calculations after one mesh refinement and the circular interface $\mathcal{I}$ (left), analytical solution (right).

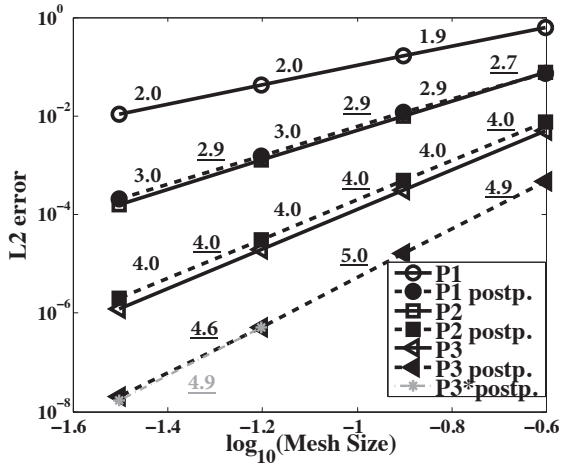


Figure 3.10: Circular interface with zero jump conditions: convergence history for the solution u (Pk, solid line), the superconvergent solution u^* (Pk postp., dashed line), and for u^* with a slightly modified mesh to avoid ill-conditioning (P3* postp.)

we choose to simply slightly modify the mesh, moving the nodes wherever we detect a bad-cut situation, as shown in Figures 3.11 and 3.12. This eliminates the ill-conditioning problem. The condition number of the matrix for the completely uniform mesh is $k_2 \simeq 10^{14}$, leading to poor accuracy in the results. After the slight modification of the mesh, avoiding the bad-cut situations, the condition number is $k_2 \simeq 5\ 10^9$. Note that this node-moving strategy is not proposed as a general solution but rather to demonstrate that X-HDG can converge optimally. The slight modification of the mesh recovers optimal convergence, with rates of order $k+1$ and $k+2$ for the solution and the post processed solution, respectively.

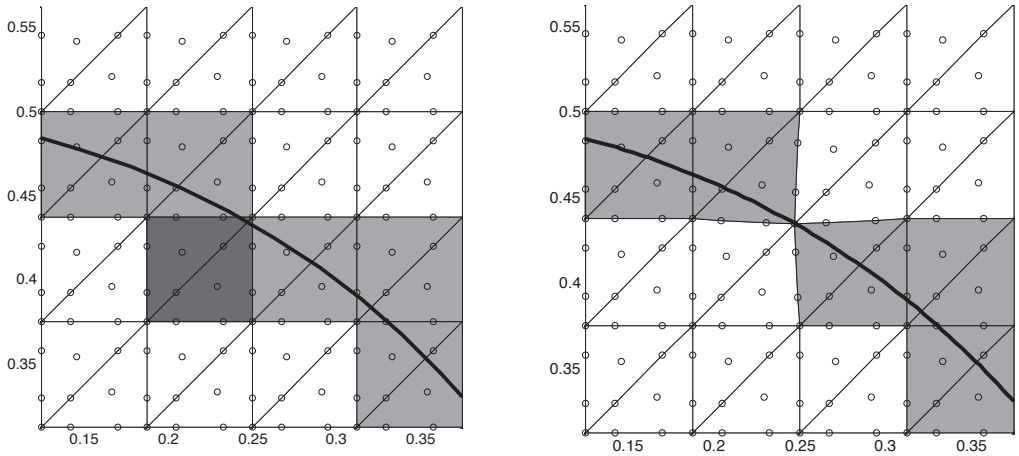


Figure 3.11: Example of bad-cut situation for the circular interface: zoom of the original uniform mesh (left) and of the modified mesh (right). Standard elements are shown in white, cut elements in gray and bad-cut elements in dark gray. In this case, the vertex that is too close to the interface is moved to be placed on top of it. The nodes of the affected elements are relocated keeping straigth sides.

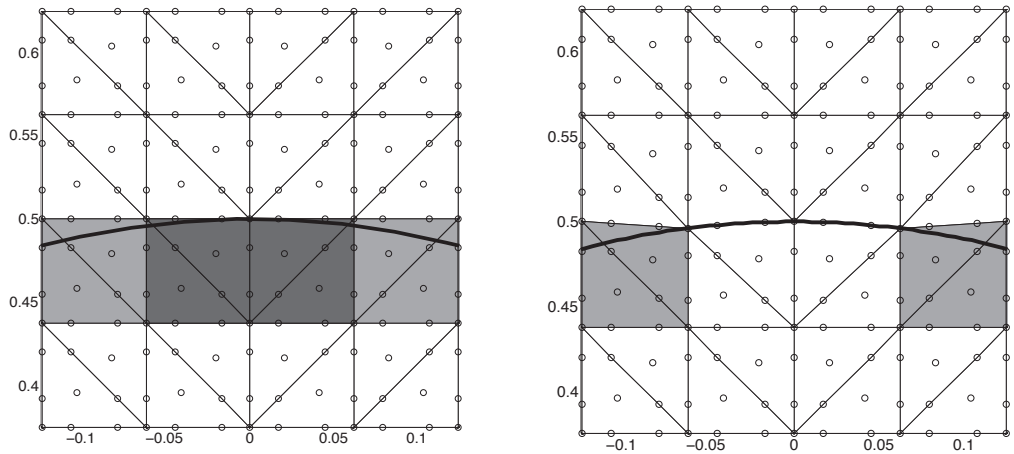


Figure 3.12: Example of bad-cut situation for the circular interface: zoom of the original uniform mesh (left) and of the modified mesh (right). Standard elements are shown in white, cut elements in gray and bad-cut elements in dark gray. Again, the vertexes that are too close to the interface are moved to be placed on top of it. If a face has two moved vertexes, the whole face is moved to a curved face fitting the interface.

This problem is solved in Huynh et al. [2013] using standard HDG with a mesh fitting the circular interface, using superparametric elements for the curved elements along the interface. In this work, optimal convergence and similar levels of accuracy are obtained with X-HDG and standard isoparametric approximations, even in the presence of the curved elements in Figure 3.12. It is also worth mentioning that

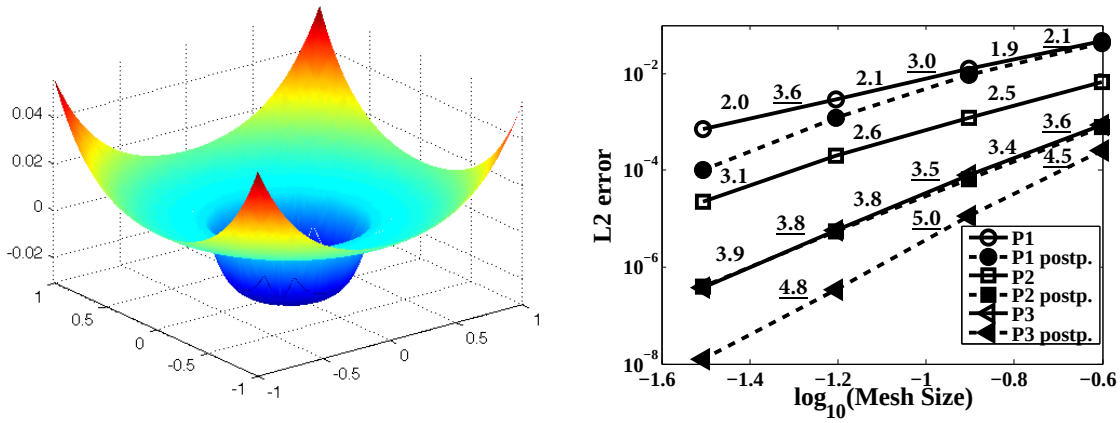


Figure 3.13: Circular interface with zero jump conditions with $\nu_1 = 100$ and $\nu_2 = 1$: analytical solution (left), convergence history for the solution u (Pk, solid line) and the superconvergent solution u^* (Pk postp., dashed line) (right).

the analytical solution for this problem has a singularity in the fifth derivative and, therefore, theoretical convergence rates are available up to degree $k = 3$. Numerical experiments show that the convergence rate for degree $k = 4$ is lower than 5.

Figure 3.13 shows the convergence history and the analytical solution in case of switching the materials in Ω_1 and Ω_2, i.e., setting $\nu_1 = 100$ and $\nu_2 = 1$ and keeping the problem statement the same. The mesh fixing strategy is also used in this computations in the presence of bad-cut situations, again leading to optimal convergence rates.

3.2.4 Kidney-shaped interface

To study the behavior of X-HDG in the context of more involved interfaces, a problem with a kidney-shaped interface according to Huynh et al. [2013] is considered next. The domain is $\Omega = (-1, 1)^2$, the level set function is defined as

$$\varphi(x, y) = \left(3((x + 0.5)^2 + y^2) - x - 0.5\right)^2 - \left((x + 0.5)^2 + y^2\right) + 0.1,$$

and the material parameters are $\nu_1 = 10$ in $\Omega_1 = \{(x, y) \in \Omega | \varphi(x, y) > 0\}$ (i.e., the outer subdomain) and $\nu_2 = 1$ in $\Omega_2 = \Omega \backslash \overline{\Omega_1}$. The source term, the Dirichlet boundary conditions, and the jump conditions on the interface are set such that the analytical solution is

$$u(x, y) = \begin{cases} \frac{1}{\nu_1} \cos(1 - x^2 - y^2) & \text{for } (x, y) \in \Omega_1 \\ \sin(2x^2 + y^2 + 2) + x & \text{for } (x, y) \in \Omega_2. \end{cases}$$

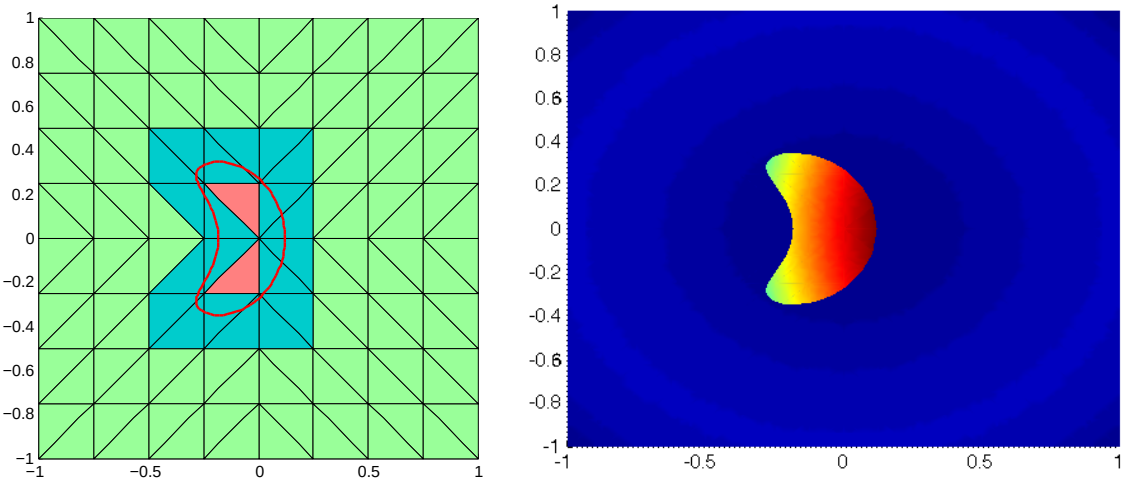


Figure 3.14: Kidney-shaped interface: mesh after two refinements together with the kidney shaped interface (left) and solution (right).

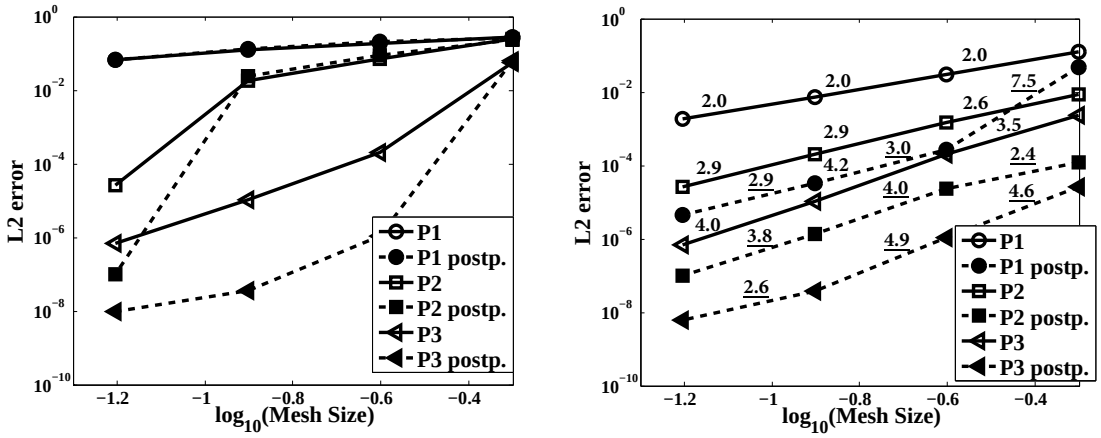


Figure 3.15: Kidney-shaped interface: convergence plots with a level set function of degree $k+1$ (left) and of degree $k+2$ (right).

Figure 3.14 shows the convergence plots for varying approximation degree k and mesh size h. The left panel shows the X-HDG solution with a k-th degree level set function and with degree $k+1$ for the superconvergent solution, whereas the right panel shows the X-HDG solution with a level set function of degree $k+2$. In this example, the geometrical description of the interface crucially influences the accuracy of the solution, and a representation with degree $k+2$ is required to reach optimal convergence. For coarse meshes or low degrees, the k-th degree interface representation induces substantial errors in the solution. Nevertheless, it is worth noting that for finer meshes these errors become less relevant and the accuracy of an interface representation of degrees k and $k+2$, respectively, is similar. For the last

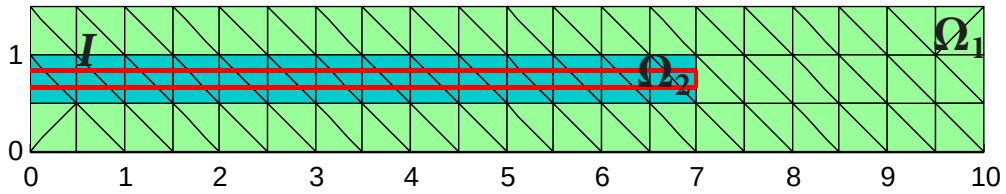


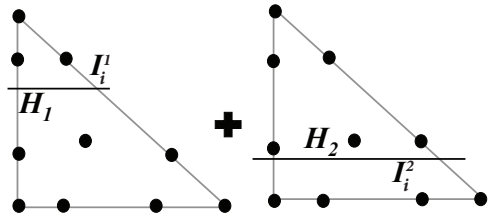
Figure 3.16: Rectangular Interface: domain and computational mesh. Elements cut by the interface are shown in light gray.

mesh and highest degree of approximation the effect of ill-conditioning is observed, limiting the accuracy to errors around 10^{-8} see Figure 3.15.

The need of an accurate representation of the interface to reach optimal convergence rates is in accordance with the analysis in Huynh et al. [2013], where the authors claim that for standard HDG with a mesh fitted to the interface, superparametric elements with a geometry description of degree $2k+1$ are necessary to ensure optimal convergence rates for the solution and for superconvergent postprocessing. The natural extension of this result to X-HDG is the need of a level set function of degree $2k+1$ to ensure optimal convergence in the general case. Nonetheless, in this example a geometrical description of the interface of degree $k+2$ is enough to obtain maximum accuracy for each mesh and degree.

3.2.5 Rectangular interface with double enrichment

The heat equation is solved over a rectangular plate $\Omega = (0, 10) \times (0, 1.5)$, with an approximation of degree 3. The plate is composed of two different materials separated by a rectangular interface. The subdomain inside the rectangular interface is $\Omega_2 = (0, 7) \times (0.65, 0.85)$, with permittivity constant $\nu_2 = 1$. The remainder of the domain is $\Omega_1 = \Omega \setminus \Omega_2$ with permittivity constant $\nu_1 = 10$, see Figure 3.16. Homogenous Neumann boundary conditions are imposed at $\{x = 0\}$ and $\{x = 10\}$, whereas at $\{y = 0\}$ and at $\{y = 1.5\}$ Dirichlet boundary conditions are set to 0 and 1, respectively. Considering the computational mesh shown in Figure 3.16, the elements cut by the interface (in light gray) are split in three regions, two with material constant $\nu = \nu_1$, and the interior one with $\nu = \nu_2$. In this situation, cut elements and cut faces should be enriched with two Heaviside functions, as explained in section 3.1.4, to have a completely independent approximation for the solution in



$$H_1 = \begin{cases} -1, & y \in (1, 0.85) \\ 1, & y \in (0.85, 0.5) \end{cases} \quad H_2 = \begin{cases} -1, & y \in (1, 0.65) \\ 1, & y \in (0.65, 0.5) \end{cases}$$

Figure 3.17: Rectangular Interface: representation of the two Heaviside functions used for the *double enrichment* at cut elements.

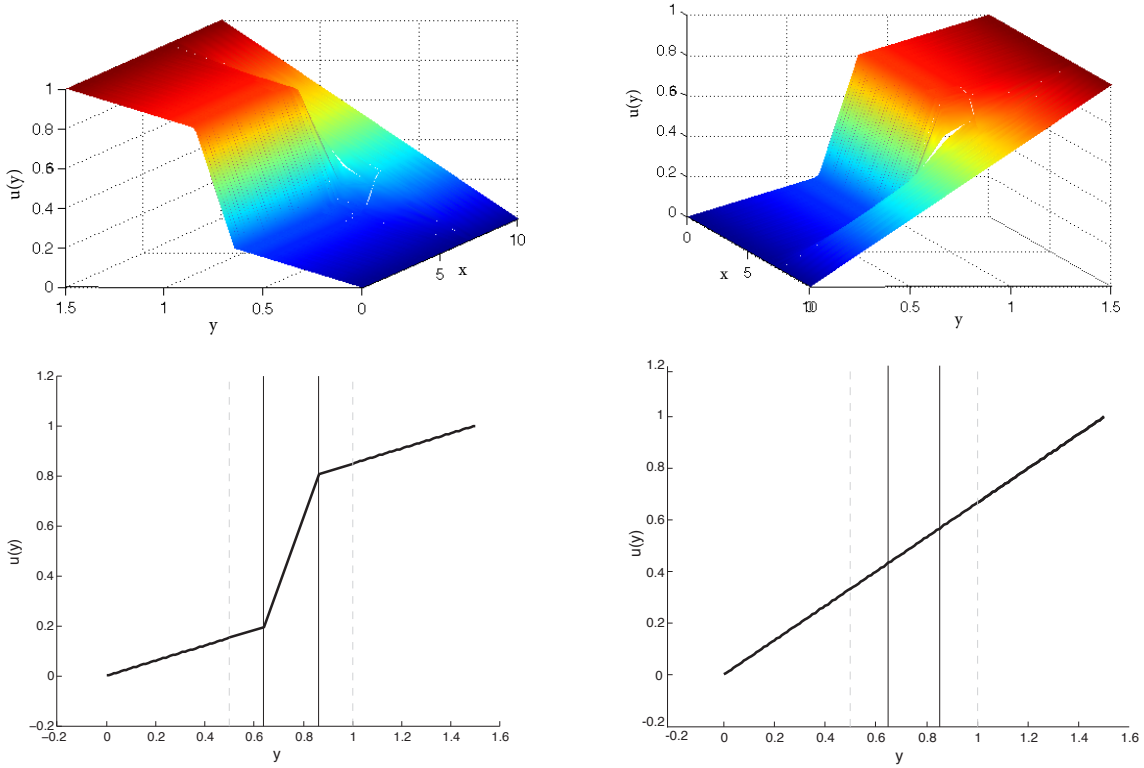


Figure 3.18: Rectangular Interface: X-HDG solution, with degree $k = 3$ and *double enrichment* at cut elements and cut faces, from two different points of view (top), and sections on $\{x = 0\}$ (bottom left) and on $\{x = 10\}$ (bottom right). Vertical discontinuous and continuous lines represent element boundaries and interfaces, respectively.

the three regions. That is, in the cut elements the solution is approximated as

$$u \simeq a + H_1 b + H_2 c$$

where a, b and c are polynomials of degree 3, and H_1 and H_2 are the two Heaviside functions represented in Figure 3.17. Faces cut by the rectangular interface are also

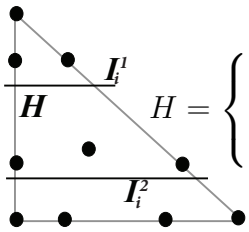
enriched with these two Heaviside functions, and the face at $y = 7$, shared by a cut element and a standard element, is not enriched.

Figure 3.18 shows the X-HDG solution from different viewing angles and over two cross sections. By using double enrichment in the cut elements, the approximations in the three regions are independent, and the X-HDG solution can freely adjust to the kinks in the analytical solution. The solution is verified to match with an HDG solution in a fine mesh adapted to the interface. As expected, on the right boundary (Neumann boundary $\{x = 10\}$ with constant permittivity parameter $\nu = \nu_1$, far enough to the material interface) the solution is close to a linear variation from temperature 0 to temperature 1. On the left boundary (Neumann boundary $\{x = 0\}$ with two materials) the solution is close to a linear variation in the three regions, with larger slope for the interior region, which has smaller permittivity $\nu = \nu_2 < \nu_1$. In the interior of the domain the solution varies in a continuous way between this two sections, with sharper variations close to $x = 7$. To verify the convenience of considering two Heaviside enrichments in the cut elements, the problem is now solved with a single Heaviside function. That is, the approximation at cut elements is

$$u \simeq a + Hb$$

with the Heaviside function H represented in Figure 3.19. This would be the standard approach for an X-FEM formulation based on continuous finite elements: a single enrichment function is defined in the whole domain, and for this example, this means a single Heaviside enrichment in the cut elements.

Figure 3.20 shows the X-HDG solution with single enrichment, again from different viewing angles and over two cross sections. The section on $\{x = 0\}$ shows how the single enrichment is not suitable to capture the solution, due to the non-physical dependency of the approximation on the top ($y > 0.85$) and bottom ($y < 0.65$) regions. This causes over- and undershoots close to the interface, and large discontinuties at the element boundaries, demonstrating that the approximation with a single enrichment is not rich enough to properly adjust to the kinks in the solution. To corroborate this conclusion, the problem is also solved with an X-FEM formulation based on continuous finite elements, with a unique Heaviside function defined in the whole domain to represent the interface, that is, with the single enrichment at cut elements. Nitsche's method (or, equivalently, the interior penalty method) is considered for weakly imposing the continuity of the solution and of the normal flux across the interface, see Legrain et al. [2012] for details. Similarly to the X-



$$H = \begin{cases} -1, & y \in (1, 0.85) \\ 1, & y \in (0.85, 0.65) \\ -1, & y \in (0.65, 0.5) \end{cases}$$

Figure 3.19: Rectangular Interface: Heaviside function for the *single enrichment* solution.

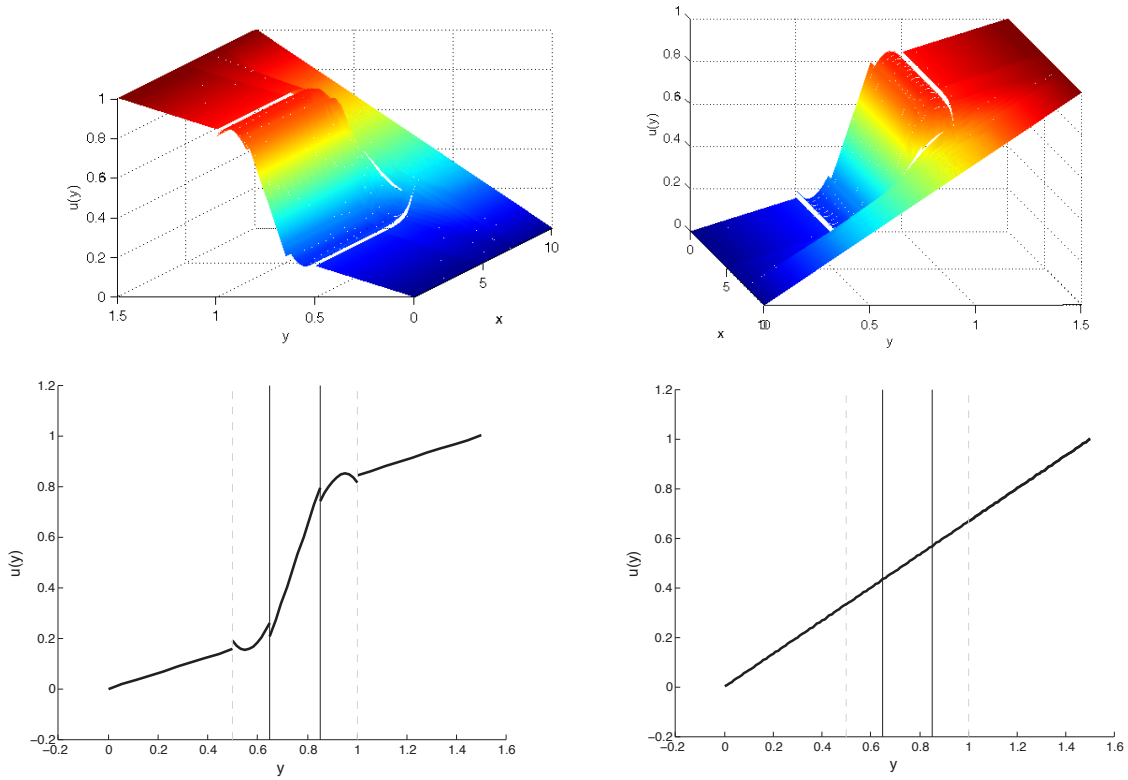


Figure 3.20: Rectangular Interface: X-HDG solution , with degree $k = 3$ and *single enrichment* at cut elements and cut faces, from two different points of view (top), and sections on $\{x = 0\}$ (bottom left) and on $\{x = 10\}$ (bottom right). Vertical discontinuous and continuous lines represent element boundaries and interfaces, respectively. The approximation is not rich enough to capture the solution.

HDG solution with single enrichment, the X-FEM solution is not able to properly represent the solution. Figure 3.21 shows the section on $\{x = 0\}$ for two different values of the Nitsche parameter β. For $\beta = 100$ the X-FEM solution presents large discontinuities across the interface and the slopes in the exterior region Ω_1 are not properly captured. Increasing the parameter to $\beta = 1000$ improves the solution, with small discontinuities and better approximation of the solution in the elements in the

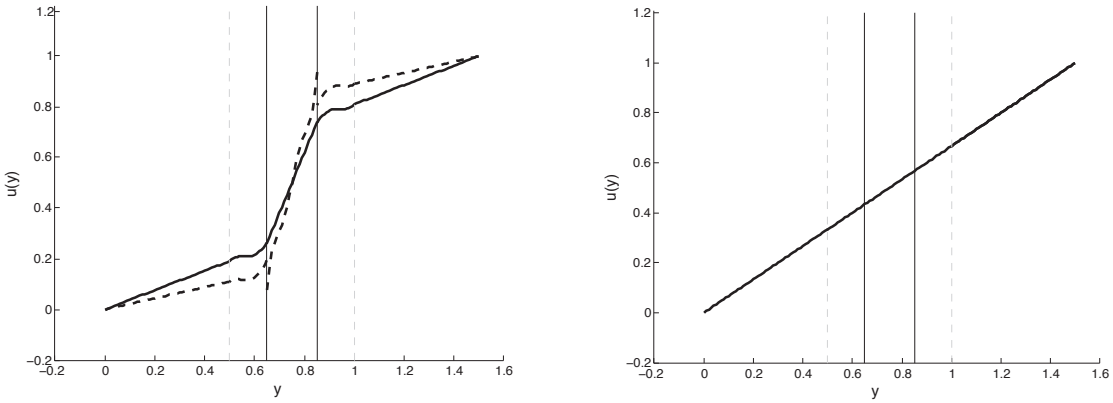


Figure 3.21: Rectangular Interface: sections of the X-FEM solution, with degree $k = 3$, on $\{x = 0\}$ (left) and on $\{x = 10\}$ (right), for $\beta = 100$ (dashed line) and $\beta = 1000$ (solid line). Vertical discontinuous and continuous lines represent element boundaries and interfaces, respectively.

exterior domain. However the approximation in the cut elements is still far from the analytical solution. Again, the approximation with single enrichment is not able to represent the kinks of the solution on the interface.

Thus, it can be concluded that two Heaviside enrichments in cut elements are necessary to properly capture the solution in this example. The approximation with double enrichment can be easily handled in the X-HDG formulation, where continuity of the approximation between elements is imposed in weak form, whereas it is not straight-forward for a classical X-FEM formulation.

3.3 Conclusions and final remarks

The eXtended Hybridizable Discontinuous Galerkin (X-HDG) method has been presented for the solution of bimaterial problems. The solution is enriched by Heaviside functions in the elements and faces cut by the interface, in order to properly represent the discontinuities in the derivatives of the solution, or even in the solution, across the interface.

The formulation for the global problem and for the local problem on elements not cut by the interface is the standard HDG formulation. For the elements cut by the interface, a new HDG local problem is derived taking into account the discontinuous approximation inside the element and the interface conditions to be imposed across the interface. Following the HDG ideas, a new trace variable on the interface is con-

sidered, which is locally eliminated afterwards using the interface conditions, leading to a local problem with the same structure as standard HDG.

Numerical experiments demonstrate that X-HDG keeps the HDG optimal convergence and superconvergence rates (of order $k+1$ in the $\mathcal{L}_2$ norm for the primal unknown and for its derivatives, and of order $k+2$ for the post-processed solution), with similar levels of accuracy.

The last numerical example also shows the capability of X-HDG to handle elements that are split by the interface in more than two regions. In this situation, more than one enrichment function should be considered to ensure an independent approximation space in every region. With an X-HDG formulation different enrichment functions can be considered in each element, because continuity between elements is imposed in a weak form. This is not the case for X-FEM methods based on continuous approximations, for which using different enrichment spaces in each element is not possible or leads to a cumbersome implementation.

Next, the implementation of X-HDG for rather complex Stokes flow for void and bimaterial interfaces is presented to amplify the validity of the method over different PDE's.

Chapter 4

X-HDG for Stokes Problem - Void Interfaces

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with an interior boundary $\mathcal{I}$ (also referred as interface) and an exterior boundary $\partial\Omega^{ext} := \partial\Omega \backslash \mathcal{I}$. The following problem is considered,

$$\begin{aligned} -\boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} \boldsymbol{u}) + \boldsymbol{\nabla} p &= \boldsymbol{f} \quad \text{in } \Omega \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} &= \mathbf{0} \quad \text{in } \Omega \\ (-\nu \boldsymbol{\nabla} \boldsymbol{u} + p\mathbf{I}) \cdot \boldsymbol{n} &= \boldsymbol{g} \quad \text{on } \mathcal{I} \\ \boldsymbol{u} &= \boldsymbol{u}_D \quad \text{on } \partial\Omega^{ext} \end{aligned} \tag{4.1}$$

where $\boldsymbol{u}$ is the velocity, p is the pressure, ν is the material viscosity, $\mathbf{I}$ is the second-order identity tensor, $\boldsymbol{f}$ is an external force, $\boldsymbol{u}_D$ are prescribed values on the exterior boundary, and $\boldsymbol{g}$ is a prescribed traction on the interior boundary, i.e. the void boundary. Neumann boundary conditions are considered on the interior boundary $\mathcal{I}$. The implementation of Dirichlet boundary conditions on $\mathcal{I}$ is developed in Section 4.3. On the exterior boundary $\partial\Omega^{ext}$, Dirichlet boundary conditions are considered; however, other boundary conditions at $\partial\Omega^{ext}$ do not add any difficulty, since they are implemented as in standard HDG. The domain Ω is now assumed to be covered by a finite element mesh with elements $\{K_i\}_{i=1}^{\mathtt{n_{el}}}$ and faces $\{\Gamma_i\}_{i=1}^{\mathtt{n_{fc}}}$ satifying the conditions in (2.2) and (2.3). That is, the mesh covers the domain, and fits the exterior boundary, but the interface $\mathcal{I}$ may cut through some elements.

Now the problem (4.1) can be split in a set of local problems and some global

equations as usual,

$$\left.\begin{aligned} \boldsymbol{L} - \boldsymbol{\nabla}\boldsymbol{u} &= \boldsymbol{0}, && \text{in } K_i \\ \boldsymbol{\nabla}\cdot(-\nu\boldsymbol{L} + p\mathbf{I}) &= \boldsymbol{f}, && \text{in } K_i \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0, && \text{in } K_i \\ \boldsymbol{u} &= \widehat{\boldsymbol{u}} && \text{on } \partial K_i \\ \frac{1}{|K_i|}\int_{K_i} p\, dV &= \rho_i \end{aligned}\right\} \text{ if } K_i \subset \Omega \tag{4.2a}$$

$$\left.\begin{aligned} \boldsymbol{L} - \boldsymbol{\nabla}\boldsymbol{u} &= \boldsymbol{0}, && \text{in } \Omega_i \\ \boldsymbol{\nabla}\cdot(-\nu\boldsymbol{L} + p\mathbf{I}) &= \boldsymbol{f}, && \text{in } \Omega_i \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0, && \text{in } \Omega_i \\ (-\nu\boldsymbol{\nabla}\boldsymbol{u} + p\mathbf{I})\cdot\boldsymbol{n} &= \boldsymbol{g} && \text{on } \mathcal{I}_i \\ \boldsymbol{u} &= \widehat{\boldsymbol{u}} && \text{on } \partial\Omega_i\backslash\mathcal{I}_i \end{aligned}\right\} \text{ if } \mathcal{I}\cap K_i \neq \emptyset \tag{4.2b}$$

$$[\![(-\nu\boldsymbol{L} + p\mathbf{I})\cdot\boldsymbol{n}]\!] = 0 \quad \text{on } \Gamma\backslash\partial\Omega^{ext}, \tag{4.3a}$$

$$\int_{\partial K_i} \widehat{\boldsymbol{u}}\cdot\boldsymbol{n}\, dS = 0 \quad \text{if } K_i \subset \Omega, \text{ for } i = 1,\dots,\mathtt{n_{el}}, \tag{4.3b}$$

$$\widehat{\boldsymbol{u}} = \mathbb{P}_2(\boldsymbol{u}_D) \quad \text{on } \partial\Omega^{ext} \tag{4.3c}$$

where, for cut elements, $\Omega_i := \Omega\cap K_i$ and $\mathcal{I}_i := \mathcal{I}\cap K_i$.

As in standard HDG, the variable $\boldsymbol{L}$ corresponds to the gradient of $\boldsymbol{u}$, allowing the splitting of the PDE in two first order PDEs, $\widehat{\boldsymbol{u}}$ corresponds to the trace of $\boldsymbol{u}$ at the mesh faces Γ, and ρ_i corresponds to the mean of pressure at the element, which is only a scalar for each element. Equations (4.2a) correspond to the Stokes problem for standard elements whereas equations (4.2b) correspond to the Stokes problem in the elements that intersect the interface, with Neumann boundary conditions on the interface $\mathcal{I}_i$ and Dirichlet boundary conditions on the element faces, $\partial K_i\backslash\mathcal{I}_i$. Note that now, the problem is not of pure Dirichlet type any more therefore, there is no need to set the pressure constraint. The local problems (4.2a) and (4.2b) can be solved element-by-element given $\widehat{\boldsymbol{u}}$ and ρ_i. Thus, now the problem reduces to determine $\widehat{\boldsymbol{u}}$ and $\{\rho_i\}_{i=1}^{\mathtt{n_{el}}}$ with the global equations (4.3). Equation (4.3a) is the conservativity condition, imposing continuity of the normal component of the gradient of velocity across faces. Finally, equation (4.3b) imposes the incompressibility condition on the boundary of the elements, ensuring well posedness. Discretization spaces, $\mathcal{V}^h$ and Λ^h, defined in (1.8), are considered now for elemental variables $\boldsymbol{u}$, $\boldsymbol{L}$ and p, and for the trace variable $\widehat{\boldsymbol{u}}$, respectively. Then, the discretization of the local problems (4.2)

and the global equations (4.3) leads to the complete X-HDG formulation. Next the details of the discretization of the local problems for the cut elements are explained, for the details of discretization of the local problems for the standard elements please refer to Section 1.3.2.

4.1 Local problem for Neumann interfaces

X-HDG local problem at cut element K_i leads to the discretization of (4.2b), that is: given $\widehat{\boldsymbol{u}} \in \Lambda^h$, find $\boldsymbol{u} \in [\mathcal{P}_k(\Omega_i)]^d$, $\boldsymbol{L} \in [\mathcal{P}_k(\Omega_i)]^{d\times d}$ and $p \in \mathcal{P}_k(\Omega_i)$ such that,

$$\begin{aligned}
&\int_{\Omega_i} \boldsymbol{\nabla}\cdot(-\nu\boldsymbol{L})\cdot\boldsymbol{v}\,dV + \int_{\Omega_i}\boldsymbol{\nabla}p\cdot\boldsymbol{v}\,dV + \int_{\partial\Omega_i\backslash\mathcal{I}_i}\nu\tau(\boldsymbol{u}-\widehat{\boldsymbol{u}})\cdot\boldsymbol{v}\,dS \\
&\qquad\qquad + \int_{\mathcal{I}_i}\nu\tau(\boldsymbol{u}-\widetilde{\boldsymbol{u}})\cdot\boldsymbol{v}\,dS = \int_{\Omega_i}\boldsymbol{f}\cdot\boldsymbol{v}\,dV \\
&\int_{\Omega_i}\boldsymbol{L}:\boldsymbol{Q}\,dV + \int_{\Omega_i}(\boldsymbol{\nabla}\cdot\boldsymbol{Q})\cdot\boldsymbol{u}\,dV - \int_{\partial\Omega_i\backslash\mathcal{I}_i}(\boldsymbol{Q}\cdot\boldsymbol{n})\cdot\widehat{\boldsymbol{u}}\,dS - \int_{\mathcal{I}_i}(\boldsymbol{Q}\cdot\boldsymbol{n})\cdot\widetilde{\boldsymbol{u}}\,dS = 0 \\
&\int_{\Omega_i}\boldsymbol{u}\cdot\boldsymbol{\nabla}q\,dV - \int_{\partial\Omega_i\backslash\mathcal{I}_i}(\widehat{\boldsymbol{u}}\cdot\boldsymbol{n})q\,dS - \int_{\mathcal{I}_i}(\widetilde{\boldsymbol{u}}\cdot\boldsymbol{n})q\,dS = 0
\end{aligned} \tag{4.4}$$

for all $\boldsymbol{v} \in [\mathcal{P}_k(\Omega_i)]^d$, $\boldsymbol{Q} \in [\mathcal{P}_k(\Omega_i)]^{d\times d}$ and $q \in \mathcal{P}_k(\Omega_i)$. The first equation in (4.4) can be derived by applying integration by parts, replacing the velocity gradient by the numerical velocity gradient (1.21) and undoing the integration by parts. The second and third equations in (4.2b) are obtained by simply applying integration by parts and replacing the boundary conditions $\boldsymbol{u} = \widehat{\boldsymbol{u}}$ on $\partial\Omega_i\backslash\mathcal{I}_i$ and $\boldsymbol{u} = \widetilde{\boldsymbol{u}}$ on $\mathcal{I}_i$. Here, being different from (1.20), two new integral terms are considered corresponding to the integrals along the interface $\mathcal{I}_i$. The new integral terms include the new trace variable $\widetilde{\boldsymbol{u}}$ defined on the interface $\mathcal{I}_i$. At a first sight equations (4.4) might seem complex however after the hybridization process, all elemental unknowns $\boldsymbol{u}$, $\boldsymbol{L}$, p as well as the new interface variable $\widetilde{\boldsymbol{u}}$ are going to be written only in terms of trace variable $\widehat{\boldsymbol{u}}$. To do that, equations (4.4) are discretized to end up with the following elemental system,

$$\begin{cases}
\left[\mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i}\right]\mathbf{u}^i + \mathbf{A}_{uL}^{\Omega_i}\mathbf{L}^i + \mathbf{A}_{up}^{\Omega_i}\mathbf{p}^i + \mathbf{A}_{u\widehat{u}}^{\Omega_i}\mathbf{\Lambda}^i + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\widetilde{\mathbf{u}}^i = \mathbf{f}_u^{K_i} \\
\mathbf{A}_{Lu}^{\Omega_i}\mathbf{u}^i + \mathbf{A}_{LL}^{\Omega_i}\mathbf{L}^i + \mathbf{A}_{L\widehat{u}}^{\Omega_i}\mathbf{\Lambda}^i + \mathbf{A}_{L\widetilde{u}}^{\mathcal{I}_i}\widetilde{\mathbf{u}}^i = \mathbf{0} \\
\mathbf{A}_{pu}^{\Omega_i}\mathbf{u}^i + \mathbf{A}_{p\widehat{u}}^{\Omega_i}\mathbf{\Lambda}^i + \mathbf{A}_{p\widetilde{u}}^{\mathcal{I}_i}\widetilde{\mathbf{u}}^i = \mathbf{0}
\end{cases} \tag{4.5}$$

Now, the local problem is closed and the new interface variable $\widetilde{\boldsymbol{u}}$ is eliminated from the system (4.5) by imposing the Neumann boundary condition on interface which

is,

$$(-\nu \boldsymbol{L} + p\mathbf{I}) \cdot \boldsymbol{n} = \mathbf{g} \quad \text{on } \mathcal{I}_i \tag{4.6}$$

The weak form of the Neumann condition on $\mathcal{I}_i$, considering the numerical flux (1.21), is: given $\boldsymbol{u} \in [\mathcal{P}_k(\Omega_i)]^d$, $\boldsymbol{L} \in [\mathcal{P}_k(\Omega_i)]^{d\times d}$ and $p \in \mathcal{P}_k(\Omega_i)$ find $\widetilde{\boldsymbol{u}} \in \mathcal{P}_k(\mathcal{I}_i)$ such that

$$\int_{\mathcal{I}_i} (-\nu \boldsymbol{L} \cdot \boldsymbol{n}) \cdot \widetilde{\boldsymbol{v}}\, dS + \int_{\mathcal{I}_i} (p\boldsymbol{n}) \cdot \widetilde{\boldsymbol{v}}\, dS + \int_{\mathcal{I}_i} \tau\nu(\boldsymbol{u} - \widetilde{\boldsymbol{u}}) \cdot \widetilde{\boldsymbol{v}}\, dS = \int_{\mathcal{I}_i} \mathbf{g} \cdot \widetilde{\boldsymbol{v}}\, dS \tag{4.7}$$

for all $\widetilde{\boldsymbol{v}} \in \mathcal{P}_k(\mathcal{I}_i)$ and its discretization leads to a system of the form

$$\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i u}\mathbf{u}^i + \mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i L}\mathbf{L}^i + \mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i p}\mathbf{p}^i + \mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i \widetilde{u}^i}\widetilde{\mathbf{u}}^i = \mathbf{g} \tag{4.8}$$

Using equation (4.8) the interface variable $\widetilde{\boldsymbol{u}}^i$ can be expressed in terms of $\mathbf{u}^i, \mathbf{L}^i$ and $\mathbf{p}^i$

$$\widetilde{\boldsymbol{u}}^i = \mathbf{T}^i_u\mathbf{u}^i + \mathbf{T}^i_L\mathbf{L}^i + \mathbf{T}^i_p\mathbf{p}^i + \mathbf{T}^i_i \tag{4.9}$$

where

$$\begin{gathered} \mathbf{T}^i_u = \left[-\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i\widetilde{u}^i}\right]^{-1}\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i u}, \quad \mathbf{T}^i_{\boldsymbol{L}} = \left[-\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i\widetilde{u}^i}\right]^{-1}\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i L}, \\ \mathbf{T}^i_p = \left[-\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i\widetilde{u}^i}\right]^{-1}\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i p}, \quad \mathbf{T}^i_i = \left[-\mathbf{A}^{\mathcal{I}_i}_{\widetilde{u}^i\widetilde{u}^i}\right]^{-1}\mathbf{g} \end{gathered} \tag{4.10}$$

Plugging (4.9) into (4.5) the final system for local problem is obtained as

$$\begin{aligned} &\left[\mathbf{A}^{\Omega_i}_{uu} + \mathbf{A}^{\mathcal{I}_i}_{uu} + \mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}^i}\mathbf{T}^i_u\right]\mathbf{u}^i + \left[\mathbf{A}^{\Omega_i}_{uL} + \mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}^i}\mathbf{T}^i_L\right]\mathbf{L}^i \\ &\qquad + \left[\mathbf{A}^{\Omega_i}_{up} + \mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}^i}\mathbf{T}^i_p\right]\mathbf{p}^i + \mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}^i}\widehat{\mathbf{u}}^i = \mathbf{f} - \mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}^i}\mathbf{T}^i_i \\ &\left[\mathbf{A}^{\Omega_i}_{Lu} + \mathbf{A}^{\mathcal{I}_i}_{L\widetilde{u}^i}\mathbf{T}^i_u\right]\mathbf{u}^i + \left[\mathbf{A}^{\Omega_i}_{LL} + \mathbf{A}^{\mathcal{I}_i}_{L\widetilde{u}^i}\mathbf{T}^i_L\right]\mathbf{L}^i + \left[\mathbf{A}^{\mathcal{I}_i}_{L\widetilde{u}^i}\mathbf{T}^i_p\right]\mathbf{p}^i + \mathbf{A}^{\mathcal{I}_i}_{L\widetilde{u}}\widehat{\mathbf{u}}^i = -\mathbf{A}^{\mathcal{I}_i}_{L\widetilde{u}^i}\mathbf{T}_i \\ &\left[\mathbf{A}^{\Omega_i}_{pu} + \mathbf{A}^{\mathcal{I}_i}_{p\widetilde{u}^i}\mathbf{T}^i_u\right]\mathbf{u}^i + \left[\mathbf{A}^{\mathcal{I}_i}_{p\widetilde{u}^i}\mathbf{T}^i_L\right]\mathbf{L}^i + \left[\mathbf{A}^{\mathcal{I}_i}_{p\widetilde{u}^i}\mathbf{T}^i_p\right]\mathbf{p}^i + \mathbf{A}^{\mathcal{I}_i}_{p\widetilde{u}^i}\widehat{\mathbf{u}}^i = -\mathbf{A}^{\mathcal{I}_i}_{p\widetilde{u}}\mathbf{T}^i_i \end{aligned} \tag{4.11}$$

Now similar to (1.24) above final system (4.11) can be solved for elemental variables $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$ for each element to obtain so called *local solver*. It is important to note that the local solver for the cut element has very similar structure to the local solver for the standard element (1.25), having the form

$$\mathbf{u}^i = \mathbf{S}^{K_i}_u\mathbf{\Lambda}^i + \mathbf{f}^{K_i}_U, \quad \mathbf{L}^i = \mathbf{S}^{K_i}_L\mathbf{\Lambda}^i + \mathbf{f}^{K_i}_L, \quad \mathbf{p}^i = \mathbf{S}^{K_i}_p\mathbf{\Lambda}^i + \mathbf{f}^{K_i}_p, \tag{4.12}$$

with modified matrices,

$$\begin{bmatrix} \mathbf{U}^{K_i} \\ \mathbf{L}^{K_i} \\ \mathbf{P}^{K_i} \end{bmatrix} = -\mathbb{A}^{-1}\begin{bmatrix} \mathbf{A}^{\Omega_i}_{u\widehat{u}} \\ \mathbf{A}^{\Omega_i}_{L\widehat{u}} \\ \mathbf{A}^{\Omega_i}_{p\widehat{u}} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{f}^{K_i}_U \\ \mathbf{f}^{K_i}_L \\ \mathbf{f}^{K_i}_P \end{bmatrix} = \mathbb{A}^{-1}\begin{bmatrix} \mathbf{f}^{\Omega_i} - \mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}}\mathbf{t}^i\mathbf{T}^i_i \\ -\mathbf{A}^{\mathcal{I}_i}_{L\widetilde{u}}\mathbf{t}^i\mathbf{T}^i_i \\ -\mathbf{A}^{\mathcal{I}_i}_{p\widetilde{u}}\mathbf{t}^i\mathbf{T}^i_i \end{bmatrix} \tag{4.13}$$

and

$$\mathbb{A} = \begin{bmatrix} \left[\mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i} + \mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{uL}^{\Omega_i} + \mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right] & \left[\mathbf{A}_{up}^{\Omega_i} + \mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right] \\ \left[\mathbf{A}_{Lu}^{\Omega_i} + \mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{LL}^{\Omega_i} + \mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right] & \left[\mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right] \\ \left[\mathbf{A}_{pu}^{\Omega_i} + \mathbf{A}_{p\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{p\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right] & \left[\mathbf{A}_{p\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right] \end{bmatrix}$$

Keeping the same structure for the element by element local problem calculations in X-HDG is an advantage since just with some matrix modifications, standard HDG codes can be modified to work for X-HDG, without fitting the mesh to the interface.

4.2 Global problem for Neumann interfaces

The local problem for a Neumann cut element leads to a local solver which includes only the trace variable $\widehat{\boldsymbol{u}}$ as unknown. Analogous to the treatment of global equations in section 1.3.2, here the discretization of (4.3) leads to

$$\int_\Gamma \widehat{\boldsymbol{v}} \cdot [\![-\nu \boldsymbol{L} + p\mathbf{I}\cdot\boldsymbol{n}]\!]\, dS + 2\int_\Gamma \widehat{\boldsymbol{v}} \cdot (\{\nu\tau \boldsymbol{u}\} - \{\nu\tau\}\widehat{\boldsymbol{u}})\, dS = 0 \quad \forall \widehat{v} \in \Lambda^h,$$
$$\int_{\partial K_i} \widehat{\boldsymbol{u}} \cdot \boldsymbol{n}\, dS = 0$$

or if written in matrix form,

$$\begin{aligned} \mathbf{A}_{\widehat{u}u}^{f,L}\mathbf{u}^{L(f)} + \mathbf{A}_{\widehat{u}\boldsymbol{L}}^{f,L}\mathbf{L}^{L(f)} + \mathbf{A}_{\widehat{u}p}^{f,L}\mathbf{p}^{L(f)} \;+\; \mathbf{A}_{\widehat{u}u}^{f,R}\mathbf{u}^{R(f)} \\ +\mathbf{A}_{\widehat{u}\boldsymbol{L}}^{f,R}\mathbf{L}^{R(f)} + \mathbf{A}_{\widehat{u}p}^{f,R}\mathbf{p}^{R(f)} \;+\; \mathbf{A}_{\widehat{u}\widehat{u}}^{f}\widehat{\mathbf{u}}^f = 0 \end{aligned} \tag{4.14}$$

for $f = 1, \ldots, \mathtt{n_{fc}}$ and

$$\mathbf{A}_{\rho\widehat{u}}\boldsymbol{\Lambda}^i = 0$$

if $K_i \subset \Omega$ and for $i = 1, \ldots, \mathtt{n_{el}}$. Replacing the local solver (4.12) for the elements $K_{L(f)}$ and $K_{R(f)}$, in (4.14) for every face Γ_f, leads to a system of equations involving only the trace variable $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{n_{fc}}}$ and $\{\rho\}_i^{\mathtt{n_{el}}}$ as unknowns and can be uniquely solved. After solving the global equations elemental variables can be obtained by simply plugging in the solution $\widehat{\boldsymbol{u}}$ in the *local solver*. Analogous to standard HDG formulation, a second element by element post processing also leads to a superconvergent solution in X-HDG, for the details please refer to Remark 2.

4.3 Dirichlet interfaces

Replacing the interface condition $(-\nu\boldsymbol{\nabla}\boldsymbol{u} + p\mathbf{I})\cdot\boldsymbol{n} = \boldsymbol{g}$ in (4.1) with Dirichlet type, the Stokes problem definition changes to,

$$\begin{aligned} -\boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla}\boldsymbol{u}) + \boldsymbol{\nabla}p &= \boldsymbol{f} \quad \text{in } \Omega \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} &= \mathbf{0} \quad \text{in } \Omega \\ \boldsymbol{u} &= \boldsymbol{u}_{\mathcal{I}} \quad \text{on } \mathcal{I} \\ \boldsymbol{u} &= \boldsymbol{u}_D \quad \text{on } \partial\Omega^{ext} \\ \frac{1}{|\Omega|}\int_\Omega p\,dV &= \rho_\Omega, \end{aligned} \tag{4.15}$$

where now the mean of the pressure is set to ρ_Ω to close the problem. Accordingly, the local problem of a Dirichlet cut element takes the form,

$$\begin{aligned} \boldsymbol{L} - \boldsymbol{\nabla}\boldsymbol{u} &= \mathbf{0}, \quad \text{in } \Omega_i \\ \boldsymbol{\nabla}\cdot(-\nu\boldsymbol{L} + p\mathbf{I}) &= \boldsymbol{f}, \quad \text{in } \Omega_i \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} &= \mathbf{0}, \quad \text{in } \Omega_i \\ \boldsymbol{u} &= \boldsymbol{u}_{\mathcal{I}} \quad \text{on } \mathcal{I}_i \\ \boldsymbol{u} &= \widehat{\boldsymbol{u}} \quad \text{on } \partial\Omega_i\backslash\mathcal{I}_i \\ \frac{1}{|\Omega_i|}\int_{\Omega_i} p\,dV &= \rho_i \end{aligned} \tag{4.16}$$

Being different from Neumann case, the last equation in (4.16) is necessary for the well posedness of the local Dirichlet problem. Using the discretization spaces, $\mathcal{V}^h$ and Λ^h, defined in (1.8), and discretization of (4.16) leads to: given $\widehat{\boldsymbol{u}} \in \Lambda^h$ and $\rho_i \in \mathbb{R}$ find $\boldsymbol{u} \in [\mathcal{P}_k(\Omega_i)]^d$, $\boldsymbol{L} \in [\mathcal{P}_k(\Omega_i)]^{d\times d}$, $p \in \mathcal{P}_k(\Omega_i)$ such that,

$$\begin{aligned} \int_{\Omega_i} \boldsymbol{\nabla}\cdot(-\nu\boldsymbol{L})\cdot\boldsymbol{v}\,dV + \int_{\Omega_i}\boldsymbol{\nabla}p\cdot\boldsymbol{v}\,dV + \int_{\partial\Omega_i\backslash\mathcal{I}_i}\nu\tau(\boldsymbol{u}-\widehat{\boldsymbol{u}})\cdot\boldsymbol{v}\,dS \\ + \int_{\mathcal{I}_i}\nu\tau\boldsymbol{u}\cdot\boldsymbol{v}\,dS = \int_{\Omega_i}\boldsymbol{f}\cdot\boldsymbol{v}\,dV + \int_{\mathcal{I}_i}\nu\tau\widetilde{\boldsymbol{u}}\cdot\boldsymbol{v}\,dS \\ \int_{\Omega_i}\boldsymbol{L}:\boldsymbol{Q}\,dV + \int_{\Omega_i}(\boldsymbol{\nabla}\cdot\boldsymbol{Q})\cdot\boldsymbol{u}\,dV - \int_{\partial\Omega_i\backslash\mathcal{I}_i}(\boldsymbol{Q}\cdot\boldsymbol{n})\cdot\widehat{\boldsymbol{u}}\,dS = \int_{\mathcal{I}_i}(\boldsymbol{Q}\cdot\boldsymbol{n})\cdot\widetilde{\boldsymbol{u}}\,dS \\ \int_{\Omega_i}\boldsymbol{u}\cdot\boldsymbol{\nabla}q\,dV - \int_{\partial\Omega_i\backslash\mathcal{I}_i}(\widehat{\boldsymbol{u}}\cdot\boldsymbol{n})q\,dS = \int_{\mathcal{I}_i}(\widetilde{\boldsymbol{u}}\cdot\boldsymbol{n})q\,dS \\ \frac{1}{|\Omega_i|}\int_{\Omega_i} p\,dV = \rho_i \end{aligned} \tag{4.17}$$

for all $\boldsymbol{v} \in [\mathcal{P}_k(\Omega_i)]^d$, $\boldsymbol{Q} \in [\mathcal{P}_k(\Omega_i)]^{d\times d}$ and $q \in \mathcal{P}_k(\Omega_i)$. The above weak from is derived analogous to Neumann case however now the integral terms along the interface, $\mathcal{I}_i$

including $\widetilde{\boldsymbol{u}}$, is no more an unknown so they are simply moved to the r.h.s to the equations.

Writing the (4.17) in matrix form, following elemental system is obtained

$$\begin{bmatrix} \mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i} & \mathbf{A}_{uL}^{\Omega_i} & \mathbf{A}_{up}^{\Omega_i} \\ \mathbf{A}_{Lu}^{\Omega_i} & \mathbf{A}_{LL}^{\Omega_i} & \mathbf{0} \\ \mathbf{A}_{pu}^{\Omega_i} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^i \\ \mathbf{L}^i \\ \mathbf{p}^i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u^{K_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i} \\ \mathbf{A}_{L\widetilde{u}}^{\mathcal{I}_i} \\ \mathbf{A}_{p\widetilde{u}}^{\mathcal{I}_i} \end{bmatrix} - \begin{bmatrix} \mathbf{A}_{u\widehat{u}}^{\Omega_i} \\ \mathbf{A}_{L\widehat{u}}^{\Omega_i} \\ \mathbf{A}_{p\widehat{u}}^{\Omega_i} \end{bmatrix} \boldsymbol{\Lambda}^i \tag{4.18}$$

Equation (4.18) in the way that it is, is defined up to a constant. To have a system with unique solution discrete form of last equation in (4.17) i.e, $\mathbf{A}_{\rho p}^{\Omega_i}\mathbf{p}^i = \rho_i$ is added to the system with Lagrange multipliers to obtain,

$$\left[\begin{array}{ccc:c} \left[\mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i}\right] & \mathbf{A}_{uL}^{\Omega_i} & \mathbf{A}_{up}^{\Omega_i} & \mathbf{0} \\ \mathbf{A}_{Lu}^{\Omega_i} & \mathbf{A}_{LL}^{\Omega_i} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{pu}^{\Omega_i} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^{\Omega_i T} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^{\Omega_i} & \mathbf{0} \end{array}\right] \begin{bmatrix} \mathbf{u}^i \\ \mathbf{L}^i \\ \mathbf{p}^i \\ \hdashline \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u^{K_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i} \\ \mathbf{A}_{L\widetilde{u}}^{\mathcal{I}_i} \\ \mathbf{A}_{p\widetilde{u}}^{\mathcal{I}_i} \\ \hdashline \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \hdashline \mathbf{1} \end{bmatrix} \rho_i - \begin{bmatrix} \mathbf{A}_{u\widehat{u}}^{\Omega_i} \\ \mathbf{A}_{L\widehat{u}}^{\Omega_i} \\ \mathbf{A}_{p\widehat{u}}^{\Omega_i} \\ \hdashline \mathbf{0} \end{bmatrix} \boldsymbol{\Lambda}^i \tag{4.19}$$

where $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$ are the vectors of nodal values of $\boldsymbol{u}$, $\boldsymbol{L}$ and p and $\boldsymbol{\Lambda}^i$ is the vector of nodal values of $\widehat{\boldsymbol{u}}$ on the n faces of the element as defined in (1.12). Finally the system (4.19) can be solved for $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$ for each Dirichlet cut element, leading to the *local solver* that is,

$$\begin{gathered} \mathbf{u}^i = \mathbf{S}_u^{\Omega_i}\boldsymbol{\Lambda}^i + \mathbf{R}_u^{\Omega_i}\rho_i + \mathbf{f}_U^{\Omega_i}, \qquad \mathbf{L}^i = \mathbf{S}_L^{\Omega_i}\boldsymbol{\Lambda}^i + \mathbf{R}_L^{\Omega_i}\rho_i + \mathbf{f}_L^{\Omega_i} \\ \mathbf{p}^i = \mathbf{S}_p^{\Omega_i}\boldsymbol{\Lambda}^i + \mathbf{R}_p^{\Omega_i}\rho_i + \mathbf{f}_p^{\Omega_i}, \end{gathered} \tag{4.20}$$

with,

$$\begin{bmatrix} \mathbf{S}_u^{\Omega_i} \\ \mathbf{S}_L^{\Omega_i} \\ \mathbf{S}_p^{\Omega_i} \\ \mathbf{S}_\lambda^{\Omega_i} \end{bmatrix} = -\mathbb{A}^{-1} \begin{bmatrix} \mathbf{A}_{u\widehat{u}}^{\Omega_i} \\ \mathbf{A}_{L\widehat{u}}^{\Omega_i} \\ \mathbf{A}_{p\widehat{u}}^{\Omega_i} \\ \mathbf{0} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{R}_u^{\Omega_i} \\ \mathbf{R}_L^{\Omega_i} \\ \mathbf{R}_p^{\Omega_i} \\ \mathbf{R}_\lambda^{\Omega_i} \end{bmatrix} = \mathbb{A}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{f}_U^{\Omega_i} \\ \mathbf{f}_L^{\Omega_i} \\ \mathbf{f}_P^{\Omega_i} \\ \mathbf{f}_\lambda^{\Omega_i} \end{bmatrix} = \mathbb{A}^{-1} \begin{bmatrix} \mathbf{f}_u^{K_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i} \\ \mathbf{A}_{L\widetilde{u}}^{\mathcal{I}_i} \\ \mathbf{A}_{p\widetilde{u}}^{\mathcal{I}_i} \\ \mathbf{0} \end{bmatrix} \tag{4.21}$$

and

$$\mathbb{A} = \begin{bmatrix} \left[\mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i}\right] & \mathbf{A}_{uL}^{\Omega_i} & \mathbf{A}_{up}^{\Omega_i} & \mathbf{0} \\ \mathbf{A}_{Lu}^{\Omega_i} & \mathbf{A}_{LL}^{\Omega_i} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{pu}^{\Omega_i} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^{\Omega_i T} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^{\Omega_i} & \mathbf{0} \end{bmatrix}$$

It is important to note that the local solver has a very similar structure for a standard element (1.25), for a Neumann cut element (4.12), or for a Dirichlet cut element (4.20), expressing the elemental variables $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$ in terms of or only the trace variable $\widehat{\boldsymbol{u}}$, or the trace variable together with the elemental pressure constant ρ_i. Now the problem reduces to determine the trace variable $\widehat{\boldsymbol{u}}$ and ρ_i, which is done using global equations (4.3), analogous to standard HDG.

4.4 Global problem for Dirichlet interfaces

With the addition of $\sum \dfrac{|K_i|}{|\Omega|}\rho_i = \rho_\Omega$ term in (4.3c), global equations for Dirichlet interfaces are,

$$\begin{aligned}
[\![(-\nu\boldsymbol{L} + p\mathbf{I})\cdot\boldsymbol{n}]\!] &= 0 \quad \text{on } \Gamma\backslash\partial\Omega^{ext},\\
\int_{\partial K_i\cap\bar{\Omega}} \widehat{\boldsymbol{u}}\cdot\boldsymbol{n}\,dS &= \begin{cases} 0 & \text{if } K_i\subset\Omega \\ -\displaystyle\int_{\mathcal{I}_i}\widetilde{\boldsymbol{u}}\cdot\boldsymbol{n}\,dS & \text{if } K_i\cap\mathcal{I} \end{cases}\\
\widehat{\boldsymbol{u}} &= \mathbb{P}_2(\boldsymbol{u}_D) \quad \text{on } \partial\Omega^{ext} \text{ and } \sum\frac{|K_i|}{|\Omega|}\rho_i = \rho_\Omega.
\end{aligned}$$

Writing the first equation also in weak form following system is obtained,

$$\begin{aligned}
&\int_\Gamma \widehat{\boldsymbol{v}}\cdot[\![(-\nu\boldsymbol{L}+p\mathbf{I})\cdot\boldsymbol{n}]\!]\,dS + 2\int_\Gamma \widehat{\boldsymbol{v}}\cdot(\{\nu\tau\boldsymbol{u}\} - \{\nu\tau\}\widehat{\boldsymbol{u}})\,dS = 0 \quad \forall\widehat{v}\in\Lambda^h,\\
&\int_{\partial K_i\cap\bar{\Omega}} \widehat{\boldsymbol{u}}\cdot\boldsymbol{n}\,dS = \begin{cases} 0 & \text{if } K_i\subset\Omega \\ -\displaystyle\int_{\mathcal{I}_i}\widetilde{\boldsymbol{u}}\cdot\boldsymbol{n}\,dS & \text{if } K_i\cap\mathcal{I} \end{cases}
\end{aligned} \tag{4.23}$$

Here, being different from (1.20b) the incompressibility condition defined on the element faces includes the new term $\displaystyle\int_{\mathcal{I}_i}\widetilde{\boldsymbol{u}}\cdot\boldsymbol{n}\,dS$ that corresponds to the Dirichlet face. Since $\widetilde{\boldsymbol{u}}$ is not an unknown, the new equation simply appears on r.h.s. The discrete form of above system leads to,

$$\begin{aligned}
\mathbf{A}^{f,L}_{\widehat{u}u}\mathbf{u}^{L(f)} + \mathbf{A}^{f,L}_{\widehat{u}\boldsymbol{L}}\mathbf{L}^{L(f)} + \mathbf{A}^{f,L}_{\widehat{u}p}\mathbf{p}^{L(f)} \;+\; \mathbf{A}^{f,R}_{\widehat{u}u}\mathbf{u}^{R(f)}(f)\\
+\mathbf{A}^{f,R}_{\widehat{u}\boldsymbol{L}}\mathbf{L}^{R(f)} + \mathbf{A}^{f,R}_{\widehat{u}p}\mathbf{p}^{R(f)} \;+\; \mathbf{A}^{f}_{\widehat{u}\widehat{u}}\widehat{\mathbf{u}}^f = 0
\end{aligned} \tag{4.24}$$

for $f = 1,\ldots,\mathtt{n_{fc}}$ and

$$\mathbf{A}_{\rho\widehat{u}}\mathbf{\Lambda}^i = \begin{cases} 0 & \text{if } K_i\subset\Omega \\ -\mathbf{A}^{\mathcal{I}_i}_{\rho\widetilde{u}}\widetilde{\mathbf{u}}^i & \text{if } K_i\cap\mathcal{I} \end{cases}$$

for $i = 1, \ldots, \mathtt{n_{el}}$. The rest, analogous to standard HDG, is done by replacing the local solver for the elements $K_{L(f)}$ and $K_{R(f)}$, in (4.24) for every face Γ_f. At the end a system of equations involving only the trace variable $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{n_{fc}}}$ and $\{\rho\}_i^{\mathtt{n_{el}}}$ as unknowns is obtained, and it can be uniquely solved only after setting outer boundary conditions and pressure constraint.

Remark 19. The extension of X-HDG formulation for Stokes problem where mixed boundary conditions are considered at the outer boundary is straightforward. The suitable local problem needs to be chosen and solved for each and every element; then, resulting local solver needs to be assembled to the global system defined by the first equation in (4.3). The second and third equations needs to be tailored, depending on the problem of interest. In case of having Dirichlet interfaces, the r.h.s of second equation in (4.3) should be set to $-\int_{\mathcal{I}_i} \widetilde{\boldsymbol{u}} \cdot \boldsymbol{n}\, dS$; if not simply to zero. Moreover, the third equation in (4.3) needs to be redefined according to the outer boundary conditions of interest. It is relevant to note that in the global system no pressure constraint is needed since in case of mixed boundaries the problem is no more pure Dirichlet type.

4.5 Numerical Examples

Performance of X-HDG is tested over following numerical examples where both Neumann and Dirichlet type boundary conditions are considered at the interface. The last example includes a kidney-shaped interface, demonstrating the necessity of proper interface definition to ensure optimum convergence.

In all numerical tests, the stabilization parameter is set to $\tau = 1$ on all faces, and the pressure is defined as

$$p = x + y + \mathbb{C}.$$

4.5.1 Circular Stokes void

The Stokes problem (4.1) is solved over a square domain, ν set to unity, and a circular void of radius 0.41 is located at the center that is, $\Omega = (-1, 1)^2 \backslash B((0, 0), 0.41)$. First Neumann, then Dirichlet boundary conditions are imposed on the void boundary $\mathcal{I} = \partial B((0, 0), 0.41)$, while Dirichlet boundary conditions are considered on the exterior boundary $\partial\Omega\backslash\mathcal{I} = \partial\left((0, 1)^2\right)$ in both cases. The analytical solution is a smooth

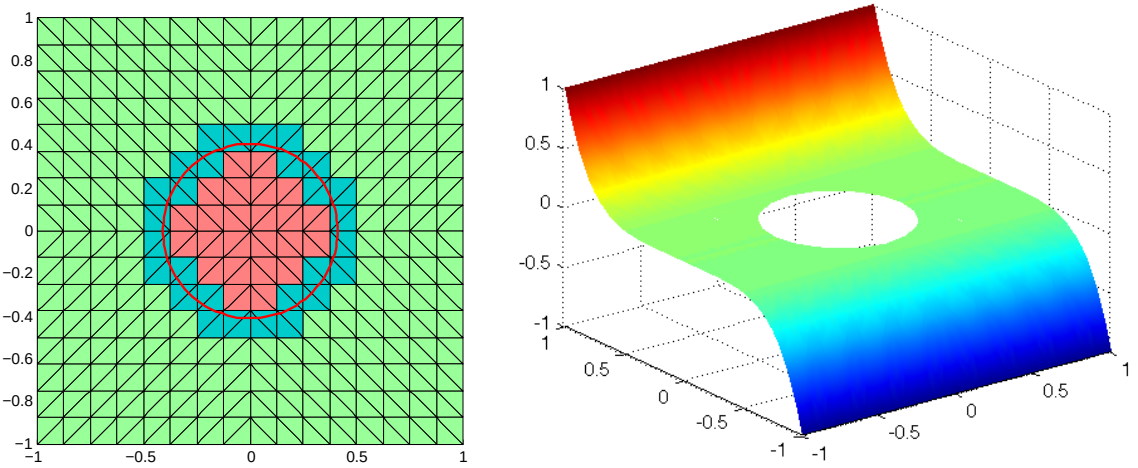


Figure 4.1: Circular Stokes void: computational mesh after two mesh refinements and the circular interface (left) the x component of the analytical solution (right).

polynomial of the form,

$$u(x, y) = (y^5, x^5)^T$$

The computational domain after two mesh refinements and the x component of the analytical solution is shown in Figure 4.1. As usual, a computational mesh covering the domain Ω and fitting the exterior boundary $\partial\Omega\backslash\mathcal{I}$ is considered. A level set function is used to describe the interface, $\mathcal{I}$. Computations are done for the approximation order changing between $k = 2$ and $k = 4$ and the mesh is refined four times. Figure 4.2 shows the evolution of the $\mathcal{L}_2$ error for decreasing mesh size, for both the solution and the postprocessed superconvergent solution. With this toy example it is simply demonstrated that X-HDG keeps the optimal convergence with rates offered by HDG method without mesh fitting. It is interesting to observe that for finest mesh and highest approximation degree, with Neumann boundary, accuracy is bounded by machine errors.

4.5.2 Kidney shaped void with Neumann boundary

Square domain of size $\Omega = (-1, 1)^2$, with $\nu = 1$, is cut by a kidney-shaped interface defined by

$$\varphi(x, y) = \big(3((x + 0.5)^2 + y^2) - x - 0.5\big)^2 - \big((x + 0.5)^2 + y^2\big) + 0.1.$$

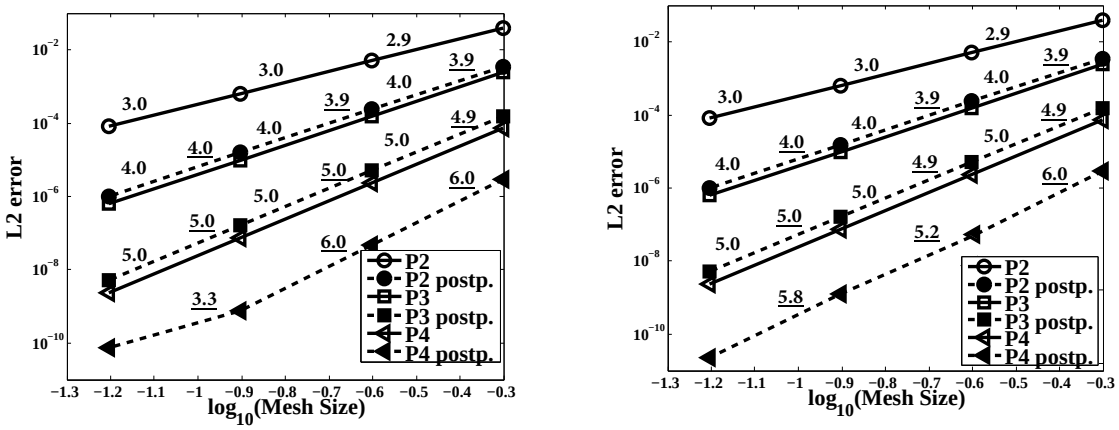


Figure 4.2: Circular Stokes void: convergence history of X-HDG. Neumann boundary condition is imposed at the interface $\mathcal{I}$ (left). Dirichlet boundary condition is imposed at the interface $\mathcal{I}$ (right). The numbers are correspond to the slope of each segment, and are underlined for postprocessed solution.

Neumann boundary conditions are considered at the interface $\mathcal{I}$. At the outer boundary Dirichlet conditions are imposed. The analytical solution reads,

$$u(x, y) = (0, \cos(1 - x^2))^T$$

The analytical solution and the computational mesh with interface $\mathcal{I}$ is shown in Figure 4.3. Computations are done for the approximation order changing between $k = 1$ and $k = 4$ and the mesh is refined four times. The level set is approximated by the same degree of approximation used for the analysis. Figure 4.4 indicates that X-HDG covers optimal and super convergence dealing with rather complex interfaces. Yet again, for the finest mesh and the highest degree of approximation, propogation of round of errors dominates the post processed solution.

4.6 Conclusions and final remarks

X-HDG for Stokes problem with void interfaces has been proposed. As usual, the local problem is modified from standard HDG formulation for the elements and faces cut by the interface. For the rest of the elements standard HDG formulation recalled in section 1.3.2 is utilized. At every cut element, an auxiliary trace variable on the boundary is introduced, which is eliminated afterwards using the boundary conditions

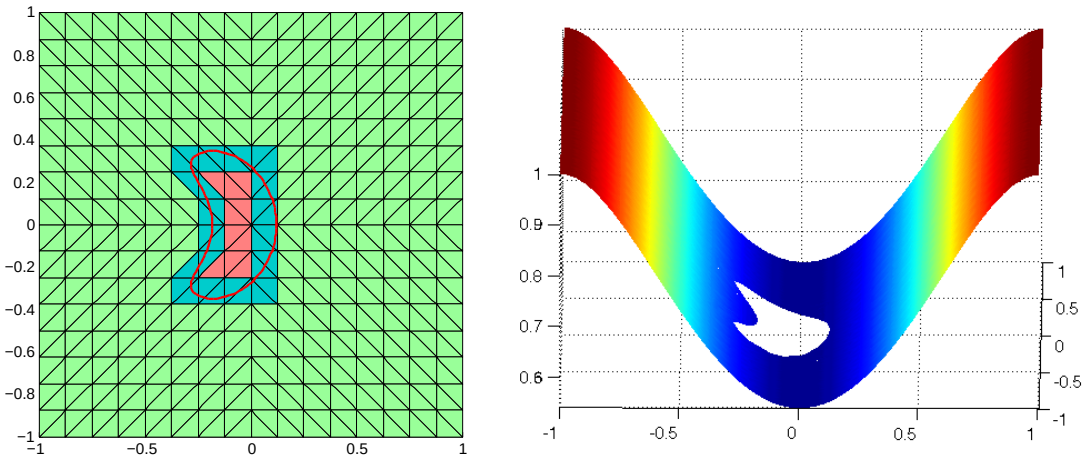


Figure 4.3: Kidney shaped void with Neumann boundary: computational mesh after two mesh refinements and the kidney-shaped interface (left) the y component of the analytical solution (right).

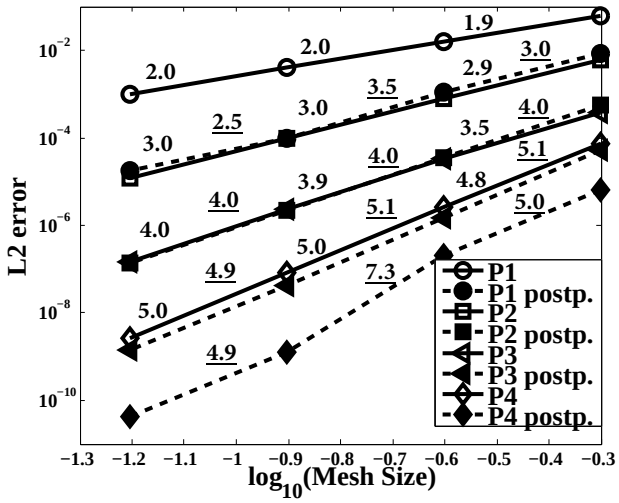


Figure 4.4: Kidney shaped void with Neumann boundary: convergence history of X-HDG. Slopes corresponding to solution $\boldsymbol{u}$ is shown in bold, whereas slopes corresponding to the post processed solution $\boldsymbol{u}^*$ is shown in bold and underlined.

on the interface, keeping the original unknowns and the structure of the local problem solver.

Pressure is represented by a constant in each element, causing no extra dof to solve the Stokes problem. Depending on the boundary conditions, for the well posedness of the problem, a compatibility condition is imposed.

Numerical examples are considered at the end to verify the X-HDG formulation. A toy problem of a curved interface with two different boundary conditions is examined.

Last but not least, X-HDG is tested with a rather complicated kidney-shaped interface over a square domain. All numerical tests suggest that X-HDG keeps the HDG optimal convergence rates for the solution, the gradient and the post-processed super-convergent solution, without the need to adapt the mesh to the boundary.

The X-HDG method is fully developed in this chapter for the solution of Stokes problem with void interfaces, highlighting the main ideas of the method. Next, in Chapter 5, Stokes problem will further be investigated using bimaterial interfaces.

Chapter 5

X-HDG for Stokes Problem - Bimaterial Interfaces

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain divided in two disjoint subdomains

$$\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2, \quad \Omega_1 \cap \Omega_2 = \emptyset$$

with an interface

$$\mathcal{I} = \overline{\Omega}_1 \cap \overline{\Omega}_2.$$

The following Stokes bimaterial problem is considered,

$$\begin{aligned} -\boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} \boldsymbol{u}) + \boldsymbol{\nabla} p &= \boldsymbol{f}, \quad \text{in } \Omega_1 \cup \Omega_2 \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} &= \mathbf{0}, \quad \text{in } \Omega_1 \cup \Omega_2 \\ [\![\boldsymbol{u} \otimes \boldsymbol{n}]\!] &= \mathbf{0}, \quad \text{on } \mathcal{I} \\ [\![(-\nu \boldsymbol{\nabla} \boldsymbol{u} + p\mathbf{I}) \cdot \boldsymbol{n}]\!] &= \mathbf{0}, \quad \text{on } \mathcal{I} \\ \boldsymbol{u} &= \boldsymbol{u}_D \quad \text{on } \partial\Omega \end{aligned} \tag{5.1}$$

where $\boldsymbol{u}$ is the velocity, p is the pressure, ν is the material viscosity discontinuous across the interface (that is, $\nu = \nu_i$ in Ω_i for $i = 1, 2$), $\mathbf{I}$ is the second-order identity tensor, $\boldsymbol{f}$ is an external force and $\boldsymbol{u}_D$ are prescribed values on the exterior boundary. The jump operator is defined in (1.7). The domain Ω is now assumed to be covered by a finite element mesh with elements $\{K_i\}_{i=1}^{\mathtt{nel}}$ and faces $\{\Gamma_i\}_{i=1}^{\mathtt{nfc}}$ satifying the conditions in (1.2) and (1.3). The problem (5.1) is now written in terms of first order PDE's as

usual that is,

$$\left.\begin{aligned} \boldsymbol{L}-\boldsymbol{\nabla}\boldsymbol{u}&=\boldsymbol{0} && \text{in } K_i\\ \boldsymbol{\nabla}\cdot(-\nu\boldsymbol{L}+p\mathbf{I})&=\boldsymbol{f} && \text{in } K_i\\ \boldsymbol{\nabla}\cdot\boldsymbol{u}&=\boldsymbol{0} && \text{in } K_i\\ \boldsymbol{u}&=\widehat{\boldsymbol{u}} && \text{on } \partial K_i\\ \frac{1}{|K_i|}\int_{K_i} p\,dV&=\rho_e \end{aligned}\right\} \text{ if } \mathcal{I}\cap K_i=\emptyset \tag{5.2a}$$

$$\left.\begin{aligned} \boldsymbol{L}-\boldsymbol{\nabla}\boldsymbol{u}&=\boldsymbol{0} && \text{in } K_i\backslash\mathcal{I}_i\\ \boldsymbol{\nabla}\cdot(-\nu\boldsymbol{L}+p\mathbf{I})&=\boldsymbol{f} && \text{in } K_i\backslash\mathcal{I}_i\\ \boldsymbol{\nabla}\cdot\boldsymbol{u}&=\boldsymbol{0} && \text{in } K_i\backslash\mathcal{I}_i\\ [\![\boldsymbol{u}\otimes\boldsymbol{n}]\!]&=\boldsymbol{0} && \text{on } \mathcal{I}_i\\ [\![(-\nu\boldsymbol{\nabla}\boldsymbol{u}+p\mathbf{I})\cdot\boldsymbol{n}]\!]&=\boldsymbol{0} && \text{on } \mathcal{I}_i\\ \boldsymbol{u}&=\widehat{\boldsymbol{u}} && \text{on } \partial K_i\\ \frac{1}{|K_i|}\int_{K_i} p\,dV&=\rho_i \end{aligned}\right\} \text{ if } \mathcal{I}\cap K_i\neq\emptyset \tag{5.2b}$$

together with the global equations

$$[\![(-\nu\boldsymbol{L}+p\mathbf{I})\cdot\boldsymbol{n}]\!]=0 \quad \text{on } \Gamma\backslash\partial\Omega, \tag{5.3a}$$

$$\int_{\partial K_i}\widehat{\boldsymbol{u}}\cdot\boldsymbol{n}\,dS=0 \quad \text{for } i=1,\ldots,\mathtt{n_{el}}, \tag{5.3b}$$

$$\widehat{\boldsymbol{u}}=\mathbb{P}_2(\boldsymbol{u}_D) \quad \text{on } \partial\Omega \text{ and } \sum\frac{|K_i|}{|\Omega|}\rho_i=\rho_\Omega. \tag{5.3c}$$

In these equations the variable $\boldsymbol{L}$ corresponds to the gradient of $\boldsymbol{u}$, $\widehat{\boldsymbol{u}}$ corresponds to the trace of $\boldsymbol{u}$ at the mesh faces Γ, and ρ_i corresponds to the scalar mean of pressure at the element. Similar to Stokes for void interfaces, (5.2a) and (5.2b) are called local problems and they are discretized and solved to obtain a local solver for each element. After that, interelement connnectivities are ensured by (5.3a). (5.3b) and (5.3c) are added to impose the incompressibility condition and to set the outer boundary condition together with the pressure constraint to ensure the well posedness, respectively.

As usual, equations (5.2a) correspond to standard elements their discretization is detailed in Section 1.3.2 wheras equations (5.2b) correspond to cut elements and their discretization is explained next.

The spaces used for discretization $\mathcal{V}^h_{enr}$ and Λ^h_{enr} are defined in (3.5) and considered for elemental variables $\boldsymbol{u}$, $\boldsymbol{L}$ and p, and for the trace variable $\widehat{\boldsymbol{u}}$, respectively. It is

important to note that for the discretization of cut elements enriched basis functions are utilized to be able to capture the discontinuities within the element. On the other hand, for standard elements, standard unenriched basis functions are used.

5.1 Local problem for cut elements

X-HDG local problem at cut element K_i leads to the discretization of (5.2b), that is given $\widehat{\boldsymbol{u}} \in \Lambda^h$ and ρ_i find $\boldsymbol{u} \in [\mathcal{V}^h_{enr}]^d$, $\boldsymbol{L} \in [\mathcal{V}^h_{enr}]^{d\times d}$ and $p \in \mathcal{V}^h_{enr}$ such that,

$$\begin{aligned}
\int_{K_i} \boldsymbol{\nabla}\cdot(-\nu\boldsymbol{L})\cdot\boldsymbol{v}\,dV + \int_{K_i}\boldsymbol{\nabla}p\cdot\boldsymbol{v}\,dV + \int_{\partial K_i\setminus\mathcal{I}_i}\nu\tau(\boldsymbol{u}-\widehat{\boldsymbol{u}})\cdot\boldsymbol{v}\,dS \\
+2\int_{\mathcal{I}_i}(\{\tau\nu\boldsymbol{u}\}-\{\tau\nu\}\widetilde{\boldsymbol{u}})\cdot\boldsymbol{v}\,dS = \int_{K_i}\boldsymbol{f}\cdot\boldsymbol{v}\,dV \\
\int_{K_i}\boldsymbol{L}:\boldsymbol{Q}\,dV + \int_{K_i}(\boldsymbol{\nabla}\cdot\boldsymbol{Q})\cdot\boldsymbol{u}\,dV - \int_{\partial K_i\setminus\mathcal{I}_i}(\boldsymbol{Q}\cdot\boldsymbol{n})\cdot\widehat{\boldsymbol{u}}\,dS - \int_{\mathcal{I}_i}[\![(\boldsymbol{Q}\cdot\boldsymbol{n})]\!]\cdot\widetilde{\boldsymbol{u}}\,dS = 0 \\
\int_{K_i}\boldsymbol{u}\cdot\boldsymbol{\nabla}q\,dV - \int_{\partial K_i\setminus\mathcal{I}_i}(\widehat{\boldsymbol{u}}\cdot\boldsymbol{n})q\,dS - \int_{\mathcal{I}_i}[\![q\boldsymbol{n}]\!]\cdot\widetilde{\boldsymbol{u}}\,dS = 0 \\
\frac{1}{|K_i|}\int_{K_i}p\,dV = \rho_i
\end{aligned} \tag{5.4}$$

for all $\boldsymbol{v} \in [\mathcal{V}^h_{enr}]^d$, $\boldsymbol{Q} \in [\mathcal{V}^h_{enr}]^{d\times d}$, $q \in \mathcal{V}^h_{enr}$. Equations (5.4) are very similar to the weak form derived for void interfaces (4.4). The derivation process is as well similar i.e. the first equation in (5.4) is derived by applying integration by parts, replacing the velocity gradient by the numerical velocity gradient (1.21) and undoing the integration by parts. The second and third equations in (5.2b) are derived by applying integration by parts and replacing the boundary condition $\boldsymbol{u} = \widehat{\boldsymbol{u}}$ on the element boundary in (5.2b). The integrals over interface $\mathcal{I}$ includes the new variable $\widetilde{\boldsymbol{u}}$ approximating the trace of the solution on the interface $\mathcal{I}_i := K_i \cap \mathcal{I}$. With the new variable $\widetilde{\boldsymbol{u}}$ the weak continuity of $\boldsymbol{u}$ is ensured across the interface by weakly setting the condition $\boldsymbol{u} = \widetilde{\boldsymbol{u}}^i$ on $\mathcal{I}_i$. Looking at it in detail one can see that (5.4) is simply obtained by taken into account the discontinuities of ν, $\boldsymbol{u}$ and $\boldsymbol{v}$ across $\mathcal{I}_i$ using the mean operator $\{\cdot\}$ which is defined in (1.15). Actually (5.4) is derived simply summing the weak form obtained for void interfaces (4.4), for each one of the domains inside the cut element, $K_i \cap \Omega_1$ and $K_i \cap \Omega_2$.

Discretization of (5.4) leads to the following system of equations

$$\begin{cases} \left[\mathbf{A}_{uu}^{K_i} + \mathbf{A}_{uu}^{\mathcal{I}_i}\right]\mathbf{u}^i + \mathbf{A}_{uL}^{K_i}\mathbf{L}^i + \mathbf{A}_{up}^{K_i}\mathbf{p}^i + \mathbf{A}_{u\widehat{u}}^{K_i}\mathbf{\Lambda}^i + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\widetilde{\mathbf{u}}^i = \mathbf{f}_u^{K_i} \\ \mathbf{A}_{Lu}^{K_i}\mathbf{u}^i + \mathbf{A}_{LL}^{K_i}\mathbf{L}^i + \mathbf{A}_{L\widehat{u}}^{K_i}\mathbf{\Lambda}^i + \mathbf{A}_{L\widetilde{u}}^{\mathcal{I}_i}\widetilde{\mathbf{u}}^i = \mathbf{0} \\ \mathbf{A}_{pu}^{K_i}\mathbf{u}^i + \mathbf{A}_{p\widehat{u}}^{K_i}\mathbf{\Lambda}^i + \mathbf{A}_{p\widetilde{u}}^{\mathcal{I}_i}\widetilde{\mathbf{u}}^i = \mathbf{0} \end{cases} \tag{5.5}$$

before adding the pressure constraint to (5.5), let us eliminate $\widetilde{\boldsymbol{u}}$ from the system using the continuity of trace on the interface which is

$$[\![(-\nu\boldsymbol{L} + p\mathbf{I})\cdot\boldsymbol{n}]\!] = \mathbf{0} \quad \text{on } \mathcal{I}_i. \tag{5.6}$$

The weak form of (5.6) is then obtained by applying integration by parts and replacing the velocity gradient with the numerical one defined in (1.21) that is,

$$\int_{\mathcal{I}_i} [\![(-\nu\boldsymbol{L}\cdot\boldsymbol{n})]\!]\cdot\widetilde{\boldsymbol{v}}\,dS + \int_{\mathcal{I}_i} [\![(p\boldsymbol{n})]\!]\cdot\widetilde{\boldsymbol{v}}\,dS + \int_{\mathcal{I}_i} 2(\{\tau\nu\boldsymbol{u}\} - \{\tau\nu\}\widetilde{\boldsymbol{u}})\cdot\widetilde{\boldsymbol{v}}\,dS = \mathbf{0} \tag{5.7}$$

for all $\widetilde{\boldsymbol{v}} \in \mathcal{P}_k(\mathcal{I}_i)$. Writing (5.7) in matrix form

$$\mathbf{A}_{\widetilde{u}^i u}^{\mathcal{I}_i}\boldsymbol{u}^i + \mathbf{A}_{\widetilde{u}^i L}^{\mathcal{I}_i}\mathbf{L}^i + \mathbf{A}_{\widetilde{u}^i p}^{\mathcal{I}_i}\mathbf{p}^i + \mathbf{A}_{\widetilde{u}^i\widetilde{u}^i}^{\mathcal{I}_i}\widetilde{\boldsymbol{u}}^i = \mathbf{0}$$

now $\widetilde{\boldsymbol{u}}$ can be written in terms of elemental unknowns $\boldsymbol{u}^i$, $\boldsymbol{L}^i$ and $\mathbf{p}^i$ such that

$$\widetilde{\boldsymbol{u}}^i = \mathbf{T}_u^i\mathbf{u}^i + \mathbf{T}_L^i\mathbf{L}^i + \mathbf{T}_p^i\mathbf{p}^i \tag{5.8}$$

where

$$\mathbf{T}_u^i = \left[-\mathbf{A}_{\widetilde{u}^i\widetilde{u}^i}^{\mathcal{I}_i}\right]^{-1}\mathbf{A}_{\widetilde{u}^i u}^{\mathcal{I}_i}, \quad \mathbf{T}_{\boldsymbol{L}}^i = \left[-\mathbf{A}_{\widetilde{u}^i\widetilde{u}^i}^{\mathcal{I}_i}\right]^{-1}\mathbf{A}_{\widetilde{u}^i L}^{\mathcal{I}_i}, \quad \mathbf{T}_p^i = \left[-\mathbf{A}_{\widetilde{u}^i\widetilde{u}^i}^{\mathcal{I}_i}\right]^{-1}\mathbf{A}_{\widetilde{u}^i p}^{\mathcal{I}_i}$$

Replacing the expression of $\widetilde{\boldsymbol{u}}$ (5.8) into (5.5) the local system of equations becomes

$$\begin{aligned} &\left[\mathbf{A}_{uu}^{K_i} + \mathbf{A}_{uu}^{\mathcal{I}_i} + \mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_u^i\right]\mathbf{u}^i + \left[\mathbf{A}_{uL}^{K_i} + \mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right]\mathbf{L}^i \\ &\qquad + \left[\mathbf{A}_{up}^{K_i} + \mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right]\mathbf{p}^i + \mathbf{A}_{u\widehat{u}^i}^{\mathcal{I}_i}\widehat{\mathbf{u}}^i = \mathbf{f} \\ &\left[\mathbf{A}_{Lu}^{K_i} + \mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_u^i\right]\mathbf{u}^i + \left[\mathbf{A}_{LL}^{K_i} + \mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right]\mathbf{L}^i + \left[\mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right]\mathbf{p}^i + \mathbf{A}_{L\widehat{u}^i}^{\mathcal{I}_i}\widehat{\mathbf{u}}^i = \mathbf{0} \\ &\left[\mathbf{A}_{pu}^{K_i} + \mathbf{A}_{p\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_u^i\right]\mathbf{u}^i + \left[\mathbf{A}_{p\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right]\mathbf{L}^i + \left[\mathbf{A}_{p\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right]\mathbf{p}^i + \mathbf{A}_{p\widehat{u}^i}^{\mathcal{I}_i}\widehat{\mathbf{u}}^i = \mathbf{0}. \end{aligned} \tag{5.9}$$

For (5.9) to have a unique solution, the discrete from of the pressure constraint – the last equation in (5.4) – $\mathbf{A}_{\rho p}^{K_i}\mathbf{p}^i = \rho_i$ is added to the system with Lagrange multipliers

to obtain,

$$\left[\begin{array}{ccc:c} \left[\mathbf{A}_{uu}^{K_i}+\mathbf{A}_{uu}^{\mathcal{I}_i}+\mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{uL}^{K_i}+\mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right] & \left[\mathbf{A}_{up}^{K_i}+\mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right] & \mathbf{0} \\ \left[\mathbf{A}_{Lu}^{K_i}+\mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{LL}^{K_i}+\mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right] & \left[\mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right] & \mathbf{0} \\ \left[\mathbf{A}_{pu}^{K_i}+\mathbf{A}_{p\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{p\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right] & \left[\mathbf{A}_{p\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right] & \mathbf{A}_{\rho p}^{K_i T} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^{K_i} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \mathbf{u}^i \\ \mathbf{L}^i \\ \mathbf{p}^i \\ \hdashline \lambda \end{array}\right] \cdots$$

$$\cdots = \left[\begin{array}{c} \boldsymbol{f} \\ \mathbf{0} \\ \mathbf{0} \\ \hdashline \mathbf{0} \end{array}\right] + \left[\begin{array}{c} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \hdashline \mathbf{1} \end{array}\right] \rho_i - \left[\begin{array}{c} \mathbf{A}_{u\widehat{u}^i}^{\mathcal{I}_i} \\ \mathbf{A}_{L\widehat{u}}^{\mathcal{I}_i} \\ \mathbf{A}_{p\widehat{u}^i}^{\mathcal{I}_i} \\ \hdashline \mathbf{0} \end{array}\right] \boldsymbol{\Lambda}^i \tag{5.10}$$

System (5.10) might seem scary in the first sight, however; it is very similar to the system obtained for standard elements using standard HDG setting (1.24) just with some additional matrices derived from the elimination of $\widetilde{\boldsymbol{u}}$. Solving the (5.10) for $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$ for each element, *local solver* is obtained as

$$\begin{gathered} \mathbf{u}^i = \mathbf{S}_u^{K_i}\boldsymbol{\Lambda}^i + \mathbf{R}_u^{K_i}\rho_i + \mathbf{f}_U^{K_i}, \qquad \mathbf{L}^i = \mathbf{S}_L^{K_i}\boldsymbol{\Lambda}^i + \mathbf{R}_L^{K_i}\rho_i + \mathbf{f}_L^{K_i} \\ \mathbf{p}^i = \mathbf{S}_p^{K_i}\boldsymbol{\Lambda}^i + \mathbf{R}_p^{K_i}\rho_i + \mathbf{f}_p^{K_i}, \end{gathered} \tag{5.11}$$

with

$$\left[\begin{array}{c} \mathbf{S}_u^{K_i} \\ \mathbf{S}_L^{K_i} \\ \mathbf{S}_p^{K_i} \\ \mathbf{S}_\lambda^{K_i} \end{array}\right] = -\mathbb{A}^{-1}\left[\begin{array}{c} \mathbf{A}_{u\widehat{u}}^{K_i} \\ \mathbf{A}_{L\widehat{u}}^{K_i} \\ \mathbf{A}_{p\widehat{u}}^{K_i} \\ \mathbf{0} \end{array}\right], \quad \left[\begin{array}{c} \mathbf{R}_u^{K_i} \\ \mathbf{R}_L^{K_i} \\ \mathbf{R}_p^{K_i} \\ \mathbf{R}_\lambda^{K_i} \end{array}\right] = \mathbb{A}^{-1}\left[\begin{array}{c} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{array}\right], \quad \left[\begin{array}{c} \mathbf{f}_U^{K_i} \\ \mathbf{f}_L^{K_i} \\ \mathbf{f}_P^{K_i} \\ \mathbf{f}_\lambda^{K_i} \end{array}\right] = \mathbb{A}^{-1}\left[\begin{array}{c} \mathbf{f} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{array}\right] \tag{5.12}$$

and

$$\mathbb{A}\left[\begin{array}{ccc:c} \left[\mathbf{A}_{uu}^{K_i}+\mathbf{A}_{uu}^{\mathcal{I}_i}+\mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{uL}^{K_i}+\mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right] & \left[\mathbf{A}_{up}^{K_i}+\mathbf{A}_{u\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right] & \mathbf{0} \\ \left[\mathbf{A}_{Lu}^{K_i}+\mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{LL}^{K_i}+\mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right] & \left[\mathbf{A}_{L\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right] & \mathbf{0} \\ \left[\mathbf{A}_{pu}^{K_i}+\mathbf{A}_{p\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{p\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_L^i\right] & \left[\mathbf{A}_{p\widetilde{u}^i}^{\mathcal{I}_i}\mathbf{T}_p^i\right] & \mathbf{A}_{\rho p}^{K_i T} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^{K_i} & \mathbf{0} \end{array}\right]$$

After obtaining the local solver where the elemental variables $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$ are expressed in terms of only $\boldsymbol{\Lambda}^i$ and ρ_i; using the global equations (5.3) $\boldsymbol{\Lambda}^i$ and ρ_i is

determined and the problem is closed. Next, the details of the discretization of global problem is presented.

Remark 20. Being analogous to Remark 16, the extension of the formulation where discontinuous conditions considered at the interface is evident. In example 5.3.2 such situation is examined. Consider a problem with non-homogeneous conditions on the interface, $\boldsymbol{u}_2 - \boldsymbol{u}_1 = \boldsymbol{\alpha}$ and $[\![(-\nu\boldsymbol{\nabla}\boldsymbol{u} + p\mathbf{I})\cdot\boldsymbol{n}]\!] = \mathbf{g}$ on $\mathcal{I}_i$, where $\boldsymbol{u}_i$ denotes the restriction of function $\boldsymbol{u}$ to Ω_i. Then, defining $\widetilde{\boldsymbol{u}}^i$ as the trace of $\boldsymbol{u}_1$ (i.e., $\boldsymbol{u}_1 = \widetilde{\boldsymbol{u}}^i$ on $\mathcal{I}_i$, we have $\boldsymbol{u}_2 = \widetilde{\boldsymbol{u}}^i + \boldsymbol{\alpha}$ on $\mathcal{I}_i$, and therefore, new terms $\int_{\mathcal{I}_i}\tau\nu_2(\boldsymbol{v}_2\cdot\boldsymbol{\alpha})\,dS$, $-\int_{\mathcal{I}_i}(\boldsymbol{Q}_2\cdot\boldsymbol{n}_2)\cdot\boldsymbol{\alpha}\,dS$ and $-\int_{\mathcal{I}_i}(q\boldsymbol{n})\cdot\boldsymbol{\alpha}\,dS$ appear in the right hand side of the first, second and third equations in (5.4), respectively, and the right hand side of (5.7) becomes $\int_{\mathcal{I}_i}\widetilde{\boldsymbol{u}}\cdot\mathbf{g}\,dS + \int_{\mathcal{I}_i}\tau\nu_2(\widetilde{\boldsymbol{u}}\cdot\boldsymbol{\alpha})\,dS$. Furthermore, the compatibility condition (5.3b) modifies to $\displaystyle\int_{\partial K_i}\widehat{\boldsymbol{u}}\cdot\boldsymbol{n}\,dS + \int_{\mathcal{I}_i}\boldsymbol{\alpha}\cdot\boldsymbol{n}\,dS = 0$

5.2 Global problem

The local problem in a standard element, or in a cut element leads to a local solver of the form (5.11) where the only unknowns are $\widehat{\boldsymbol{u}}$ and $\{\rho\}_i^{\mathtt{n_{el}}}$. As usual to determine the unknowns on interelement boundaries Γ, global equations (5.3a) and (5.3b) are discretized and $\boldsymbol{L}$ is replaced by the numerical flux (1.21) to obtain

$$\begin{aligned}\mathbf{A}^{f,L}_{\widehat{u}u}\mathbf{u}^{L(f)} + \mathbf{A}^{f,L}_{\widehat{u}\boldsymbol{L}}\mathbf{L}^{L(f)} + \mathbf{A}^{f,L}_{\widehat{u}p}\mathbf{p}^{L(f)} \;+\; \mathbf{A}^{f,R}_{\widehat{u}u}\mathbf{u}^{R(f)}(f)&\\ +\mathbf{A}^{f,R}_{\widehat{u}\boldsymbol{L}}\mathbf{L}^{R(f)} + \mathbf{A}^{f,R}_{\widehat{u}p}\mathbf{p}^{R(f)} \;+\; \mathbf{A}^{f}_{\widehat{u}\widehat{u}}\widehat{\mathbf{u}}^{f} = 0&\\ \mathbf{A}_{\rho\widehat{u}}\boldsymbol{\Lambda}^i = 0 \text{ for } i = 1,\ldots,\mathtt{n_{el}}&\end{aligned} \tag{5.13}$$

Plugging in the local solver (1.25) for standard or (5.11) for cut elements for the terms $K_{L(f)}$ and $K_{R(f)}$, in (5.13) for every face Γ_f; a system of equations involving only the trace variable $\{\widehat{\mathbf{u}}^f\}_{f=1}^{\mathtt{n_{fc}}}$ and $\{\rho\}_i^{\mathtt{n_{el}}}$ is obtained. After solving the global system, elemental variables can be recovered by simply placing $\widehat{\boldsymbol{u}}$ and $\{\rho\}_i^{\mathtt{n_{el}}}$ in the *local solver*. Analogous to standard HDG formulation, a second element by element post processing also leads to a superconvergent solution in X-HDG for bimaterial setting, for the details please refer to Remark 18.

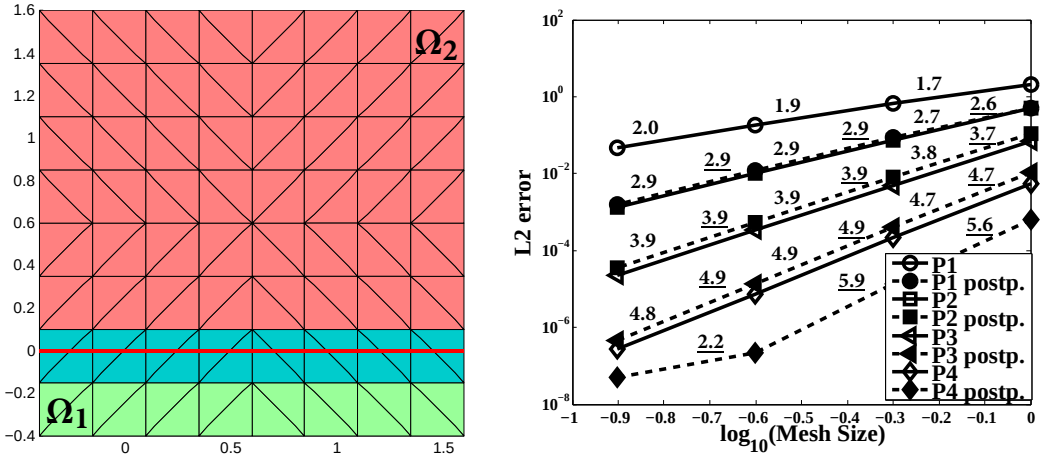


Figure 5.1: A continuous problem, linear interface: computational mesh after two mesh refinements and the linear interface (left). Convergence history (right).

5.3 Numerical examples

Numerical examples with both continuous and discontinuous solution considered at the interface are examined next; to verify the applicability of X-HDG formula for Stokes bimaterial problem derived previously. In both examples the pressure is set to

$$p = x + y + \mathbb{C}.$$

5.3.1 A continuous problem, linear interface

Influenced by Hansbo et al. [2014], a continuous problem is solved over a square domain $\Omega = (-0.4, 1.6)^2$ where the interface is a straight line passing through $y = 0$. Material parameters are set to $\nu_1 = 1$ at Ω_1 —standard elements sits below the interface— and to $\nu_2 = 2$ at Ω_2 see Figure 5.1 left panel. Dirichlet type outer boundary $\partial\Omega$ is considered where the analytic solution —see Figure 5.2— is,

$$u(x,y) = \begin{cases} ((x^5y)/\nu_1 \quad , \quad (-2.5x^4y^2)/\nu_1)^T & \text{in } \Omega_1, \\ ((x^5y)/\nu_2 \quad , \quad (-2.5x^4y^2)/\nu_2)^T & \text{in } \Omega_2 \end{cases}$$

For the convergence studies, approximation degree is varied between $k = 1$ and $k = 4$ and four mesh refinements are considered. Figure 5.1 right panel shows that as expected optimum and super convergence is achieved. For $k = 4$ and finest mesh, machine error limit the convergence.

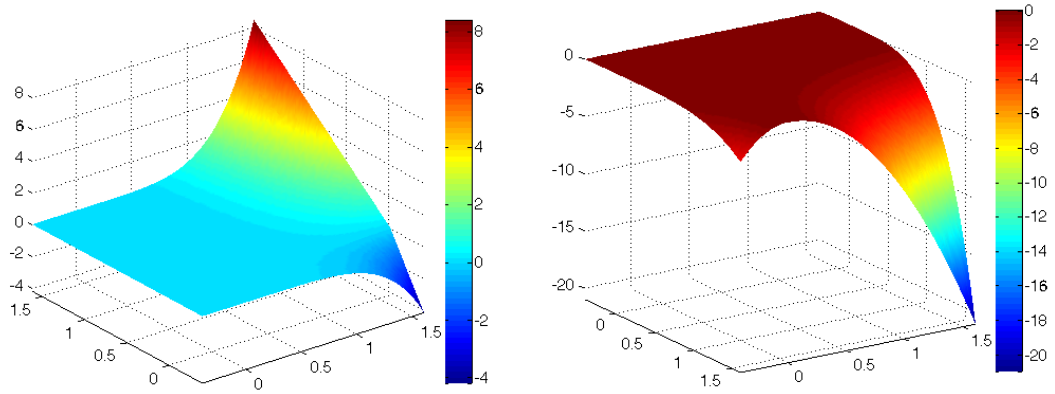


Figure 5.2: A continuous problem, linear interface: Analytic solution in x direction (left). Analytic solution in y direction (right)

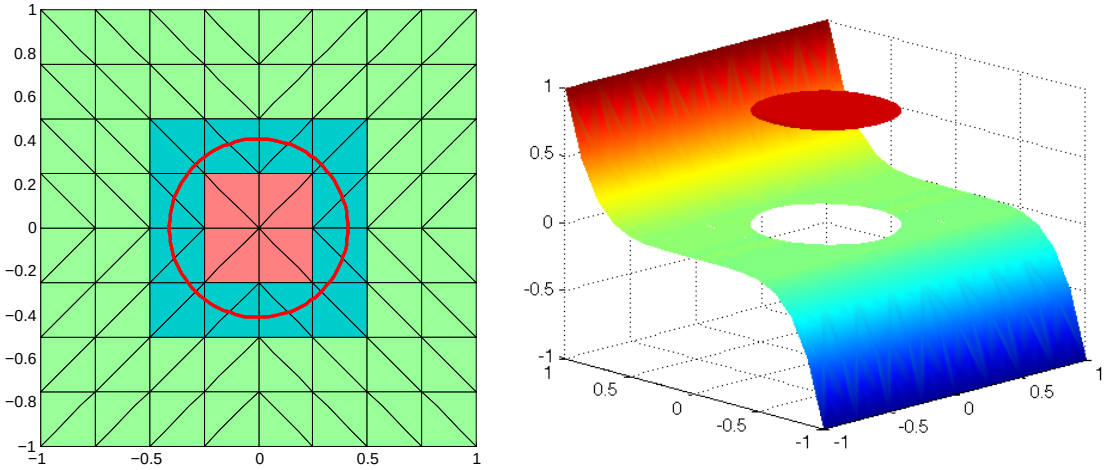


Figure 5.3: A discontinuous problem, curved interface: computational mesh after two mesh refinements and the circular interface (left), the discontinuous analytic solution (right).

5.3.2 A discontinuous problem, curved interface

Stokes interface problem defined in (5.1) is solved over square domain of size $\Omega = (-1,1)^2$. Circular interface of radius $R = 0.41$ divides the computational domain into two regions; Ω_2 is the area enclosed by the interface with material parameter $\nu_2 = 1$ whereas Ω_1 is the area outside the circle with $\nu_1 = 10$ see Figure 5.3. Outer boundary $\partial\Omega\backslash\mathcal{I}$ is set to be Dirichlet type where the analytical solution reads,

$$u(x,y) = \begin{cases} (y^5, x^5)^T & \text{in } \Omega_1, \\ (\cos(1-y^5), \sin(1-x^5))^T & \text{in } \Omega_2 \end{cases}$$

It can easily be seen that neither the solution nor the traction is continuous at the interface $\mathcal{I}$. Following the strategy explained in Remark 20 the discontinuous

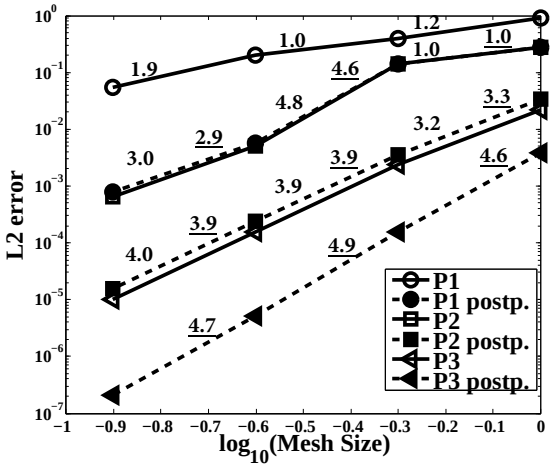


Figure 5.4: A discontinuous problem, curved interface: Convergence of X-HDG solving a discontinuous bimaterial problem.

problem is handled easily with X-HDG. Elements of degree varying between $k = 1$ and $k = 3$ are utilized for convergence studies and the computational mesh is refined four consecutive times. The convergence history is presented in Figure 5.4 demonstrating that the optimum and super convergence rates are achieved as expected.

5.4 Conclusions and final remarks

X-HDG for Stokes problem with bimaterial interfaces has been presented. Analogous to previous chapters, a modified local problem is derived in section 5.1 and applied on cut elements whereas the standard elements are treated using standard HDG formulation. At the end the local solver structure and the original unknowns are kept the same as standard HDG. Pressure is represented by a constant in each element, leading no extra dof to solve. Depending on the boundary conditions, for the well posedness of the problem, a compatibility condition is imposed. Numerical examples of both continuous and discontinuous nature are considered at section 5.3 verifying the applicability of X-HDG as well as demonstrating that X-HDG keeps the optimum and super convergence properties of standard HDG without mesh fitting.

X-HDG is presented for the solution of Stokes problem with bimaterial interfaces, underlying the differences as well as the similarities to standard HDG formulation. Next, X-HDG for the moving boundaries is exhibited.

Chapter 6

X-HDG for Moving Interfaces

6.1 X-HDG for Moving Interfaces

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain divided in two disjoint subdomains

$$\overline{\Omega} = \overline{\Omega}_1(t) \cup \overline{\Omega}_2(t), \quad \Omega_1(t) \cap \Omega_2(t) = \emptyset$$

with an interface

$$\mathcal{I}(t) = \overline{\Omega}_1(t) \cap \overline{\Omega}_2(t).$$

The following time dependent heat equation is considered,

$$\begin{aligned} \frac{\partial u}{\partial t} - \boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} u) &= f \quad \text{in } \Omega_1(t) \cup \Omega_2(t), \\ [\![u\boldsymbol{n}]\!] &= 0 \quad \text{on } \mathcal{I}(t), \\ [\![\nu \boldsymbol{n} \cdot \boldsymbol{\nabla} u]\!] &= 0 \quad \text{on } \mathcal{I}(t), \\ u &= u_D \quad \text{on } \Gamma_D = \partial\Omega \\ u(x,0) &= u_0(x) \quad \text{in } \Omega_1(t) \cup \Omega_2(t), \end{aligned} \tag{6.1}$$

where $u(x,t)$ is the velocity depends both on spatial coordinate x and time t. Similarly the interface $\mathcal{I}(t)$ and $\Omega_i(t)$ are evolving in time. ν is the material coefficient with discontinuous definition across the interface (that is, $\nu = \nu_i$ in Ω_i for $i = 1, 2$), f is a given source term and u_D are prescribed values on the boundary. Above problem statement is fairly similar to (3.1) with a time derivative term and interface $\mathcal{I}(t)$ that evolves in time.

The domain Ω is assumed to be covered by a finite element mesh as usual with $\mathtt{n_{el}}$ disjoint elements K_i and $\mathtt{n_{fc}}$ number of faces as described in (1.2) and (1.3), respectively.

Equation (6.1) is now expressed as a first order system; some element by element local equations and some global equations defined in interelement boundaries Γ similar to section 3.1. The local element-by-element problems correspond to the statement of the PDE in (6.1), with Dirichlet boundary conditions, in each element K_i is,

$$\left.\begin{aligned} \frac{\partial u}{\partial t} + \boldsymbol{\nabla}\cdot\boldsymbol{q} &= f \quad \text{in } K_i \\ \boldsymbol{q} + \nu\boldsymbol{\nabla}u &= 0 \quad \text{in } K_i \\ u &= \widehat{u} \quad \text{on } \partial K_i \end{aligned}\right\} \text{ if } \mathcal{I}\cap K_i = \emptyset, \tag{6.2a}$$

$$\left.\begin{aligned} \frac{\partial u}{\partial t} + \boldsymbol{\nabla}\cdot\boldsymbol{q} &= f \quad \text{in } K_i\backslash\mathcal{I} \\ \boldsymbol{q} + \nu\boldsymbol{\nabla}u &= 0 \quad \text{in } K_i\backslash\mathcal{I} \\ [\![u\boldsymbol{n}]\!] &= 0 \quad \text{on } \mathcal{I}\cap K_i \\ [\![\boldsymbol{q}\cdot\boldsymbol{n}]\!] &= 0 \quad \text{on } \mathcal{I}\cap K_i \\ u &= \widehat{u} \quad \text{on } \partial K_i \end{aligned}\right\} \text{ if } \mathcal{I}\cap K_i \neq \emptyset, \tag{6.2b}$$

The global equations and the boundary conditions are the same as in steady case see (3.3) and (3.4), respectively. The global equations (3.3) together with the local problems (6.2), with the boundary condition (3.4), leads to the complete X-HDG formulation.

Sections 6.2 and 6.3 state the discretization of the local problem for standard and cut elements, respectively. The discretization of global equations is analogous to section 3.1.3.

6.2 Local problem for standard elements

Using the spaces defined in (3.5) for standard elements, the discretization of local problem (6.2a) stated such as given the $\widehat{u}\in\Lambda^h$, find $u\in\mathcal{P}_k(K_i)$ and $\boldsymbol{q}\in[\mathcal{P}_k(K_i)]^d$

$$\begin{aligned} \int_{K_i}\frac{\partial u}{\partial t}v\,dV + \int_{K_i} v\boldsymbol{\nabla}\cdot\boldsymbol{q}\,dV + \int_{\partial K_i}\tau\nu v(u-\widehat{u})\,dS &= \int_{K_i} vf\,dV \;\; \forall v\in\mathcal{P}_k(K_i) \\ \int_{K_i}\boldsymbol{q}\cdot\boldsymbol{w}\,dV - \int_{K_i}\nu u\boldsymbol{\nabla}\cdot\boldsymbol{w}\,dV + \int_{\partial K_i}\nu\widehat{u}\boldsymbol{w}\cdot\boldsymbol{n}\,dS &= 0 \;\; \forall\boldsymbol{w}\in[\mathcal{P}_k(K_i)]^d \end{aligned} \tag{6.3}$$

Equations (6.3) are similar to (1.9) and are derived analogously. Here, the only difference is the time dependent approximation of u i.e., the first term in (6.3) which is further discretized using Backward Euler time discretization such as:

$$\int_{K_i} \frac{\partial u}{\partial t} v \, dV \simeq \int_{K_i} \frac{1}{\Delta t} u^n v \, dV - \int_{K_i} \frac{1}{\Delta t} u^{n-1} v \, dV \tag{6.4}$$

where the superscript n stands for the approximation of u at the time instant t^n. In (6.4) the term $\int_{K_i} \frac{1}{\Delta t} u^{n-1} v \, dV$ is known since it includes the solution from previous time step $n-1$; it can simply be move to the r.h.s of the (6.3), however, the term $\int_{K_i} \frac{1}{\Delta t} u^n v \, dV$ stays as an unknown on the l.h.s. since it is an unknown at the current time instant n. The final system then takes the shape,

$$\begin{gathered} \int_{K_i} \frac{1}{\Delta t} u^n v \, dV + \int_{K_i} v \boldsymbol{\nabla} \cdot \boldsymbol{q}^n \, dV + \int_{\partial K_i} \tau v (u^n - \widehat{u}^n) \, dS = \int_{K_i} v f^n \, dV + \int_{K_i} \frac{1}{\Delta t} u^{n-1} v \, dV \\ \int_{K_i} \boldsymbol{q}^n \cdot \boldsymbol{w} \, dV - \int_{K_i} \nu u^n \boldsymbol{\nabla} \cdot \boldsymbol{w} \, dV + \int_{\partial K_i} \nu \widehat{u}^n \boldsymbol{w} \cdot \boldsymbol{n} \, dS = 0 \end{gathered} \tag{6.5}$$

in matrix form the final system is

$$\begin{cases} \left[\mathbf{A}_{uu}^{K_i} + \mathbf{A}_{uu^n}^{K_i} \right] (\mathbf{u}^n)^i + \mathbf{A}_{uq}^{K_i} (\mathbf{q}^n)^i + \mathbf{A}_{u\widehat{u}}^{K_i} (\boldsymbol{\Lambda}^n)^i = \mathbf{f}_u^{K_i} + \mathbf{A}_{uu^{n-1}}^{K_i} (\mathbf{u}^{n-1})^i \\ \mathbf{A}_{qu}^{K_i} (\mathbf{u}^n)^i + \mathbf{A}_{qq}^{K_i} (\mathbf{q}^n)^i + \mathbf{A}_{q\widehat{u}}^{K_i} (\boldsymbol{\Lambda}^n)^i = \mathbf{0} \end{cases} \tag{6.6}$$

where $(\mathbf{u}^n)^i$ and $(\mathbf{q}^n)^i$ are the vectors of nodal values of u and $\boldsymbol{q}$ in element K_i at current time instant n, and $(\boldsymbol{\Lambda}^n)^i$ is the vector of nodal values of $\widehat{u}$ on the faces of the element. Now the system (6.6) can be solved for $(\mathbf{u}^n)^i$ and $(\mathbf{q}^n)^i$ to obtain the *local solver* in the element K_i that is identical to (1.13) with,

$$\begin{bmatrix} \mathbf{U}^{K_i} \\ \mathbf{Q}^{K_i} \end{bmatrix} = -\mathbb{A}^{-1} \begin{bmatrix} \mathbf{A}_{u\widehat{u}}^{K_i} \\ \mathbf{A}_{q\widehat{u}}^{K_i} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{f}_U^{K_i} \\ \mathbf{f}_Q^{K_i} \end{bmatrix} = \mathbb{A}^{-1} \begin{bmatrix} \mathbf{f}_u^{K_i} + \mathbf{A}_{uu^{n-1}}^{K_i} \\ \mathbf{0} \end{bmatrix}$$

and

$$\mathbb{A} = \begin{bmatrix} \left[\mathbf{A}_{uu}^{K_i} + \mathbf{A}_{uu^n}^{K_i} \right] & \mathbf{A}_{uq}^{K_i} \\ \mathbf{A}_{qu}^{K_i} & \mathbf{A}_{qq}^{K_i} \end{bmatrix}$$

6.3 Local problem for cut elements

Using the spaces defined in (3.5) for cut elements i.e. spaces with Heaviside enrichment, the discretization of local problem (6.2b) leads to

$$\begin{aligned}\int_{K_i}\frac{\partial u}{\partial t}v\,dV+\int_{K_i}v\boldsymbol{\nabla}\cdot\boldsymbol{q}\,dV+\int_{\partial K_i}\tau\nu\;v(u-\widehat{u})\,dS+2\int_{\mathcal{I}_i}\tau\{\nu v(u-\widetilde{u})\}\,dS=\int_{K_i}vf\,dV\\ \int_{K_i}\frac{1}{\nu}\boldsymbol{q}\cdot\boldsymbol{w}\,dV-\int_{K_i}u\boldsymbol{\nabla}\cdot\boldsymbol{w}\,dV+\int_{\partial K_i}\widehat{u}\boldsymbol{w}\cdot\boldsymbol{n}\,dS+\int_{\mathcal{I}_i}\widetilde{u}[\![\boldsymbol{w}\cdot\boldsymbol{n}]\!]\,dS=0\end{aligned}\tag{6.7}$$

for given $\widehat{u}\in\Lambda^h$ to find $u\in\mathcal{P}_k(K_i)\oplus H\mathcal{P}_k(K_i)$, $\boldsymbol{q}\in[\mathcal{P}_k(K_i)\oplus H\mathcal{P}_k(K_i)]^d$ for all $v\in\mathcal{P}_k(K_i)\oplus H\mathcal{P}_k(K_i)$ and $\boldsymbol{w}\in[\mathcal{P}_k(K_i)\oplus H\mathcal{P}_k(K_i)]^d$. Variables appear in (6.7) are standard for X-HDG formulation and are very well known now so no further explanation is added here. The derivation of the weak form is done analogous to (3.6). Using Backward Euler method to discretize the time integral term, the final system takes the form

$$\begin{aligned}\int_{K_i}\frac{1}{\Delta t}u^nv\,dV+\int_{K_i}v\boldsymbol{\nabla}\cdot\boldsymbol{q}^n\,dV+\int_{\partial K_i}\tau\nu\;v(u^n-\widehat{u}^n)\,dS\\ +2\int_{\mathcal{I}_i}\tau\{\nu v(u^n-\widetilde{u}^n)\}\,dS=\int_{K_i}vf^n\,dV+\int_{K_i}\frac{1}{\Delta t}u^{n-1}v\,dV\\ \int_{K_i}\frac{1}{\nu}\boldsymbol{q}^n\cdot\boldsymbol{w}\,dV-\int_{K_i}u^n\boldsymbol{\nabla}\cdot\boldsymbol{w}\,dV+\int_{\partial K_i}\widehat{u}^n\boldsymbol{w}\cdot\boldsymbol{n}\,dS+\int_{\mathcal{I}_i}\widetilde{u}^n[\![\boldsymbol{w}\cdot\boldsymbol{n}]\!]\,dS=0\end{aligned}\tag{6.8}$$

Eliminating the unknown $\widetilde{u}$ analogous to section 3.1.2 using the weak form of the conservativity condition across $\mathcal{I}(t)$ (3.8), local solver for cut elements can be obtained that is identical to (1.13) with

$$\begin{bmatrix}\mathbf{U}^{K_i}\\ \mathbf{Q}^{K_i}\end{bmatrix}=-\mathbb{A}^{-1}\begin{bmatrix}\mathbf{A}^{K_i}_{u\widehat{u}}\\ \mathbf{A}^{K_i}_{q\widehat{u}}\end{bmatrix},\quad\begin{bmatrix}\mathbf{f}^{K_i}_U\\ \mathbf{f}^{K_i}_Q\end{bmatrix}=\mathbb{A}^{-1}\begin{bmatrix}\mathbf{f}^{K_i}_u+\mathbf{A}^{K_i}_{uu^{n-1}}\\ \mathbf{0}\end{bmatrix}\tag{6.9}$$

and

$$\mathbb{A}=\begin{bmatrix}\left[\mathbf{A}^{K_i}_{uu}+\mathbf{A}^{K_i}_{uu^n}+\mathbf{A}^{\mathcal{I}_i}_{uu}+\mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}}\mathbf{T}^i_u\right] & \left[\mathbf{A}^{K_i}_{uq}+\mathbf{A}^{\mathcal{I}_i}_{u\widetilde{u}}\mathbf{T}^i_q\right]\\ \left[\mathbf{A}^{K_i}_{qu}+\mathbf{A}^{\mathcal{I}_i}_{q\widetilde{u}}\mathbf{T}^i_u\right] & \left[\mathbf{A}^{K_i}_{qq}+\mathbf{A}^{\mathcal{I}_i}_{q\widetilde{u}}\mathbf{T}^i_q\right]\end{bmatrix}.$$

Note that similar to section 3.1, both standard and cut elements lead to a local solver of the same form, allowing the expression of elemental unknowns in terms of the trace unknown. To determine the trace unknown and set the outer boundary conditions, the global equations are used their discretization is identical to section 3.1.3.

6.4 Evolving the interface

The strategy used to update the level set function of the Stefan problem studied in section 6.5.2 is explained here. Osher and Sethian [1988] presented extensive studies on how to propagate interfaces defined by a level set function $\Phi(x,t)$ by means of following transport equation

$$\frac{\partial \Phi}{\partial t} + V_n^{ext} \|\boldsymbol{\nabla} \Phi\| = 0 \tag{6.10}$$

where $V_n^{ext}(x,t)$ is the speed function in the outward normal direction given in the entire domain of Φ. However, this information is not always available. For the Stefan problem, the velocity V_n is known only at the interface at its discrete points, but not in the entire domain of Φ. Luckily, there are different techniques to extend this interface velocity to the whole domain of Φ. A fast marching method is presented in Ji et al. [2002], on the other hand, "closest point projection" idea is followed here (also presented in Mourad et al. [2005]) to define the so called *extended velocity* V_n^{ext}. First the closest point projection $\mathbf{p}$ of each node $\mathbf{x}_i$ onto the interface is calculated and then the extension velocity V_n^{ext} simply set to

$$V_n^{ext}(\mathbf{x}_i) = V_n(\mathbf{p}(\mathbf{x}_i))$$

Having the extension velocity V_n^{ext} at hand, the level set function can now be advanced solving (6.10), however Mourad et al. [2005] studied the advantages of assuming the signed-distance is maintained from the outset and used

$$\frac{\partial \Phi}{\partial t} + V_n^{ext} = 0 \tag{6.11}$$

to advance the level set function Φ. It is proved that for sufficiently smooth Φ and V_n^{ext}, the signed distance is actually preserved so easier to handle (6.11) can replace (6.10). In this document, transport equation (6.11) is used to advance the Φ.

6.5 Numerical examples

Two numerical examples are considered next, verifying the behavior of X-HDG for moving interfaces. First a circular interface advancing with unit speed is considered. Lastly, two-phase Stefan problem is solved where the interface velocity is defined by the gradient jump, using X-HDG and comparisons are done with X-FEM solution of the problem.

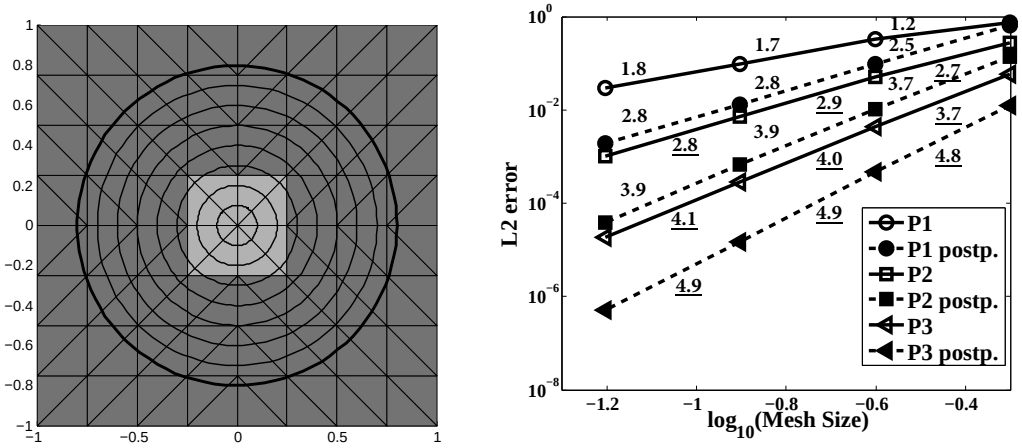


Figure 6.1: Circle collapsing at unit speed: Interface location every 10 time steps with $\Delta t = 0.01$ for $t \in [0, 0.7]$ (left) and convergence history (right). Initial position is shown with a thicker line, gray elements are cut elements for $t = 0.7$.

6.5.1 Circle collapsing with unit speed

The time-dependent heat equation (6.1) is solved over a square domain $\Omega = (-1, 1)^2$ split into two subdomains, $\overline{\Omega} = \overline{\Omega_1} \cup \overline{\Omega_2}$, with an evolving circular interface $\mathcal{I}(t)$. The time-dependent radius is

$$R(t) = 0.8 - t,$$

and material parameters are $\nu_2 = 5$ in $\Omega_2(t) = \{(x, y) \in \Omega | x^2 + y^2 < R(t)^2\}$ and $\nu_1 = 1$ in $\Omega_1 = \Omega \backslash \overline{\Omega_2}$. Boundary conditions and source term are set such that the analytical solution is

$$u(x,t) = \begin{cases} \frac{1}{\nu_1}(x^2+y^2)^{5/2} & \text{in } \Omega_1(t), \\ \frac{1}{\nu_2}(x^2+y^2)^{5/2} + (\frac{1}{\nu_1} - \frac{1}{\nu_2})R(t)^5 & \text{in } \Omega_2(t). \end{cases}$$

It is important to note that in this example, the enriched approximation spaces at time $n-1$ and time n are not the same due to the fact that the interface location changes. Therefore, the Heaviside enrichment function is different. To facilitate the implementation of the discrete system, the solution u^{n-1} is interpolated to the approximation space of time t^n at every time step in this work, so that the enriched approximation space at time n can be considered for the discretization of all terms in the equations. Furthermore, the interface velocity V_n^{ext} is set to be known and unity, so no closest point projection is needed to determine the extended velocity. The calculations are done on a sequence of four uniform meshes with changing approximation degree k. The time step size $\Delta t = 0.01$ leads to a time error that is

smaller than the spatial discretization errors. Figure 6.1 shows the location of the interface every ten time steps, for $t \in [0, 0.7]$, and the h-convergence history. Optimal convergence rates are recorded for both the solution and the superconvergent solution, showing the applicability and good performance of X-HDG for problems with moving interfaces.

6.5.2 Two-phase Stefan problem

Last but not least, two-phase Stefan problem is considered. Stefan problem models the freezing of a semi-infinite domain and is defined as,

$$\begin{aligned} c\frac{\partial T}{\partial t} - \boldsymbol{\nabla} \cdot (\kappa \boldsymbol{\nabla} T) = f \quad & \text{in } \Omega_1(t) \cup \Omega_2(t), \\ T = T_{melting} \quad & \text{on } \mathcal{I}(t), \\ T = T_{left} \quad & \text{at } x = 0, \\ T = T_{right} \quad & \text{at } x = 10, \\ T(x,0) = T_0(x) \quad & \text{in } \Omega_1(t) \cup \Omega_2(t). \end{aligned} \tag{6.12}$$

Problem (6.12) is solved over a domain of size $\Omega = [0, 10]$x$[0, 1.5]$. On the left edge of the rectangular domain, the temperature T is set to a constant below melting temperature $T_{left} = -10 < T_{melting}$ whereas on the right edge the temperature is a constant above melting temperature $T_{right} = 4 > T_{melting}$, driving the freezing process, while the linear interface separating ice and water is moving towards the right edge of the domain. On the top and bottom edges, homogenous Neumann boundary conditions are imposed. It is important to note that, here, the condition imposed at the interface $\mathcal{I}(t)$ is Dirichlet type, so in local equation (6.7), terms including $\widetilde{u}$ can simply be moved to the right hand side, setting the value for $\widetilde{u}$ to $T_{melting}$.

To evolve the interface, strategy explained in section 6.4 is followed. The velocity V_n at discrete points on the interface $\mathcal{I}(t)$ is defined by,

$$V_n = [\![\kappa \boldsymbol{\nabla} T \cdot \boldsymbol{n}]\!]/L$$

where L is the volumetric latent heat of fusion (cal/cm^3), constant value being equal to 19.2 and $\boldsymbol{n}$ is the normal vector pointing liquid domain. To advance the interface, problem (6.12) is solved and heat flux jump is calculated to obtain V_n, interface velocity, then it is extended to the domain to obtain V_n^{ext}. Finally, the level set function defining the interface is updated using (6.11) at each time step.

The analytical solution describing the interface motion $\mathcal{I}(t) = x_{front}(t)$ is well known and given by

$$x_{front}(t) = 2\lambda\sqrt{\beta_1 t}$$

where $\beta_1 = \kappa_1/c_1$ is the thermal diffusivity of ice; t is time; and λ is a constant being equal to 0.3073 for this problem. Other constants used for ice/solid domain Ω_1 and water/liquid domain Ω_2 can be listed as; volumetric heat capacity $(cal/°Ccm^3)$, $c_1 = 0.49$, $c_2 = 0.62$, the thermal conductivity $(cal/cm\ s\ °C)$ $\kappa_1 = 9.6\text{x}10^{-3}$, $\kappa_2 = 6.9\text{x}10^{-3}$, and the melting temperature $(°C)$ $T_{melting} = 0$. The ratio of thermal diffusivities is indicated by $\eta = \beta_1/\beta_2$. The temperature field in ice/solid phase where $x < x_{front}$ is defined by

$$T = T_{left} + \frac{T_{melting} - T_{left}}{\text{erf}(\lambda)}\,\text{erf}(\frac{x}{2\sqrt{\beta_1 t}})$$

and in water/liquid phase where $x > x_{front}$

$$T = T_{right} - \frac{T_{right} - T_{melting}}{\text{erfc}(\lambda\sqrt{\eta})}\,\text{erfc}(\frac{x}{2\sqrt{\beta_2 t}}).$$

Knowing the analytical expression for temperature fields, convergence studies are done using different time step size and its effect over convergence properties of X-HDG is studied. Four meshes are considered with changing element size from $h = 1$ to $h = 0.125$ where the approximation degree is varied from $k = 1$ to $k = 3$. It is worth to note that, similar to collapsing circle example, the solution T^{n-1} is interpolated to the approximation space of time t^n to facilitate the implementation. Starting from $t_{initial} = 185$ and running the simulations till $t_{final} = 185.1$ time steps, Figure 6.2 shows the convergence behavior of X-HDG for different time step sizes. The effect of time step size can easily be recognized. Time errors are dominant for solution gradient even for smallest Δt chosen, showing the need of an advanced time discretization technique to get the most of X-HDG, removing the time step size limitation.

Above described Stefan problem is as well studied in Merle and Dolbow [2002] using X-FEM method. Merle and Dolbow [2002] considered domain of size $[0, 10]$ cm, partitioned into 20 uniformly spaced elements of length $h = 0.5$ and Δt was chosen as $h^2/\beta_1 = 12.76$. In X-HDG, 20 elements of size $h = 1$ is considered. To make sure that time errors are not dominating the solution; Δt is chosen as 0.1. Figure 6.3 shows the comparison of the percent error in front position through 200 seconds. With X-HDG, using a coarser mesh but smaller Δt; interface position is

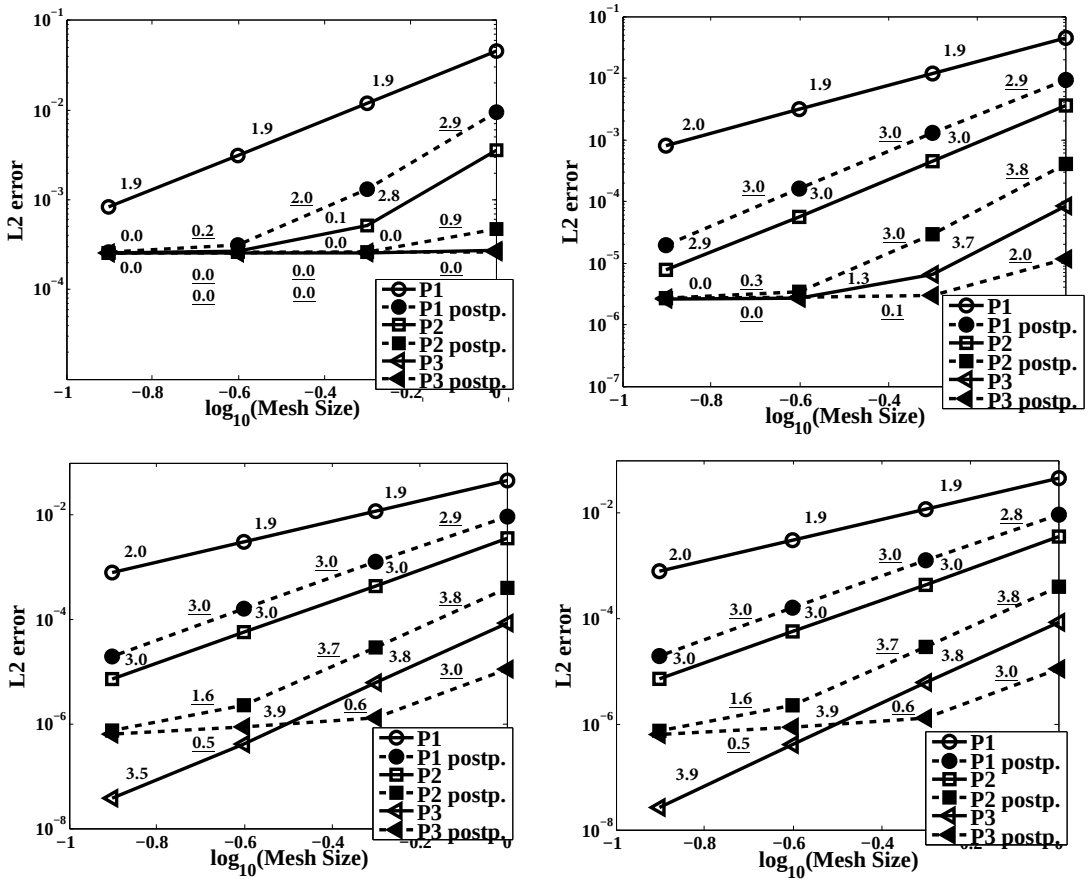


Figure 6.2: Two-phase Stefan problem: Convergence of X-HDG with $\Delta t = 1e{-}2$ (top,left) with $\Delta t = 1e{-}4$ (top,right) with $\Delta t = 1e{-}6$ (bottom,left) and with $\Delta t = 1e{-}14$ (bottom,right). Bold numbers show the slopes for solution T, underlined bold numbers show the slopes for postprocessed solution T^*.

approximated more accurately, as expected. The maximum percent error in front position with X-HDG is observed to be -0.1116%; however, with X-FEM, the same value goes up to -2.1040%. It is important to note that however, for X-HDG time step size is a limitation, leading to higher computational cost to achieve the more accurate results. In Figure 6.4, percent error in front position is shown using a mesh with elements of size $h = 0.5$ and keeping the $\Delta t = 0.1$. On the bottom, without using the mesh adaptation strategy explained in section 3.2.3 and on the top, with modified mesh in case of bad cuts. It can be observed very clearly that, non treated bad cuts cause accumulation of errors and very poor approximation for interface potion and finally explosion of errors. With this test, the effect of ill-conditioning is observed very clearly. Figure 6.4 reveals another interesting fact that is, using a mesh of element size $h = 0.5$, the maximum percent error is in interface solution is

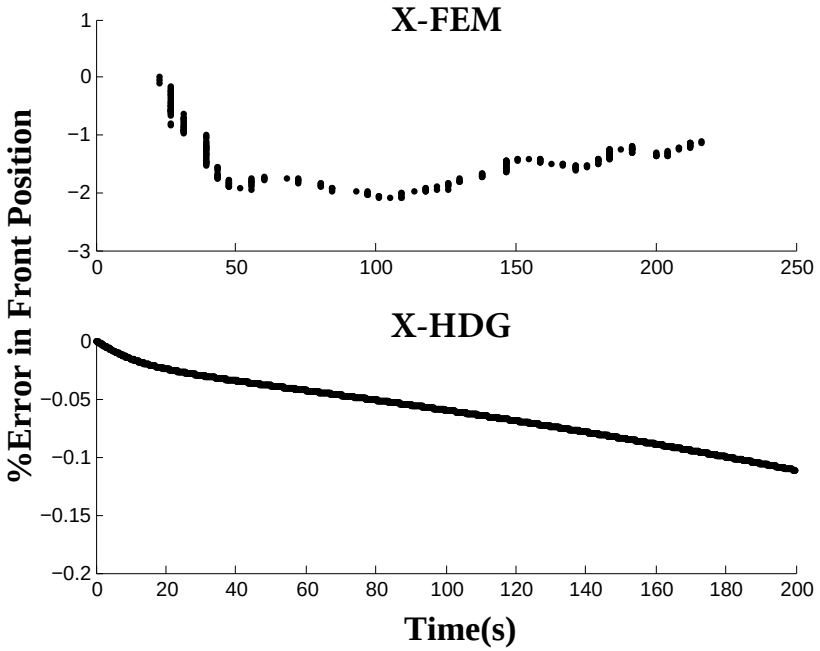


Figure 6.3: Two-phase Stefan problem: The percent error in front position. The results obtained with X-FEM, taken from Merle and Dolbow [2002] with elements of size $h = 0.5$ (top), the results obtained with X-HDG with elements of size $h = 1$ (bottom).

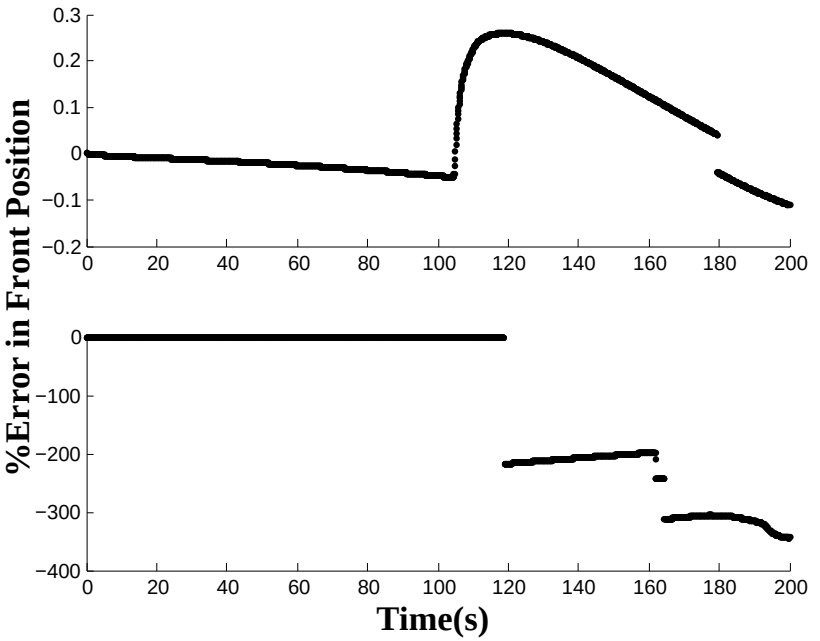


Figure 6.4: Two-phase Stefan problem: The percent error in front position. Using mesh modification strategy (top), without mesh modification (bottom).

0.2590% whereas it is -0.1116% with a mesh of element size $h = 1$. This is because, with the coarser mesh, interface advances within a single element through the entire simulation such that no element in the domain changes its state from being cut to uncut or vice versa. Meaning the approximation spaces at time t^{n-1} and time t^n are the same for the simulation. With a finer mesh on the other hand, interface crosses several interelement boundaries and solutions T^{n-1} are interpolated to the approximation space of time t^n even though the enriched approximation spaces at time t^{n-1} and time t^n are not the same. Those interpolation errors do not dominate the solution however their accumulation causes slightly bad approximated interface position. Improvement of this issue is considered as future work see in Chapter 7.

6.6 Conclusions and final remarks

Detailed X-HDG formulation for time dependent heat equation has been presented. A modified local problem is derived and considered for cut elements and the standard elements are solve following standard HDG setting. The local solver obtained with X-HDG has the same structure as HDG, easing the understanding of the method. The local solver then plugged into discretized global equations to end up with a global system that only has the trace variable $\widehat{u}$ as an unknown. At the end X-HDG is tested over two numerical examples, first a circle collapsing with unit speed is considered and convergence studies are done, demonstrating that the optimal and super convergence is achieved without mesh fitting or re-meshing. Second, influenced from Merle and Dolbow [2002], two phase Stefan problem is considered.

X-HDG is especially advantageous against current available interface methods dealing with moving interface problems. First, because with X-HDG, the computational mesh does not need to fit to the interface reducing the cost of re-meshing at each time step. Second, the problems where interface velocity is determined by the gradient jump, thanks to the fact that with X-HDG, gradient is approximated with order $k + 2$, interface position is better determined. The performance of X-HDG in such case is examined with numerical example presented in section 6.5.2.

The X-HDG method is fully developed for the solution of unsteady heat equation, describing the principles of the method. The core of this document is finalized with this chapter.

Chapter 7

Conclusions and Future Work

As it has been presented in introductory chapter, interface problems are of high interest. Many clever strategies have been studied and implemented to tackle them through the time, however; with the proposition of high order super convergent HDG method, some extra room is provided to improve existing methods. This thesis proposes a new technique; the eXtended Hybridizable Discontinuous Galerkin (X-HDG) method, to deal with interface problems by combining HDG strategy with X-FEM philosophy. In X-HDG, the interface geometry is defined using a level set function, saving high order accurate meshing cost for complex interface geometries and remeshing costs for moving interfaces. Moreover, X-HDG inherits favored properties of HDG, such as its optimum and super convergence, stability, element by element fashion, reduced system size compared to other DG methods or its suitability for parallel computing.

Hence, the foremost contributions of this thesis are:

1. **X-HDG for second order elliptic equations:** Previous proposals where second order elliptic interfaces are examined, for example in Hansbo and Hansbo [2002]; optimum convergence was proved over an unfitted mesh, however only using linear elements. In Sala Lardies et al. [2012], high order optimum convergence was presented without mesh fitting but, no super convergence was shown. In Bastian and Engwer [2009], a high order DG method, over unfitted meshes is studied and optimum convergence is demonstrated. Moreover, for problems with sufficient regularity theoretical super convergence is proved.

Here, in Chapters 2 and 3, X-HDG is presented for second order elliptic equations. First, problems including voids are studied. A modified weak form is derived for the elements and faces cut by the interface; moreover, a modified quadrature is implemented to handle the integrations properly. For the elements not cut by the interface i.e., standard elements, standard HDG formulation is followed. Numerical examples demonstrate that X-HDG keeps optimal and super convergence of HDG together with its local solver structure without need for mesh adaptation. Second, bimaterial problems are examined. Modified weak form is derived analogously for cut elements and faces but this time with Heaviside enrichment to take into account the discontinuities within the cut elements and faces. With numerical examples, again, optimum and super convergence of X-HDG became evident. Examples as well show that X-HDG handles the cases where multiple enrichment is needed more accurately than continuous methods.

Overall, X-HDG outperform the previous proposals by demonstrating high order optimum and super convergence, together with reduced system size thanks to its hybrid nature without mesh fitting.

2. **X-HDG for Stokes problem:** Among previous proposals where Stokes interfaces are studied; for example, Becker et al. [2009] presented their work where optimum convergence was shown using linear elements with level set definition of the interface. On the other hand in Wang and Khoo [2013], Stokes interface problem was treated using HDG over a fitted mesh where optimum and super convergence was proved.

   Here, X-HDG for Stokes interface problems is presented in Chapters 4 and 5. First void interfaces are studied. X-HDG formulation is derived where the local solver structure is kept similar to the one of standard HDG. Numerical examples revealed that X-HDG inherits desired HDG properties without mesh adaptation. Second bimaterial interfaces are examined. A Heaviside enrichment is introduced in the modified weak form, at cut element and faces, analogous to the case of second order elliptic equations. Numerical examples verified the applicability of X-HDG for Stokes bimaterial interfaces. It is important to note that in X-HDG, similar to standard HDG, the pressure term appearing in Stokes equation is treated just as a scalar for each element rather than an unknown to be solved for. Hence, it adds no extra difficulty to the calculations.

Here, with X-HDG, high order convergence is demonstrated over unfitted meshes; where pressure is not an unknown, leading to reduced system size, significantly improving previous proposals treating Stokes interfaces.

3. **X-HDG for Moving interfaces:** Previous proposals where moving interfaces are considered using immersed boundary method such as Li [1997] or Hou et al. [1997]; up to second order accuracy was obtained. On the other hand, in Merle and Dolbow [2002], X-FEM method was successfully implemented for the Stefan problem where interface position was predicted with less than 2% error at early times.

   In Chapter 6, X-HDG is presented for moving interface problems. It is demonstrated that X-HDG offers high order optimal convergence for time-dependent problems. Moreover, with Stefan problem, using a polynomial degree k, a more accurate approximation of interface position is demonstrated against X-FEM, thanks to $k+1$ convergent gradient approximation of X-HDG. Yet again, results obtained by previous proposals are improved.

The main contributions are being listed, next, future work considerations are outlined.

- **Improving X-HDG for moving interface problems:** As the Stefan problem studied in section 6.5.2 revealed, one of the obstacles in front of X-HDG to be comparable with X-FEM is the time discretization method currently implemented. Backward Euler is a stable, first order accurate technique, however; to ensure that the time errors are not dominant in the problem, small time step size need to be considered in our implementation. Interface position is better (compared to X-FEM) approximated with X-HDG even with a coarser mesh (see Figure 6.3), but with smaller time step size.

  To get the most of X-HDG and to make it more competitive, an advanced time integration scheme is needed such that, using similar time step size as X-FEM; interface position is better approximated. The pros and cons of explicit and implicit time discretization methods, their implementation and further comparisons of X-HDG with X-FEM are considered as future work.

  Moreover, as it is demonstrated on section 6.5.2, solving the Stefan problem with finer mesh leads to a poorer interface position approximation. This is

because of the accumulation of interpolation errors, caused by the interpolation of the solution u^{n-1} to the approximation space of time t^n while the enriched approximation spaces at time t^{n-1} and time t^n are not the same. To improve our results for moving interface problems, implementation of an $\mathcal{L}_2$ projection for the solution u^{n-1} to the approximation space of time t^n is also considered as a part of the future work.

- **Stabilized X-HDG:** Throughout this document, ill-conditioning problem arising from bad-cut situations is mentioned often. This is a common problem for methods dealing with interface problems where interface cuts the elements very close to its edges or corners. In this case, the area ratio between two domains (or the domain with void) separated by the interface is very small, causing ill conditioned system matrix and accordingly convergence loss.

  In the context of continuous methods, clever stabilization strategies are proposed, see for example Burman and Hansbo [2012]. However, for high order unfitted methods, including X-HDG, a stabilized formula to avoid ill conditioning problem is open area of research. In the context of this document, an easy solution is proposed, avoiding the problem in section 3.2.3. Further in Appendix, some ideas for a more general stabilization strategy are listed.

  Combining the stabilization ideas presented in Appendix section 8.2 –leading stabilized local matrix– together with the ones presented in section 8.4 –leading to stabilized global matrix– and analyzing results is surely an interesting topic; might lead to an interface position free stable X-HDG formula.

- **X-HDG for Navier-Stokes Equations:** To give more impact to X-HDG method and applicability to real life problems, treating Navier-Stokes equations with X-HDG is a must. This is, indeed, a challenging task but it most definitely be considered as one of the next research topics.

- **Optimizing X-HDG code:** The MATLAB code created and used for all the work presented in this thesis is very basically optimized; hence, there is room to further optimize and parellize it to get the most of the discontinuous approximation. This is especially necessary to be able to run fair comparison analysis in the future.

- **Comparison of X-HDG with X-FEM:** Often in this document the convergence properties of X-HDG and its advantages are mentioned. It is evident that for polynomial degree k, X-HDG inherits the order $k+1$ convergence for primal unknown, but more importantly for its derivative. $k+1$ convergent gradients is not usual for other methods, or more specifically to X-FEM. Here in Chapter 6 with Stefan example, the advantage of this strong property is intended to be shown. For future reference, a fair, further comparison between X-HDG and X-FEM might be of high interest, to reveal if X-HDG can offer better approximations than X-FEM, at least for certain problem types, or not.

Bibliography

C. Annavarapu, M. Hautefeuille, and J. Dolbow. A robust Nitsche's formulation for interface problems. *Computer Methods in Applied Mechanics and Engineering*, 225-228:44–54, 2012. doi: 10.1016/j.cma.2012.03.008.

D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5): 1749–1779, 2001. ISSN 0036-1429.

D. C. Assêncio and J. M. Teran. A second order virtual node algorithm for Stokes flow problems with interfacial forces, discontinuous material properties and irregular domains. *J. Comput. Phys.*, 250:77–105, 2013. ISSN 0021-9991.

I. Babuška. Finite element method for elliptic equations with discontinuous coefficients. *Computing*, 5(3):207–213, 1970. doi: 10.1007/bf02248021.

J. W. Barrett and C. M. Elliott. Fitted and Unfitted Finite-Element Methods for elliptic equations with smooth interfaces. *Journal of Numerical Analysis*, 7:283–300, 1987.

P. Bastian and C. Engwer. An unfitted finite element method using discontinuous Galerkin. *International Journal for Numerical Methods in Engineering*, 79:1557–1576, 2009. doi: 10.1002/nme.2631.

R. Becker, E. Burman, and P. Hansbo. A Nitsche extended finite element method for incompressible elasticity with discontinuous modulus of elasticity. *Computer Methods in Applied Mechanics and Engineering*, 198(41-44):3352–3360, 2009. ISSN 00457825. doi: 10.1016/j.cma.2009.06.017.

E. Burman and P. Hansbo. Fictitious domain finite element methods using cut elements: II. a stabilized Nitsche method. *Applied Numerical Mathematics*, 62(4): 328–341, 2012. doi: 10.1016/j.apnum.2011.01.008.

K. W. Cheng and T. P. Fries. Higher-order XFEM for curved strong and weak discontinuities. *Int. J. Numer. Methods Eng.*, 82(5):564–590, 2010. ISSN 0029-5981.

P. G. Ciarlet and P. A. Raviart. Interpolation theory over curved elements, with applications to finite element methods. *Comput. Methods Appl. Mech. Eng.*, 1(1): 217–249, 1972.

R. W. Clough. The finite element method in plane stress analysis. *Proceedings of the 2nd Conference on Electronic Computation of American Society of Civil Engineers*, pages 345–378, 1960.

B. Cockburn and M. Solano. Solving convection-diffusion problems on curved domains by extensions from subdomains. *J. Sci. Comput.*, 59(2):512–543, 2014.

B. Cockburn, B. Dong, and J. Guzmán. A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. *Math. Comp.*, 77(264):1887–1916, 2008. ISSN 0025-5718.

B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J. Numer. Anal.*, 47(2):1319–1365, 2009.

B. Cockburn, J. Gopalakrishnan, N. C. Nguyen, J. Peraire, and F.-J. Sayas. Analysis of HDG methods for Stokes flow. *Math. Comp.*, 80(274):723–760, 2011.

B. Cockburn, W. Qiu, and K. Shi. Conditions for superconvergence of HDG methods for second-order elliptic problems. *Math. Comp.*, 81(279):1327–1353, 2012.

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.

H. Dong, B. Wang, Z. Xie, and L.-L. Wang. An unfitted hybridizable discontinuous Galerkin method for the Poisson interface problem and its error analysis. *IMA J Numer Anal*, 37(1):444–476, 2016.

K. Dréau, N. Chevaugeon, and N. Moës. Studied X-FEM enrichment to handle material interfaces with higher order finite element. *Comput. Methods Appl. Mech. Eng.*, 199(29–32):1922–1936, 2010. ISSN 0045-7825.

I. Ergatoudis, B. Irons, and O. Zienkiewicz. Curved, isoparametric, "quadrilateral" elements for finite element analysis. *International Journal of Solids and Structures*, 4(1):31–42, 1968. doi: 10.1016/0020-7683(68)90031-0.

K. Fidkowski and D. Darmofal. A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 225:1653–1672, 2007. doi: 10.1016/j.jcp.2007.02.007.

T. P. Fries and T. Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304, 2010. doi: 10.1002/nme.2904.

G. Giorgiani, S. Fernández-Méndez, and A. Huerta. Hybridizable Discontinuous Galerkin p-adaptivity for wave propagation problems. *Int. J. Numer. Methods Fluids*, 72(12):1244–1262, 2013a.

G. Giorgiani, D. Modesto, S. Fernández-Méndez, and A. Huerta. High-order continuous and discontinuous Galerkin methods for wave problems. *Int. J. Numer. Methods Fluids*, 73(10):883–903, 2013b.

S. Gross and A. Reusken. An extended pressure finite element space for two-phase incompressible flows with surface tension. *J. Comput. Phys.*, 224(1):40–58, 2007. ISSN 0021-9991.

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. ISSN 1573-7691.

C. Gürkan, M. Kronbichler, and S. Fernández-Méndez. eXtended hybridizable discontinuous Galerkin with heaviside enrichment for heat bimaterial problems. *J. Sci. Comput.*, 2017. doi: 10.1007/s10915-017-0370-6.

G. Guyomarch, C. Lee, and K. Jeon. A discontinuous Galerkin method for elliptic interface problems with application to electroporation. *Communications in Numerical Methods in Engineering*, 25:991–1008, 2009.

A. Hansbo and P. Hansbo. An unfitted finite element method, based on Nitsche method for elliptic interface problems. *Comput. Methods Appl. Mech. Eng.*, 191: 5537–5552, 2002.

P. Hansbo, M. Larson, and S. Zahedi. A cut finite element method for a Stokes interface problem. *Applied Numerical Mathematics*, 85:90–114, 2014. doi: 10.1016/j.apnum.2014.06.009.

T. Y. Hou, Z. Li, S. Osher, and H. Zhao. A hybrid method for moving interface problems with application to the Hele-Shaw flow. *Journal of Computational Physics*, 134(2):236–252, 1997. doi: 10.1006/jcph.1997.5689.

A. Huerta, A. Angeloski, X. Roca, and J. Peraire. Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *Int. J. Numer. Methods Eng.*, 96(9):529–560, 2013.

L. T. Huynh, N. Nguyen, J. Peraire, and B. Khoo. A high-order hybridizable discontinuous Galerkin method for elliptic interface problems. *Int. J. Numer. Meth. Eng.*, 93(2):183–200, 2013. ISSN 1097-0207.

H. Ji, D. Chopp, and J. E. Dolbow. A hybrid extended finite element/level set method for modelling phase transformations. *International Journal for Numerical Methods in Engineering*, 54:1209–1233, 2002. doi: 10.1002/nme.468.

R. Kirby, S. J. Sherwin, and B. Cockburn. To CG or to HDG: A comparative study. *J. Sci. Comput.*, 51(1):183–212, 2011. ISSN 0885-7474.

G. Legrain, N. Chevaugeon, and K. Dréau. High order X-FEM and level sets for complex microstructures: Uncoupling geometry and approximation. *Comput. Methods Appl. Mech. Eng.*, 241-244:172–189, 2012. ISSN 0045-7825.

R. J. Leveque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis SIAM J. Numer. Anal.*, 31(4):1019–1044, 1994. doi: 10.1137/0731054.

R. J. Leveque and Z. Li. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput. SIAM Journal on Scientific Computing*, 18(3):709–735, 1997. doi: 10.1137/s1064827595282532.

Z. Li. Immersed interface methods for moving interface problems. *Numerical Algorithms*, 14(4):269–293, 1997.

R. Massjung. An Unfitted Discontinuous Galerkin Method Applied to Elliptic Interface Problems. *SIAM Journal on Numerical Analysis*, 50:3134–3162, 2012. doi: 10.1137/090763093.

J. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1):289 – 314, 1996. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/S0045-7825(96)01087-0.

R. Merle and J. Dolbow. Solving thermal and phase change problems with the eXtended finite element method. *Comput. Mech.*, 28(5):339–350, 2002. ISSN 0178-7675.

N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering Int. J. Numer. Meth. Engng.*, 46(1):131–150, Oct 1999.

A. Montlaur, S. Fernández-Méndez, and A. Huerta. Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. *Int. J. Numer. Methods Fluids*, 57(9):1071–1092, 2008.

H. M. Mourad, J. E. Dolbow, and K. Garikipati. An assumed-gradient finite element method for the level set equation. *International Journal for Numerical Methods in Engineering*, 00:1–6, 2005.

N. Nguyen, J. Peraire, and B. Cockburn. A hybridizable discontinuous Galerkin method for Stokes flow. *Comput. Methods Appl. Mech. Eng.*, 199(9-12):582–597, 2010. ISSN 0045-7825.

N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 230(4):1147–1170, 2011.

S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988. doi: 10.1016/0021-9991(88)90002-2.

J. Peraire and P. O. Persson. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM J. Sci. Comput.*, 30(4):1806–1824, 2008. ISSN 1064-8275.

C. Peskin. Flow patterns around heart valves : a digital computer method for solving the equations of motion. *Dissertation*, 1972.

C. S. Peskin and D. M. Mcqueen. A three-dimensional computational method for blood flow in the heart I. immersed elastic fibers in a viscous incompressible fluid. *Journal of Computational Physics*, 81(2):372–405, 1989. doi: 10.1016/0021-9991(89)90213-1.

W. H. Reed and T. R. Hill. Triangular Mesh Methods for the Neutron Transport Equation. *Los Alamos Scientific Laboratory, Los Alamos Report LA-UR-73-479*, 1973.

E. Sala Lardies, S. Fernández-Méndez, and A. Huerta. Optimally convergent high-order X-FEM for problems with voids and inclusions. *Proceedings of the European Congress on Computational Mechanics in Applied Sciences and Engineering (ECCOMAS 2012)*, pages 8080–8093, 2012.

R. Sevilla, S. Fernández-Méndez, and A. Huerta. Comparison of high-order curved finite elements. *Int. J. Numer. Methods Eng.*, 87(8):719–734, August 2011. ISSN 0029-5981.

N. Sukumar, D. L. Chopp, N. Moës, and T. Belytschko. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering*, 190(46-47):6183–6200, 2001. doi: 10.1016/S0045-7825(01)00215-8.

M. Turner, R. Clough, H. Martin, and L. Topp. Stiffness and deflection analysis of complex structures. *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)*, 23:805–823, 1956. doi: 10.2514/8.3664.

B. Wang and B. Khoo. Hybridizable discontinuous Galerkin method (HDG) for Stokes interface flow. *J. Comput. Phys.*, 247:262–278, 2013.

Q. Wang and J. Chen. An Unfitted Discontinuous Galerkin Method for Elliptic Interface Problems. *Journal of Applied Mathematics*, 2014. doi: 10.1155/2014/241890.

O. Zienkiewicz and Y. K. Cheung. Finite elements in the solution of field problems. *The Engineer*, 1965.

O. C. Zienkiewicz, R. L. Taylor, and J. Zhu. *The Finite Element Method:Its Basis and Fundamentals*. Butterworth-Heinemann, 2005.

Chapter 8

Appendix

Further thoughts on stabilized X-HDG

A "bad cut", in the context of this document, for a bimaterial problem implies that the area ratio of two material domains –or single material domain with void domain– separated by the interface is less than 0.1.

As it is presented in section 3.2, numerical methods dealing with interface problems suffers the "bad-cut" situations in general, see for example Hansbo and Hansbo [2002], causing an ill conditioned system matrix. With X-HDG as well, bad-cut situations and their negative effect on convergence rates are observed closely. Although finally the mesh adjustment strategy presented in section 3.2 is applied to cure numerical examples, a more general stabilization strategy for X-HDG is also investigated. Here, different trials for a stabilized X-HDG formulation and their effect on condition number of local matrices are presented. It is important to note that for a bad cut element the condition number of the local matrix can grow as high as $1e15$ whereas this number is in the order of $1e3$ for a standard element.

As it became evident that stabilized X-HDG requires additional developments and diverts from the framework of this thesis, it has been studied only preliminarily and summarized in this appendix.

8.1 Applying Hansbo's Idea to X-HDG

Burman and Hansbo [2012] proposed a stabilized Nitsche formulation for cut finite element method where the optimal error estimates are proved in $\mathcal{H}^1$ and $\mathcal{L}_2$ norms for linear elements and the condition number of the system matrix is bounded.

An interface position free, high order stabilized formula sounds very appealing. As a first attempt to stabilize X-HDG, ideas presented in Burman and Hansbo [2012] are tried to be applied for X-HDG. After a priliminary study, it has been evident that the ghost penalty term used by Burman and Hansbo is very suitable for a continuous setting; however, it does not fit to our discontinuous setting. The reason is that, the stabilization term added to the weak form glues the bad cut element with its neighbor by smoothly extending the polynomial approximation. In X-HDG setting, this idea would not allow the treatment of elements one by one or the local problem would not be satisfied after stabilization.

An initial investigation revealed that while this approach is indeed very suitable for high-order continuous approximations, it leads to the loss of favored properties of discontinuous X-HDG setting, no further implementations are done.

8.2 Adding stabilization terms to X-HDG local problem

As a second attempt, analytically zero stabilization terms, suitable for X-HDG formulation, have been added to the weak form of the local problem. As an example, let us consider the second order elliptic equation over the domain with size $\Omega = (-1, 1)^2$ and a circular Neumann void of radius $R = 0.5$ at the center. With fourth order, triangular elements of size $h = 0.125$; the condition number of the local matrix is of order $1e15$ while the global system matrix is conditioned in order $1e12$. The local problem for Neumann cut elements is recalled from section 2.1, now together with the new stabilization terms takes the form,

$$\begin{aligned} &\int_{\Omega_i} v\boldsymbol{\nabla}\cdot\boldsymbol{q}\,dV + \int_{\partial\Omega_i\setminus\mathcal{I}_i} \nu\tau v(u-\widehat{u})\,dS + \int_{\mathcal{I}_i} \nu\tau v(u-\widetilde{u})\,dS + ST_1 = \int_{\Omega_i} vf\,dV \\ &\int_{\Omega_i} \boldsymbol{q}\cdot\boldsymbol{w}\,dV - \int_{\Omega_i} \nu u\boldsymbol{\nabla}\cdot\boldsymbol{w}\,dV + \int_{\partial\Omega_i\setminus\mathcal{I}_i} \nu\widehat{u}\boldsymbol{w}\cdot\boldsymbol{n}\,dS + \int_{\mathcal{I}_i} \nu\widetilde{u}\boldsymbol{w}\cdot\boldsymbol{n}\,dS + ST_2 = 0 \end{aligned} \tag{8.1}$$

above equations in discrete form, after the elimination of $\widetilde{u}$ leads to,

$$\begin{cases} \left[\mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right]\mathbf{u}^i & + \left[\mathbf{A}_{uq}^{\Omega_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_q^i\right]\mathbf{q}^i + \mathbf{A}_{u\widehat{u}}^{\Omega_i}\mathbf{\Lambda}^i = \mathbf{f}_u^{\Omega_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{t}^i \\ \left[\mathbf{A}_{qu}^{\Omega_i} + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right]\mathbf{u}^i & + \left[\mathbf{A}_{qq}^{\Omega_i} + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_q^i\right]\mathbf{q}^i + \mathbf{A}_{q\widehat{u}}^{\Omega_i}\mathbf{\Lambda}^i = \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{t}^i \end{cases} \tag{8.2}$$

and the local matrix is,

$$\mathbb{A} = \begin{bmatrix} \left[\mathbf{A}_{uu}^{\Omega_i} + \mathbf{A}_{uu}^{\mathcal{I}_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{uq}^{\Omega_i} + \mathbf{A}_{u\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_q^i\right] \\ \left[\mathbf{A}_{qu}^{\Omega_i} + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_u^i\right] & \left[\mathbf{A}_{qq}^{\Omega_i} + \mathbf{A}_{q\widetilde{u}}^{\mathcal{I}_i}\mathbf{T}_q^i\right] \end{bmatrix}$$

For the sake of simplicity, lets rename the entries of the local matrix,

$$\mathbb{A} = \begin{bmatrix} \mathbf{M}_{uu}^{\Omega_i} & \mathbf{M}_{uq}^{\Omega_i} \\ \mathbf{M}_{qu}^{\Omega_i} & \mathbf{M}_{qq}^{\Omega_i} \end{bmatrix}$$

To stabilize the above local matrix the idea here is to add analytically zero stabilization terms –ST_1 and ST_2– to the local problem so that after stabilization, the local problem is still satisfied. Different trials for ST_1 and ST_2 terms and their effect on local matrix conditioning is explained next. The parameters β_1, β_2 and β_3 are constants to be tuned by the user. In the integral equations below, OAF and OAE are used as ackronyms for $OverAllFace$ and $OverAllElement$.

1. Stabilizing only the first equation in (8.1),
   $ST_1 = \beta_1 \int_{OAF} \tau v(u - \widehat{u})\, dS \quad$ and $\quad ST_2 = 0$

   using this setting, only the entry $\mathbf{M}_{uu}^{\Omega_i}$ was modified at the local matrix. Choosing $\beta_1 = 100$, local matrix got more conditioned to order $1e18$ while the conditioning of the global system stayed the same, in order $1e12$. Changing β_1 to unity did not improve the results either; the condition number of local and global matrices did not change. Further changing $\beta_1 = 10$; the condition number of the local system was calculated to be $1e16$ while the global system was not affected. Overall with above choice, the symmetry of the local matrix was lost and condition numbers did not improve neither in local nor in global level.

2. Stabilizing both equations in (8.1),
   $ST_1 = \beta_1 \int_{OAF} \tau v(u - \widehat{u})\, dS \quad$ and $\quad ST_2 = \beta_2 \int_{OAE} (\boldsymbol{q} + \nu \boldsymbol{\nabla} u) \cdot \boldsymbol{w}\, dV$

with this scenario, at discrete level, the matrices $\mathbf{M}_{uu}^{\Omega_i}$, $\mathbf{M}_{qu}^{\Omega_i}$ and $\mathbf{M}_{qq}^{\Omega_i}$ were modified, leading again to a non symmetric local matrix. Choosing both $\beta_1 = 1$ and $\beta_2 = 1$, the condition number of the local system was improved to order $1e11$ but still not healed. The global system on the other hand, worsened, condition number rising up to $1e15$. For different values of β_1 and β_2 the results did not enhanced, the best scenario was obtained while both of them are set to unity. However, even in the best case, the second choice did not heal the problem.

3. Stabilizing all entries of local matrix,
$ST_1 = \beta_1 \int_{OAF} \tau v(u - \widehat{u})\, dS + \beta_3 \int_{OAE} v(\boldsymbol{\nabla} \cdot \boldsymbol{q} - f)$ and
$ST_2 = \beta_2 \int_{OAE} (\boldsymbol{q} + \nu \boldsymbol{\nabla} u) \cdot \boldsymbol{w}\, dV$

with the above choice, all the entries of the local matrix, $\mathbf{M}_{uu}^{\Omega_i}$, $\mathbf{M}_{qu}^{\Omega_i}$, $\mathbf{M}_{qq}^{\Omega_i}$ and $\mathbf{M}_{uq}^{\Omega_i}$ were modified. However, because of the nonsymmetric stabilization terms added to symmetric $\mathbf{M}_{qu}^{\Omega_i}$ and $\mathbf{M}_{uq}^{\Omega_i}$ matrices, the overall symmetry of the local matrix is again lost after the stabilization. On the other hand, setting β_1, β_2 and β_3 to unity; the local system matrix was healed, with a condition number of order $1e6$. The global system matrix condition number, however; worsened to $1e15$. The best choice for βs was observed to be unity and choosing all βs in the same order lead to better results.

To sum up, third strategy mentioned above successfully healed the local system but some further stabilization was needed to heal the global matrix, to achieve an overall stabilized X-HDG. Further thoughts on this topic has been explained in section 8.4.

8.3 Amplifying the approximation basis

As an easy and practical solution, amplifying the basis functions used for bad cut elements and faces was tried. For volume integrals, the basis functions are amplified by a factor of $\frac{Area_{total}}{Area_{badcut}}$ and for face integrals by a factor of $\frac{Length_{total}}{Length_{badcut}}$. With this strategy, even though the condition number of the local matrix was improved a little, but its order did not change. Hence the idea was not simply enough to cure the problem.

8.4 Overall Stabilized X-HDG

Very preliminary, an overall stabilized X-HDG can be achieved using the final stabilization choice presented in section 8.2 together with a stabilized global matrix. Looking into the conservativity condition $[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!]=0$ in weak form,

$$\int_\Gamma \widehat{v}[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!]\,dS + 2\int_\Gamma \widehat{v}\left(\{\tau\nu u\}-\{\tau\nu\}\widehat{u}\right)\,dS = 0 \tag{8.3}$$

it is simple to realize that in case of a bad cut face, the integral expression above is calculated over a very small length and needs to be stabilized as it has been done to integrals at local equations. One simple idea is amplifying the stabilization parameter τ with a constant $\gamma=\frac{Length_{AllFace}}{Length_{BadCut}}$. As a second option, analytically zero stabilization terms of type $ST_3=\beta_4\int_{OAF}\widehat{v}[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!]\,dS$, $ST_4=2\beta_5\int_{OAF}\widehat{v}\left(\{\tau\nu u\}-\{\tau\nu\}\widehat{u}\right)\,dS$ can be added to (8.3) to achieve a better conditioned global matrix. AOF here is an ackronym for $OverAllFace$.

A fully stabilized X-HDG, further investigation of the ideas have been presented above and their implementation to different problem types is an open area of research.