Dual-Primal Isogeometric Tearing and Interconnecting solvers for multipatch dG-IgA equations

C. Hofer, U. Langer

G+S Report No. 43

December 2015

Dual-Primal Isogeometric Tearing and Interconnecting solvers for multipatch dG-IgA equations

Christoph Hofer[1] and Ulrich Langer[1]

[1] Johann Radon Institute for Computational and Applied Mathematics (RICAM),
Austrian Academy of Sciences
christoph.hofer@ricam.oeaw.ac.at
ulrich.langer@ricam.oeaw.ac.at

Abstract. In this paper we consider a variant of the dual-primal isogeometric tearing and interconnecting (IETI-DP) method for solving large-sacle linear systems of algebraic equations arising from discontinuous Galerkin (dG) isogeometric analysis of diffusion problems on multipatch domains with non-matching meshes. The dG formulation is used to couple the local problems across patch interfaces. The purpose of this paper is to present this new method and provide numerical examples indicating a polylogarithmic condition number bound for the preconditioned system and showing an incredible robustness with respect to large jumps in the diffusion coefficient across the interfaces.

Key words: Elliptic boundary value problems, diffusion problems, heterogeneous diffusion coefficients, Isogeometric Analysis, domain decomposition, FETI, IETI-DP algorithms, discontinuous Galerkin.

1 Introduction

Isogeometric Analysis (IgA) is an approach to approximate numerically the solution of a partial differential equation (PDE) using the same basis functions for parametrizing the geometry and for representing the solution. IgA was introduced by Huges, Cottrell and Bazilevs in [24], see also [3] for the first results on the numerical analysis of IgA, the monograph [10] for a comprehensive presentation of the IgA, and the recent survey article [4] on the mathematical analysis variational isogeometric methods. There exists a wide variety of different basis functions, e.g., B-Splines, Non Uniform Rational B-Spline (NURBS), T-Splines, Hierarchical B-Splines (HB-Splines) and Truncated Hierarchical B-Splines (THB-Splines), see, e.g., [10], [4], [18] and [19]. A major advantage over the common finite element method (FEM) is more flexibility of h- and p-refinement, resulting also in C^k, $k \geq 0$ continuous basis functions. However, due to the larger support of the basis functions, the resulting system matrices are denser and due to the more involved evaluation mechanism of B-Splines, NURBS, etc., we have to deal with much higher assembling times, see, e.g., Section 8 in [4] for a discussion of this issue and for relevant references. The low-rank tensor approximation technique, proposed in [32], is certainly a very smart and at the same time simple technology to overcome this bottleneck, see also [31] for a further development of this technique.

Beside matrix generation, the efficient solution of the linear systems arising from IgA discretization of linear elliptic boundary value problem or from the linearization of non-linear IgA equations turns out to be another bottleneck for the efficiency of the IgA technology. In this paper, we consider a non-overlapping domain decomposition method based on the finite element tearing and interconnecting (FETI) for IgA, called isogeometric tearing and interconnecting (IETI). In particular, we focus on the dual primal variant (IETI-DP), introduced by [26] in 2012. A comprehensive theoretical analysis of the FETI-DP and the equivalent Balancing Domain Decomposition by Constraints (BDDC) method can be found in the monographs [36] and [34] where the reader also find the references to the corresponding original papers. The theoretical

analysis of the IETI-DP and BDDC method was initiated in [6] and extended in [21]. Based on the FE work in [12], a recent improvement for the IgA BDDC preconditioner with a more advanced scaling technique, the so called *deluxe scaling*, can be found in [8]. Domain decomposition methods for IgA is currently a very active field of research. We mention developments in overlapping Schwarz methods, see, e.g. [5], [7], [9], and isogeometric mortaring discretizations, see [20].

In this paper, based on the IETI-DP mentioned above method, we are going to develop an efficient and robust solver for IgA systems arising from a discretization where the different patches are coupled with a discontinuous Galerkin (dG) method, called dG-IETI-DP. This setting is of special importance when considering non-matching meshes, see, e.g., [29], and in case of non-matching interface parametrizations, resulting in gaps and overlaps, see, [22] and [23]. The proposed method is based on the corresponding version for finite elements (FE) proposed in [15] and [16], where a rigorous analysis for 2D and 3D proves the same properties as for the classical FETI-DP, see also [14] for an analysis of the corresponding BDDC preconditioner.

In the present paper, we consider the following second-order elliptic boundary value problem in a bounded Lipschitz domain $\Omega \subset \mathbb{R}^d, d \in \{2, 3\}$, as a typical model problem: Find $u : \overline{\Omega} \to \mathbb{R}$ such that

$$-\operatorname{div}(\alpha \nabla u) = f \text{ in } \Omega, \ u = 0 \text{ on } \Gamma_D, \text{ and } \alpha\frac{\partial u}{\partial n} = g_N \text{ on } \Gamma_N, \tag{1}$$

with given, sufficient smooth data f, g_N and α, where the coefficient α is uniformly bounded from below and above by some positive constants α_{min} and α_{max}, respectively. The boundary $\Gamma = \partial\Omega$ of the computational domain Ω consists of a Dirichlet part Γ_D and a Neumann part Γ_N. Furthermore, we assume that the Dirichlet boundary Γ_D is always a union of complete domain sides (edges / face in 2d / 3d) which are uniquely defined in IgA. Without loss of generality, we assume homogeneous Dirichlet conditions. This can always be obtained by homogenization. By means of integration by parts, we arrive at the weak formulation of (1) which reads as follows: Find $u \in V_D = \{u \in H^1 : \gamma_0 u = 0 \text{ on } \Gamma_D\}$ such that

$$a(u, v) = \langle F, v \rangle \quad \forall v \in V_D, \tag{2}$$

where γ_0 denotes the trace operator. The bilinear form $a(\cdot, \cdot) : V_D \times V_D \to \mathbb{R}$ and the linear form $\langle F, \cdot \rangle : V_D \to \mathbb{R}$ are given by the expressions

$$a(u, v) = \int_\Omega \alpha\nabla u\nabla v\, dx \quad \text{and} \quad \langle F, v \rangle = \int_\Omega fv\, dx + \int_{\Gamma_N} g_N v\, ds.$$

The remainder of this paper is organized as follows. Section 2 gives a short overview of the main principles of IgA, and presents the dG-IgA formulation. The dG-IETI-DP method is defined and discussed in Section 3. The numerical results, presented in Section 4, demonstrate the numerical behaviour of dG-IETI-DP method. In particular, we study the influence of the mesh size h, the patch diameter H, the use of non-matching meshes quantified by the mesh size ratio $h^{(k)}/h^{(l)}$ across the patch faces, and the polynomial degree p on the condition number of the preconditioned system and, thus, on the number of iterations. Finally, in Section 5, we draw some conclusions and give some outlook.

2   Preliminaries

In this section, we give an overview of the tools required to describe the IETI-DP method for multipatch dG-IgA equations. A more comprehensive study of IgA, IETI-DP and related topics can be found in [21].

2.1   B-Splines and IgA

Let $[0, 1]$ be the unit interval, the vector $\Xi = \{\xi_1 = 0, \xi_2, \ldots, \xi_m = 1\}$ with non-decreasing real values ξ_i forms a partition of $[0, 1]$ and is called *knot vector*. Given a knot vector Ξ, $p \in \mathbb{N}$ and $n = m - p - 1$, we can define the B-Spline function via the following recursive formulation:

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi \leq \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi), \tag{4}$$

where $i = 1, \ldots, n$ and p is called *degree*. From this recursion, we can observe that $N_{i,p}$ is a piecewise polynomial of degree p. Furthermore, we only consider open knot vectors, i.e., the first and the last node is repeated p times.

Since we are considering d-dimensional problems, we need to extent the concept of B-Splines to the d-dimensional space, which is done via the tensor product. Let $(p_1, \ldots, p_d)$ be a vector in $\mathbb{N}^d$, and let, for all $\iota = 1, \ldots, d$, Ξ^ι be a knot vector. Furthermore, we denote the i_ι univariate B-Spline defined on the knot vector Ξ^ι by $N_{i_\iota,p}^\iota(\xi^\iota)$. Then the d-dimensional tensor product B-Spline (TB-Spline) is defined by

$$N_{(i_1,\ldots,i_d),(p_1,\ldots,p_d)}(\xi) = \prod_{\iota=1}^{d} N_{i_j,p_\iota}^\iota(\xi^\iota). \tag{5}$$

In order to avoid cumbersome notations, we will again denote the tensor product B-Spline by $N_{i,p}$ and interpret i and p as multi-indices. Additionally, we define the set of multi-indices $\mathcal{I}$ by

$$\mathcal{I} := \{(i_1, \ldots, i_d) : i_\iota \in \{1, \ldots, M_\iota\}\}.$$

Since the knot vector Ξ provides a partition of $(0, 1)^d$, called *parameter domain* in the following , it introduces a mesh $\hat{\mathcal{Q}}_h$, and we will denote a mesh element by $\hat{Q}$, called *cell*.

Now we are in a position to describe our computational domain, called *physical domain*, $\Omega = G((0, 1)^d)$ by means of the *geometrical mapping* G defined by

$$G : (0, 1)^d \to \mathbb{R}^g$$
$$G(\xi) := \sum_{i \in \mathcal{I}} P_i N_{i,p}(\xi).$$

In practice, it is often necessary to describe the physical domain Ω by N non overlapping domains $\Omega^{(k)}$, called *patches*. Each $\Omega^{(k)}$ is the image of an associated geometrical mapping $G^{(k)}$, defined on the parameter domain $(0, 1)^d$, i.e., $\Omega^{(k)} = G^{(k)}\left((0, 1)^d\right)$ for $k = 1, \ldots, N$, and $\overline{\Omega} = \bigcup_{k=1}^{N} \overline{\Omega}^{(k)}$.

We denote the interface between the two patches $\Omega^{(k)}$ and $\Omega^{(l)}$ by $\Gamma^{(k,l)}$, and the collection of all interfaces by Γ, i.e.,

$$\Gamma^{(k,l)} = \overline{\Omega}^{(k)} \cap \overline{\Omega}^{(l)} \quad \text{and} \quad \Gamma := \bigcup_{l>k} \Gamma^{(k,l)}.$$

Furthermore, the boundary of the domain Ω is denoted by $\partial\Omega$. Note that the interface Γ is sometimes called *skeleton*.

The key point of IgA is to use the same basis functions for representing the geometry via the geometrical mapping also for generating the trial and test spaces. Therefore, we define the basis functions in the physical domain as $\check{N}_{i,p} := N_{i,p} \circ G^{-1}$ and the discrete function space by

$$V_h = \text{span}\{\check{N}_{i,p}\}_{i\in\mathcal{I}} \subset H^1(\Omega). \tag{6}$$

Moreover, each function $u(x) = \sum_{i\in\mathcal{I}} u_i \check{N}_{i,p}(x)$ is associated with the vector $\boldsymbol{u} = (u_i)_{i\in\mathcal{I}}$. This map is known as *Ritz isomorphism*. One usually writes this relation as $u_h \leftrightarrow \boldsymbol{u}$, and we will use it in the following without further comments. If we consider a single patch $\Omega^{(k)}$ of a multipatch domain Ω, we will use the notation $V_h^{(k)}$, $\check{N}_{i,p}^{(k)}$, $N_{i,p}^{(k)}$ and $G^{(k)}$ with the analogous definitions.

2.2 Conforming Galerkin IgA Scheme

In conforming Galerkin IgA schemes, we use functions which are continuous across patch interfaces, i.e.

$$V_h = \{v \mid v|_{\Omega^{(k)}} \in V_h^{(k)}\} \cap H^1(\Omega).$$

The decomposition of the space V_h into basis function associated to Γ and to the interior of each patch plays an important role for deriving IETI methods. Let us define the spaces

$$V_{\Gamma,h} := \text{span}\{\check{N}_{i,p} \mid i \in \mathcal{I}_B\} \subset H^1(\Omega) \quad \text{and} \quad V_{I,h}^{(k)} := V_h^{(k)} \cap H_0^1(\Omega^{(k)}),$$

where $\mathcal{I}_B$ denotes all indices of basis functions having support on Γ. We are now able to state the desired decomposition

$$V_h = \prod_{k=1}^{N} V_{I,h}^{(k)} \oplus \mathcal{H}(V_{\Gamma,h}), \tag{7}$$

where $\mathcal{H}$ is the *discrete spline harmonic extension*, see [21] and references therein.

The Galerkin IgA scheme reads as follows: Find $u_h \in V_{D,h}$ such that

$$a(u_h, v_h) = \langle F, v_h \rangle \quad \forall v_h \in V_{D,h}, \tag{8}$$

where $V_{D,h} \subset V_D$ is the space of all functions from V_h which vanish on the Dirichlet boundary Γ_D. A basis for this space is given by the B-Spline functions $\{\check{N}_{i,p}\}_{i\in\mathcal{I}_0}$, where $\mathcal{I}_0$ contains all indices of $\mathcal{I}$ which do not have a support on the Dirichlet boundary Γ_D. Hence, the Galerkin IgA scheme (8) is equivalent to the linear system of algebraic equations

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{9}$$

where $\boldsymbol{K} = (\boldsymbol{K}_{i,j})_{i,j\in\mathcal{I}_0}$ and $\boldsymbol{f} = (\boldsymbol{f}_i)_{i\in\mathcal{I}_0}$ denote the stiffness matrix and the load vector, respectively, with $\boldsymbol{K}_{i,j} = a(\check{N}_{j,p}, \check{N}_{i,p})$ and $\boldsymbol{f}_i = \langle F, \check{N}_{i,p} \rangle$, and $\boldsymbol{u}$ is the vector representation of u_h given by the IgA isomorphism.

2.3 Discontinuous Galerkin IgA Scheme

The main principle of the dG-IgA Scheme is to use again the spaces $V_h^{(k)}$ of continuous functions on each patch $\Omega^{(k)}$, whereas discontinuities are allowed across the patch interface. The continuity of the function value and its normal derivative are then enforced in a weak sense by adding additional terms to the bilinear form. This situation is especially important when we consider non-matching grids on each patch. For the remainder of this paper, we define the dG-IgA space

$$V_h := V_h(\Omega) := \{v \,|\, v|_{\Omega^{(k)}} \in V_h^{(k)}\}, \tag{10}$$

where $V_h^{(k)}$ is defined as in (6).

We now follow the notation used in [13] and [15]. A comprehensive study of dG schemes for FE can be found in [35] and [11]. For an analysis of the dG-IgA scheme, we refer to [29].

Dirichlet boundary conditions can be handled in different ways. We can use the dG technique to incorporating them in a weak sense, see, e.g., [1] and [2]. This methods for imposing Dirichlet boundary conditions was already proposed by Nitsche [33]. Another method consists in enforcing them in a strong sense via an L^2 projection and homogenization. In this paper, for simplicity of presentation, we will follow the latter one, where we assume that the given Dirichlet data can be represented exactly with B-Splines. Hence, we define $V_{D,h}$ as the space of all functions from V_h which vanish on the Dirichlet boundary Γ_D. Furthermore, we denote the set of all indices l such that $\Omega^{(k)}$ and $\Omega^{(l)}$ have a common edge/face (2D/3D) $F^{(kl)}$ by $\mathcal{I}_{\mathcal{F}}^{(k)}$. Having these definitions at hand, we can define the discrete problem based on the Symmetric Interior Penalty (SIP) dG formulation as follows: Find $u_h \in V_{D,h}$ such that

$$a_h(u_h, v_h) = \langle F, v_h \rangle \quad \forall v_h \in V_{D,h}, \tag{11}$$

where

$$a_h(u,v) := \sum_{k=1}^{N} a_e^{(k)}(u,v) \quad \text{and} \quad \langle F, v \rangle := \sum_{k=1}^{N} \int_{\Omega^{(k)}} f v^{(k)} dx,$$

$$a_e^{(k)}(u,v) := a^{(k)}(u,v) + s^{(k)}(u,v) + p^{(k)}(u,v),$$

and

$$a^{(k)}(u,v) := \int_{\Omega^{(k)}} \alpha^{(k)} \nabla u^{(k)} \nabla v^{(k)} dx,$$

$$s^{(k)}(u,v) := \sum_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} \int_{F^{(kl)}} \frac{\alpha^{(k)}}{2} \left(\frac{\partial u^{(k)}}{\partial n}(v^{(l)} - v^{(k)}) + \frac{\partial v^{(k)}}{\partial n}(u^{(l)} - u^{(k)}) \right) ds,$$

$$p^{(k)}(u,v) := \sum_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} \int_{F^{(kl)}} \frac{\delta \alpha^{(k)}}{2h^{(kl)}}(u^{(l)} - u^{(k)})(v^{(l)} - v^{(k)}) \, ds.$$

The notation $\frac{\partial}{\partial n}$ means the derivative in the direction of the outer normal vector, δ is a positive sufficiently large penalty parameter, and $h^{(kl)}$ is the harmonic average of the adjacent mesh sizes, i.e., $h^{(kl)} = 2h^{(k)}h^{(l)}/(h^{(k)} + h^{(l)})$.

We equip $V_{D,h}$ with the broken Sobolev norm

$$\|u\|_{dG}^2 = \sum_{k=1}^{N} \left[\alpha^{(k)} \left\| \nabla u^{(k)} \right\|_{L^2(\Omega^{(k)})}^2 + \sum_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} \frac{\delta \alpha^{(k)}}{h^{(kl)}} \int_{F^{(kl)}} (u^{(k)} - u^{(l)})^2 ds \right].$$

Furthermore, we define the bilinear forms

$$d_h(u,v) = \sum_{k=1}^{N} d^{(k)}(u,v) \quad \text{and} \quad d^{(k)}(u,v) = a_e^{(k)}(u,v) + p^{(k)}(u,v).$$

for later use. We note that $\|u_h\|_{dG}^2 = d_h(u_h, u_h)$.

We are now able to show existence and uniqueness of a solution to (11). The following Lemma is an equivalent statement of Lemma 2.1 in [15] for IgA, and the proof is based on the results in [29].

Lemma 1. *Let δ be sufficiently large. Then there exist two positive constants γ_0 and γ_1 which are independent of $h^{(k)}, H^{(k)}, \delta, \alpha^{(k)}$ and u_h such that the inequalities*

$$\gamma_0 d^{(k)}(u_h, u_h) \le a_e^{(k)}(u_h, u_h) \le \gamma_1 d^{(k)}(u_h, u_h), \quad \forall u_h \in V_{D,h} \tag{12}$$

are valid for all $k = 1, 2, \ldots, N$. Furthermore, we have the inequalities

$$\gamma_0 \|u_h\|_{dG}^2 \le a_h(u_h, u_h) \le \gamma_1 \|u_h\|_{dG}^2, \quad \forall u_h \in V_{D,h}. \tag{13}$$

Proof. Rewriting the proofs of Lemma 4.6 and Lemma 4.7 in [29] for a single patch gives the desired inequalities (12). In order to show the boundedness, we additionally need to apply the discrete inverse inequality $\|\nabla u_h\|_{L^2(F^{(kl)})}^2 \le C/h^{(k)} \|\nabla u_h\|_{L^2(\Omega^{(k)})}^2$, see, e.g., [17], to the term $\sum_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} \alpha^{(k)} h^{(k)} \|\nabla u_h\|_{L^2(F^{(kl)})}^2$ appearing in the bound of Lemma 4.7 in [29]. Then we easily arrive at the estimate

$$\sum_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} \alpha^{(k)} h^{(k)} \|\nabla u_h\|_{L^2(F^{(kl)})}^2 \le C \sum_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} \alpha^{(k)} \|\nabla u_h\|_{L^2(\Omega^{(k)})}^2.$$

Hence, the right-hand side can be bounded by $d^{(k)}(u_h, u_h)$. Formula (13) immediately follows from (12), which concludes the proof. □

We note that, the results obtained in [29] are for the Incomplete Interior Penalty (IIP) scheme, an extension to SIP-dG and using harmonic averages for h and/or α is discussed in Remark 3.1. in [29], see also [28].

A direct implication of (13) is the well posedness of the discrete problem (8) by the Theorem of Lax-Milgram. The consistency of the method together with interpolation estimates for B-spline quasi-interpolant lead to the following a-priori error estimate, as established in [29].

Theorem 1. *Let $u \in H^1(\Omega) \cap \prod_{k=1}^{N} W^{l+1,q}(\Omega^{(k)})$ with $q \in (\min\{1, 2d/(d+2l)\}, 2]$ and some integer $l \ge 1$, solves (2), and let $u_h \in V_{D,h}$ solves the discrete problem (11). Then the discretization $u - u_h$ satisfies the estimate*

$$\|u - u_h\|_{dG}^2 \le \sum_{k=1}^{N} C^{(k)} \left(\left(h^{(k)}\right)^{2r} + \sum_{j \in \mathcal{I}_{\mathcal{F}}^{(k)}} \alpha^{(k)} \frac{h^{(k)}}{h^{(j)}} \left(h^{(k)}\right)^{2r} \right),$$

where $r = \min\{l + (\frac{d}{2} - \frac{d}{q}), p\}$, and $C^{(k)}$ is a positive constant which depends on p, $\|u\|_{W^{l+1,q}(\Omega^{(k)})}$, and $\max_{l_0 \leq l+1} \|\nabla^{l_0} G^{(k)}\|_{L^\infty(\Omega^{(k)})}$, but not on h. Here $W^{l+1,q}(\Omega^{(k)})$ denotes the Sobolev space of all functions from the space $L^q(\Omega^{(k)})$ such that all weak derivatives up to order $l + 1$ belong to $L^q(\Omega^{(k)})$ as well.

As explained in Section 2.2, we choose the B-Spline functions $\{\check{N}_{i,p}\}_{i \in \mathcal{I}_0}$ as basis for the space V_h, see (10), where $\mathcal{I}_0$ contains all indices of $\mathcal{I}$, which do not have a support on the Dirichlet boundary. Hence, the dG-IgA scheme (11) is equivalent to the system of linear equations

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{14}$$

where $\boldsymbol{K} = (\boldsymbol{K}_{i,j})_{i,j \in \mathcal{I}_0}$ and $\boldsymbol{f} = (\boldsymbol{f}_i)_{i \in \mathcal{I}_0}$ denote the stiffness matrix and the load vector, respectively, with $\boldsymbol{K}_{i,j} = a(\check{N}_{j,p}, \check{N}_{i,p})$ and $\boldsymbol{f}_i = \langle F, \check{N}_{i,p} \rangle$, and $\boldsymbol{u}$ is the vector representation of u_h.

3   IsoGeometric Tearing and Interconnecting for multipatch dG

Let us consider a multipatch domain, where the interfaces are geometrically matching, but not the meshes, i.e. the meshes can be different on different patches. Therefore, the considered solution and test space do not provide continuity across patch interfaces. Hence, we cannot enforce continuity of the solution by means of the jump operator as in the conforming IETI-DP. As proposed in [15], the remedy will be to introduce an additional layer of dofs on the interfaces and enforce continuity between the different layers. The considered method can then be seen as a conforming IETI-DP on an extended grid of dofs. We will follow the derivation presented in [15] with adopted notations. In the following, let V_h be the dG-IgA space which fulfils the Dirichlet boundary conditions as defined in Section 2.3 and we denote by $\{\check{N}_{i,p}\}_{i \in \mathcal{I}}$ the corresponding B-Spline basis.

3.1   Basic setup and local space description

As already introduced in Section 2.3, let $\mathcal{I}_{\mathcal{F}}^{(k)}$ be the set of all indices l such that $\Omega^{(k)}$ and $\Omega^{(l)}$ share a common edge/face. We may denote $\mathcal{I}_{\mathcal{F}}^{(k)}$ by $\mathcal{E}^{(k)}$ when considering 2D domains and by $\mathcal{F}^{(k)}$ for 3D domains. If we consider 3D objects, we additionally define $\overline{E}^{(klm)}$ as the edge shared by the patches $\Omega^{(k)}$, $\Omega^{(l)}$ and $\Omega^{(m)}$, i.e., $E^{(klm)} = \partial F^{(kl)} \cap \partial F^{(km)}$ for $l \in \mathcal{F}^{(k)}, m \in \mathcal{F}^{(k)}$. The set of all indices (l, m) of $\Omega^{(l)}$ and $\Omega^{(m)}$, such that $\overline{E}^{(klm)}$ is an edge of patch $\Omega^{(k)}$ is denoted by $\mathcal{E}^{(k)}$. Note, although $\overline{F}_{lk} \subset \partial\Omega^{(l)}$ and $\overline{F}_{kl} \subset \partial\Omega^{(k)}$ are geometrically the same, they are treated as different objects. The same applies for edges $\overline{E}^{(klm)}, \overline{E}^{(lkm)}$ and $\overline{E}^{(mkl)}$. In order to keep the presentation of the method simple, we assume that the considered patch $\Omega^{(k)}$ does not touch the Dirichlet boundary. The other case can be handled in an analogous way.

As already introduced above, the computational domain Ω is given by $\overline{\Omega} = \bigcup_{k=1}^N \overline{\Omega}^{(k)}$, where $\Omega^{(k)} = G^{(k)}\left((0,1)^d\right)$ for $k = 1, \ldots, N$, and the interface by $\Gamma^{(k)} = \overline{\partial\Omega^{(k)} \backslash \partial\Omega}$. For each patch $\Omega^{(k)}$, we introduce its extended version $\Omega_e^{(k)}$ via the union with all neighbouring interfaces $\overline{F}_{lk} \subset \partial\Omega^{(l)}$:

$$\overline{\Omega}_e^{(k)} := \overline{\Omega}^{(k)} \cup \Big\{ \bigcup_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} \overline{F}^{(lk)} \Big\}.$$

Moreover, the extended interface $\Gamma_e^{(k)}$ is given by the union of $\Gamma^{(k)}$ with all neighbouring interfaces $\overline{F}_{lk} \subset \partial\Omega^{(k)}$:

$$\Gamma_e^{(k)} := \Gamma^{(k)} \cup \{ \bigcup_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} \overline{F}^{(lk)} \}.$$

Based on the definitions above, we can introduce

$$\overline{\Omega}_e = \bigcup_{k=1}^{N} \overline{\Omega}_e^{(k)}, \quad \Gamma = \bigcup_{k=1}^{N} \Gamma^{(k)} \text{ and } \quad \Gamma_e = \bigcup_{k=1}^{N} \Gamma_e^{(k)}.$$

The next step is to describe appropriate discrete function spaces to reformulate (11) in order to treat the new formulation in the spirit of the conforming IETI-DP method. We start with a description of the discrete function spaces for a single patch.

As defined in (6), let $V_h^{(k)}$ be the discrete function space defined on the patch $\Omega^{(k)}$. Then we define the corresponding function space for the extended patch $\Omega_e^{(k)}$ by

$$V_{h,e}^{(k)} := V_h^{(k)} \times \prod_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} V_h^{(k)}(\overline{F}^{(lk)}),$$

where $V_h^{(k)}(\overline{F}^{(lk)}) \subset V_h^{(l)}$ is given by

$$V_h^{(k)}(\overline{F}^{(lk)}) := \text{span}\{\check{N}_{i,p}^{(l)} \mid \text{supp}\{\check{N}_{i,p}^{(l)}\} \cap \overline{F}^{(lk)} \neq \emptyset\}.$$

According to the notation introduced in [15], we will represent a function $u^{(k)} \in V_{h,e}^{(k)}$ as

$$u^{(k)} = \{(u^{(k)})^{(k)}, \{(u^{(k)})^{(l)}\}_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}}\}, \tag{15}$$

where $(u^{(k)})^{(k)}$ and $(u^{(k)})^{(l)}$ are the restrictions of $u^{(k)}$ to $\Omega^{(k)}$ and $\overline{F}^{(lk)}$, respectively. Moreover, we introduce an additional representation of $u^{(k)} \in V_{h,e}^{(k)}$, as $u^{(k)} = (u_I^{(k)}, u_{B_e}^{(k)})$, where

$$u_I^{(k)} \in V_{I,h}^{(k)} := V_h^{(k)} \cap H_0^1(\Omega^{(k)}),$$

and

$$u_{B_e}^{(k)} \in W^{(k)} := \text{span}\{\check{N}_{i,p}^{(l)} \mid \text{supp}\{\check{N}_{i,p}^{(l)}\} \cap \Gamma_e^{(k)} \neq \emptyset \text{ for } l \in \mathcal{I}_{\mathcal{F}}^{(k)} \cup \{k\}\}.$$

This provides a representation of $V_{h,e}^{(k)}$ in the form of $V_{I,h}^{(k)} \times W^{(k)}$.

3.2   Schur complement and discrete harmonic extensions

We note that the bilinear form $a_e^{(k)}(\cdot, \cdot)$ is defined on the space $V_{h,e}^{(k)} \times V_{h,e}^{(k)}$, since it requires function values of the neighbouring patches $\Omega^{(l)}, l \in \mathcal{I}_{\mathcal{F}}^{(k)}$. Therefore, it depicts a matrix representation $\boldsymbol{K}_e^{(k)}$ satisfying the identity

$$a_e^{(k)}(u^{(k)}, v^{(k)}) = (\boldsymbol{K}_e^{(k)} \boldsymbol{u}, \boldsymbol{v})_{l_2} \quad \text{for } u^{(k)}, v^{(k)} \in V_{h,e}^{(k)},$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ denote the vector representation of $u^{(k)}$ and $v^{(k)}$, respectively. By means of the representation $V_{I,h}^{(k)} \times W^{(k)}$ for $V_{h,e}^{(k)}$, we can partition the matrix $\boldsymbol{K}_e^{(k)}$ as

$$\boldsymbol{K}_e^{(k)} = \begin{bmatrix} \boldsymbol{K}_{e,II}^{(k)} & \boldsymbol{K}_{e,IB_e}^{(k)} \\ \boldsymbol{K}_{e,B_eI}^{(k)} & \boldsymbol{K}_{e,B_eB_e}^{(k)} \end{bmatrix}. \tag{16}$$

This enables us to define the Schur complement of $\boldsymbol{K}_e^{(k)}$ with respect to $W^{(k)}$ as

$$\boldsymbol{S}_e^{(k)} := \boldsymbol{K}_{e,B_eB_e}^{(k)} - \boldsymbol{K}_{e,B_eI}^{(k)} \left(\boldsymbol{K}_{e,II}^{(k)} \right)^{-1} \boldsymbol{K}_{e,IB_e}^{(k)}. \tag{17}$$

We denote the corresponding bilinear form by $s_e^{(k)}(\cdot, \cdot)$, and the corresponding operator by $S_e^{(k)} : W^{(k)} \to W^{(k)*}$, i.e.

$$(\boldsymbol{S}_e^{(k)} \boldsymbol{u}_{B_e}^{(k)}, \boldsymbol{v}_{B_e}^{(k)})_{l^2} = \left\langle S_e^{(k)}, u_{B_e}^{(k)}, u_{B_e}^{(k)} \right\rangle = s_e^{(k)}(u_{B_e}^{(k)}, u_{B_e}^{(k)}), \quad \forall u_{B_e}^{(k)}, u_{B_e}^{(k)} \in W^{(k)}.$$

The Schur complement has the property that

$$\left\langle S_e^{(k)}, u_{B_e}^{(k)}, u_{B_e}^{(k)} \right\rangle = \min_{w^{(k)} = (w_I^{(k)}, w_{B_e}^{(k)}) \in V_{h,e}^{(k)}} a_e^{(k)}(w^{(k)}, w^{(k)}), \tag{18}$$

such that $w_{B_e}^{(k)} = u_{B_e}^{(k)}$ on $\Gamma_e^{(k)}$. We define the *discrete NURBS harmonic extension* $\mathcal{H}_e^{(k)}$ (in the sense of $a_e^{(k)}(\cdot, \cdot)$) for patch $\Omega_e^{(k)}$ by

$$\mathcal{H}_e^{(k)} : W^{(k)} \to V_{h,e}^{(k)} :$$
$$\begin{cases} \text{Find } \mathcal{H}_e^{(k)} u_{B_e} \in V_{h,e}^{(k)} : \\ a_e^{(k)}(\mathcal{H}_e^{(k)} u_{B_e}, u^{(k)}) = 0 \qquad \forall u^{(k)} \in V_{I,h}^{(k)}, \\ \mathcal{H}_e^{(k)} u_{B_e|\Gamma^{(k)}} = u_{B_e|\Gamma^{(k)}}, \end{cases} \tag{19}$$

where $V_{I,h}^{(k)}$ is here interpreted as subspace of $V_{h,e}^{(k)}$ with vanishing function values on $\Gamma_e^{(k)}$. One can show that the minimizer in (18) is given by $\mathcal{H}_e^{(k)} u_{B_e}$. In addition, we introduce the *standard discrete NURBS harmonic extension* $\mathcal{H}^{(k)}$ (in the sense of $a^{(k)}(\cdot, \cdot)$) of $u_{B_e}^{(k)}$ as follows:

$$\mathcal{H}^{(k)} : W^{(k)} \to V_{h,e}^{(k)} :$$
$$\begin{cases} \text{Find } \mathcal{H}^{(k)} u_{B_e} \in V_{h,e}^{(k)} : \\ a^{(k)}(\mathcal{H}^{(k)} u_{B_e}, u^{(k)}) = 0 \qquad \forall u^{(k)} \in V_{I,h}^{(k)}, \\ \mathcal{H}^{(k)} u_{B_e|\Gamma^{(k)}} = u_{B_e|\Gamma^{(k)}}, \end{cases} \tag{20}$$

where $V_{I,h}^{(k)}$ is the same space as in (19), and $a^{(k)}(\cdot, \cdot)$ is a bilinear form on the space $V_{h,e}^{(k)} \times V_{h,e}^{(k)}$. The crucial point is to show equivalence in the energy norm $d_h(u_h, u_h)$ between functions, which are discrete harmonic in the sense of $\mathcal{H}_e^{(k)}$ and $\mathcal{H}^{(k)}$. This property is summarized in the following Lemma, cf. also Lemma 3.1 in [15].

Lemma 2. *There exists a positive constant which is independent of $\delta, h^{(k)}, H^{(k)}, \alpha^{(k)}$ and $u_{B_e}^{(k)}$ such that the inequalities*

$$d^{(k)}(\mathcal{H}^{(k)} u_{B_e}, \mathcal{H}^{(k)} u_{B_e}) \leq d^{(k)}(\mathcal{H}_e^{(k)} u_{B_e}, \mathcal{H}_e^{(k)} u_{B_e}) \leq C d^{(k)}(\mathcal{H}^{(k)} u_{B_e}, \mathcal{H}^{(k)} u_{B_e}), \tag{21}$$

hold for all $u_{B_e}^{(k)} \in W^{(k)}$.

Proof. The proof is identical to that one presented in [14] for Lemma 4.1 up to the point where we have to use the discrete trace inequality

$$\|u_h\|_{L^2(\partial \Omega^{(k)})}^2 \le C h^{-1} \|u_h\|_{L^2(\Omega^{(k)})}^2, \quad \forall u_h \in V_h^{(k)},$$

for IgA function spaces, see, e.g., [17]. $\qquad \square$

The subsequent statement immediately follows from Lemma 1 and Lemma 2, see also [15].

Corollary 1. *The spectral equivalence inequalities*

$$C_0 d^{(k)}(\mathcal{H}^{(k)} u_{B_e}, \mathcal{H}^{(k)} u_{B_e}) \le a_e^{(k)}(\mathcal{H}_e^{(k)} u_{B_e}, \mathcal{H}_e^{(k)} u_{B_e}) \le C_1 d^{(k)}(\mathcal{H}^{(k)} u_{B_e}, \mathcal{H}^{(k)} u_{B_e}), \quad (22)$$

hold for all $u_{B_e}^{(k)} \in W^{(k)}$, where the constants C_0 and C_1 are independent of $\delta, h^{(k)}, H^{(k)}, \alpha^{(k)}$ and $u_{B_e}^{(k)}$.

3.3 Global space description

Based on the definitions of the local spaces in Section 3.1, we can introduce the corresponding spaces

$$V_{h,e} := \{v \mid v^{(k)} \in V_{h,e}^{(k)}, k \in \{1, \dots, N\}\}.$$

for the whole extended domain Ω_e. Additionally, we need a description of the global extended interface spaces

$$W := \{v_{B_e} \mid v_{B_e}^{(k)} \in W^{(k)}, k \in \{1, \dots, N\}\} = \prod_{k=1}^{N} W^{(k)}.$$

We note that according to [15], we will also interpret this space as subspace of $V_{h,e}$, where its functions are discrete harmonic in the sense of $\mathcal{H}_e^{(k)}$ on each $\Omega^{(k)}$. For completeness, we define the discrete NURBS harmonic extension in the sense of $\sum_{k=1}^{N} a_e^{(k)}(\cdot, \cdot)$ and $\sum_{k=1}^{N} a^{(k)}(\cdot, \cdot)$ for W as $\mathcal{H}_e u = \{\mathcal{H}_e^{(k)} u^{(k)}\}_{k=1}^{N}$ and $\mathcal{H}_e u = \{\mathcal{H}^{(k)} u^{(k)}\}_{k=1}^{N}$, respectively.

The goal is to reformulate (11) and (14) in terms of the extended domain Ω_e. In order to achieve this, we need a coupling of the now independent interface dofs. In the context of tearing and interconnecting methods, we need a "continuous" subspace $\widehat{W}$ of W such that $\widehat{W}$ is equivalent to $V_{\Gamma,h}$, i.e., $\widehat{W} \equiv V_{\Gamma,h}$. Since the space $V_{\Gamma,h}$ consists of functions which are discontinuous across the patch interface, the common understanding of continuity makes no sense. We follow the way in [14], providing an appropriate definition of continuity in the context of the spaces $\widehat{W}, W, V_{\Gamma,h}, V_{h,e}$ and V_h.

Definition 1. *We say that $u \in V_{h,e}$ is continuous on Γ_e if the relations*

$$(\boldsymbol{u}^{(k)})_i^{(k)} = (\boldsymbol{u}^{(l)})_j^{(k)} \quad \forall(i,j) \in B_e(k,l), \; \forall l \in \mathcal{I}_{\mathcal{F}}^{(k)}, \quad (23)$$

and

$$(\boldsymbol{u}^{(k)})_i^{(l)} = (\boldsymbol{u}^{(l)})_j^{(l)} \quad \forall(i,j) \in B_e(l,k), \; \forall l \in \mathcal{I}_{\mathcal{F}}^{(k)}. \quad (24)$$

hold for all $k \in \{1, \dots, N\}$. We denote the set of index pairs (i,j) such that the i-th basis function in $V_h^{(k)}$ can be identified with the j-th basis function in $V_h^{(l)}(\overline{F}^{(kl)})$ by $B_e(k,l)$. We note that $B_e(k,l) \ne B_e(l,k)$. Moreover, $\widehat{V}_{h,e}$ denotes the subspace of continuous functions on Γ_e of $V_{h,e}$. Furthermore, $\widehat{V}_{h,e}$ can be identified with V_h.

The operator $B : W \to U^* := \mathbb{R}^A$, which realizes constraints (23) and (24) in the form

$$Bu = 0,$$

is called *jump operator*. The space of all functions in W which belong to the kernel of B is denoted by $\widehat{W}$, and can be identified with $V_{\Gamma,h}$, i.e.

$$\widehat{W} = \{w \in W \,|\, Bw = 0\} \equiv V_{\Gamma,h}.$$

Furthermore, we define the restriction of $\widehat{W}$ to $\Omega_e^{(k)}$ by $\widehat{W}^{(k)}$.

Remark 1. According to [15], the space $\widehat{W}$ can also be interpreted as the set of all functions in $\widehat{V}_{h,e}$, which are discrete harmonic in the sense of $\mathcal{H}_e$.

We introduce the set of patch vertices $\mathcal{V}$ and the corresponding extended set $\mathcal{V}_e$, given by

$$\mathcal{V}_e^{(k)} = \mathcal{V}^{(k)} \cup \{ \bigcup_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} \partial F^{(lk)} \}, \text{ where } \quad \mathcal{V}^{(k)} = \{ \bigcup_{l \in \mathcal{I}_{\mathcal{F}}^{(k)}} \partial F^{(kl)} \},$$

for 2D domains and by

$$\mathcal{V}_e^{(k)} = \mathcal{V}^{(k)} \cup \{ \bigcup_{(l,m) \in \mathcal{E}^{(k)}} \partial E^{(lkm)} \cup \partial E^{(mkl)} \}, \text{ where } \quad \mathcal{V}^{(k)} = \{ \bigcup_{(l,m) \in \mathcal{E}^{(k)}} \partial E^{(klm)} \},$$

for 3D domains. The set $\mathcal{V}$ and $\mathcal{V}_e$ is then given by the union of all $\mathcal{V}^{(k)}$ and $\mathcal{V}_e^{(k)}$, respectively. Moreover, we denote by $\mathcal{V}_e^{(kl)} \subset \mathcal{V}_e$ all vertices which belong to the interface $F^{(kl)}$.

Now we are in the position to reformulate (14) in terms of $\widehat{V}_{h,e}$, leading to the system

$$\widehat{\boldsymbol{K}}_e \boldsymbol{u}_e = \widehat{\boldsymbol{f}}_e, \tag{25}$$

where the matrix $\widehat{\boldsymbol{K}}_e$ is given by the assembly of the patchwise matrices $\boldsymbol{K}_e^{(k)}$, i.e.

$$\widehat{\boldsymbol{K}}_e = \sum_{k=1}^{N} \boldsymbol{A}_{\Omega_e^{(k)}} \boldsymbol{K}_e^{(k)} \boldsymbol{A}_{\Omega_e^{(k)}}^T \text{ and } \quad \widehat{\boldsymbol{f}}_e = \sum_{k=1}^{N} \boldsymbol{A}_{\Omega_e^{(k)}} \boldsymbol{f}_e^{(k)}. \tag{26}$$

Here $\boldsymbol{A}_{\Omega_e^{(k)}}$ denotes the Boolean patch assembling matrix for $\Omega_e^{(k)}$. By means of the local Schur complements $\boldsymbol{S}_e^{(k)}$, see (16) and (17), we can reformulate equation (25) as

$$\widehat{\boldsymbol{S}}_e u_{B_e} = \widehat{\boldsymbol{g}}_e, \tag{27}$$

where $\widehat{\boldsymbol{S}}_e$ and $\widehat{\boldsymbol{g}}_e$ are given by

$$\widehat{\boldsymbol{S}}_e = \left(\sum_{k=1}^{N} \boldsymbol{A}_{\Gamma_e^{(k)}} \boldsymbol{S}_e^{(k)} \boldsymbol{A}_{\Gamma_e^{(k)}}^T \right) \text{ and } \quad \widehat{\boldsymbol{g}}_e = \sum_{k=1}^{N} \boldsymbol{A}_{\Gamma_e^{(k)}} \boldsymbol{g}_e^{(k)}. \tag{28}$$

The Boolean matrix $\boldsymbol{A}_{\Gamma_e^{(k)}}$ is the corresponding assembling matrix and the vector $\boldsymbol{g}^{(k)}$ is defined by $\boldsymbol{g}^{(k)} = \boldsymbol{f}_{e,B_e}^{(k)} - \boldsymbol{K}_{e,B_e I}^{(k)} \left(\boldsymbol{K}_{e,II}^{(k)} \right)^{-1} \boldsymbol{f}_{e,I}^{(k)}$. Furthermore, we can express (28) in operator notation as

$$\sum_{k=1}^{N} \left\langle S_e^{(k)} u_{B_e}^{(k)}, v^{(k)} \right\rangle = \sum_{k=1}^{N} \left\langle g_e^{(k)}, v^{(k)} \right\rangle \quad \forall v \in \widehat{W}, \tag{29}$$

where $u_{B_e} \in \widehat{W}$, $g_e^{(k)} \in \widehat{W}^{(k)^*}$ and $S_e^{(k)} : \widehat{W}^{(k)} \to \widehat{W}^{(k)^*}$.

In order to formulate the IETI-DP algorithm, we also define the Schur complement and the right-hand side functional on the "discontinuous" space W, i.e.

$$S_e : W \to W^*, \quad \langle S_e v, w \rangle := \sum_{k=1}^{N} \left\langle S_e^{(k)} v^{(k)}, w^{(k)} \right\rangle \quad \forall v, w \in W,$$

and

$$g_e \in W^*, \quad \langle g_e, w \rangle := \sum_{k=1}^{N} \left\langle g_e^{(k)}, w^{(k)} \right\rangle \quad \forall w \in W.$$

In matrix form, we can write $\boldsymbol{S}$ and $\boldsymbol{g}$ as

$$\boldsymbol{S}_e := \mathrm{diag}(\boldsymbol{S}_e^{(k)})_{k=1}^{N} \text{ and } \quad \boldsymbol{g}_e := [\boldsymbol{g}_e^{(k)}]_{k=1}^{N}.$$

It is easy to see that problem (27) is equivalent to the minimization problem

$$u_{B_e,h} = \operatorname*{argmin}_{w \in W, Bw=0} \frac{1}{2} \langle S_e w, w \rangle - \langle g_e, w \rangle. \tag{30}$$

In the following we will only work with the Schur complement system. In order to to simplify the notation, we will use u instead of $u_{B,h}$, when we consider functions in $V_{\Gamma,h}$. If we have to made a distinction between u_h, $u_{B,h}$ and $u_{I,h}$, we will add the subscripts again.

3.4 Intermediate space and primal constraints

The key point of the dual-primal approach is the definition of an intermediate space $\widetilde{W}$ in the sense $\widehat{W} \subset \widetilde{W} \subset W$ such that S_e restricted to $\widetilde{W}$ is positive definite. Let $\Psi \subset V_{\Gamma,h}^*$ be a set of linearly independent *primal variables*. Then we define the spaces

$$\widetilde{W} := \{w \in W : \forall \psi \in \Psi : \psi(w^{(k)}) = \psi(w^{(l)}), \forall k > l\}$$

and

$$W_\Delta := \prod_{k=1}^{N} W_\Delta^{(k)}, \text{ with } \quad W_\Delta^{(k)} := \{w^{(k)} \in W^{(k)} : \forall \psi \in \Psi : \psi(w^{(k)}) = 0\}.$$

Moreover, we introduce the space $W_\Pi \subset \widehat{W}$ such that $\widetilde{W} = W_\Pi \oplus W_\Delta$. We call W_Π *primal space* and W_Δ *dual space*. If we choose Ψ such that $\widetilde{W} \cap \ker(S_e) = \{0\}$, then

$$\widetilde{S}_e : \widetilde{W} \to \widetilde{W}^*, \text{ with } \left\langle \widetilde{S}_e v, w \right\rangle = \langle S_e v, w \rangle \quad \forall v, w \in \widetilde{W},$$

is invertible. If a set Ψ fulfils $\widetilde{W} \cap \ker(S_e) = \{0\}$, then we say that the set Ψ *controls the kernel*. In the following, we will always assume that such a set is chosen. We will work with the following typical choices for the primal variables ψ:

- Vertex evaluation: $\psi^{\mathcal{V}}(v) = v(\mathcal{V})$,

- Edge averages: $\psi^{\mathcal{E}}(v) = \frac{1}{|\mathcal{E}|} \int_{\mathcal{E}} v \, ds$,

– Face averages: $\psi^{\mathcal{F}}(v) = \frac{1}{|\mathcal{F}|} \int_{\mathcal{F}} v\, ds$.

Since we are considering a non-conforming test space $\widehat{W}$ and V_h we cannot literally use the same set of primal variables as presented in [36], [34], or [21]. As proposed in [15] and [16], we will use the following interpretation of continuity at corners, and continuous edge and face averages.

Definition 2. *Let $\mathcal{V}_e^{(k)}$, $\mathcal{E}^{(k)}$ and $\mathcal{F}^{(k)}$ be the set of vertices, edges and faces, respectively, for the patch $\Omega_e^{(k)}$, where in 2D the set $\mathcal{E}^{(k)}$ is empty.*

We say that $u \in W$ is continuous at $\mathcal{V}^{(k)}$, $k \in \{1, \ldots, N\}$, if the relations

$$(\boldsymbol{u}^{(k)})_i^{(k)} = (\boldsymbol{u}^{(l)})_j^{(k)} \quad \forall (i,j) \in B_{\mathcal{V}}(k,l) \tag{31}$$

are valid for all $l \in \mathcal{I}_{\mathcal{F}}^{(k)}$, where $B_{\mathcal{V}}(k,l) \subset B(k,l)$ is given by all index pairs corresponding to the vertices $\mathcal{V}_e^{(k)}$. We define the corresponding primal variable as

$$\psi^{\nu^{(kl)}}(v) := \begin{cases} (\boldsymbol{v}^{(k)})_i^{(k)} & \text{if } v \in W^{(k)}, \\ (\boldsymbol{v}^{(l)})_j^{(k)} & \text{if } v \in W^{(l)}, \\ 0 & \text{else}, \end{cases} \tag{32}$$

where $l \in \mathcal{I}_{\mathcal{F}}^{(k)}$, $\nu^{(kl)} \in \mathcal{V}_e^{(kl)}$ and $(i,j) \in B_{\mathcal{V}}(k,l)$ corresponds to $\nu^{(kl)}$.

We say that $u \in W$ has continuous (inter-)face averages at $\mathcal{F}^{(k)}$, $k \in \{1, \ldots, N\}$, if the relations

$$\frac{1}{|F^{(kl)}|} \int_{F^{(kl)}} (u^{(k)})^{(k)}\, ds = \frac{1}{|F^{(kl)}|} \int_{F^{(kl)}} (u^{(l)})^{(k)}\, ds \tag{33}$$

hold for all $l \in \mathcal{I}_{\mathcal{F}}^{(k)}$. We define the corresponding primal variable as

$$\psi^{F^{(kl)}}(v) := \begin{cases} \frac{1}{|F^{(kl)}|} \int_{F^{(kl)}} (u^{(k)})^{(k)}\, ds & \text{if } v \in W^{(k)}, \\ \frac{1}{|F^{(kl)}|} \int_{F^{(kl)}} (u^{(l)})^{(k)}\, ds & \text{if } v \in W^{(l)}, \\ 0 & \text{else}, \end{cases} \tag{34}$$

where $l \in \mathcal{I}_{\mathcal{F}}^{(k)}$.

We say that $u \in W$ has continuous edge averages at $\mathcal{E}^{(k)}$, $k \in \{1, \ldots, N\}$, if the relations

$$\frac{1}{|E^{(klm)}|} \int_{E^{(klm)}} (u^{(k)})^{(k)}\, ds = \frac{1}{|E^{(klm)}|} \int_{E^{(klm)}} (u^{(l)})^{(k)}\, ds, \tag{35}$$

$$\frac{1}{|E^{(klm)}|} \int_{E^{(klm)}} (u^{(k)})^{(k)}\, ds = \frac{1}{|E^{(klm)}|} \int_{E^{(klm)}} (u^{(m)})^{(k)}\, ds \tag{36}$$

hold for all $(l,m) \in \mathcal{E}^{(k)}$. We define the corresponding primal variable as

$$\psi^{E^{(klm)}}(v) := \begin{cases} \frac{1}{|E^{(klm)}|} \int_{E^{(klm)}} (u^{(k)})^{(k)}\, ds & \text{if } v \in W^{(k)}, \\ \frac{1}{|E^{(klm)}|} \int_{E^{(klm)}} (u^{(l)})^{(k)}\, ds & \text{if } v \in W^{(l)}, \\ \frac{1}{|E^{(klm)}|} \int_{E^{(klm)}} (u^{(m)})^{(k)}\, ds & \text{if } v \in W^{(m)}, \\ 0 & \text{else}, \end{cases} \tag{37}$$

where $(l,m) \in \mathcal{E}^{(k)}$.

By means of Definition 2, we can now introduce different sets of primal variables Ψ:

- Algorithm A: $\Psi^A := \{\psi^\nu, \ \forall \nu \in \mathcal{V}^{(k)}\}_{k=1}^N$,
- Algorithm B: $\Psi^B := \{\psi^\nu, \ \forall \nu \in \mathcal{V}^{(k)}\}_{k=1}^N \cup \{\psi^E, \ \forall E \in \mathcal{E}^{(k)}\}_{k=1}^N \cup \{\psi^F, \ \forall F \in \mathcal{E}^{(k)}\}_{k=1}^N$,
- Algorithm C: $\Psi^C := \{\psi^\nu, \ \forall \nu \in \mathcal{V}^{(k)}\}_{k=1}^N \cup \{\psi^E, \ \forall E \in \mathcal{E}^{(k)}\}_{k=1}^N$.

3.5 IETI - DP and preconditioning

Since $\widetilde{W} \subset W$, there is a natural embedding $\widetilde{I} : \widetilde{W} \to W$. Let the jump operator restricted to $\widetilde{W}$ be

$$\widetilde{B} := B\widetilde{I} : \widetilde{W} \to U^*. \tag{38}$$

Then we can formulate problem (30) as saddle point problem in $\widetilde{W}$ as follows: Find $(u, \boldsymbol{\lambda}) \in \widetilde{W} \times U$:

$$\begin{bmatrix} \widetilde{S}_e & \widetilde{B}^T \\ \widetilde{B} & 0 \end{bmatrix} \begin{bmatrix} u \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \widetilde{g} \\ 0 \end{bmatrix}, \tag{39}$$

where $\widetilde{g} := \widetilde{I}^T g$, and $\widetilde{B}^T = \widetilde{I}^T B^T$. Here, $\widetilde{I}^T : W^* \to \widetilde{W}^*$ denotes the adjoint of $\widetilde{I}$, which can be seen as a partial assembling operator.

By construction, $\widetilde{S}_e$ is SPD on $\widetilde{W}$. Hence, we can define the Schur complement F and the corresponding right-hand side of equation (39) as follows:

$$F := \widetilde{B}\widetilde{S}_e^{-1}\widetilde{B}^T, \quad d := \widetilde{B}\widetilde{S}_e^{-1}\widetilde{g}.$$

Hence, the saddle point system (39) is equivalent to the Schur complement problem:

$$\text{Find } \boldsymbol{\lambda} \in U : \quad F\boldsymbol{\lambda} = d. \tag{40}$$

Equation (40) is solved by means of the PCG algorithm, but it requires an appropriate preconditioner in order to obtain an efficient solver. According to [15] and [16], the right choice for FE is the *scaled Dirichlet preconditioner*, adapted for the extended set of dofs. The numerical tests presented in Section 4 indicate that the scaled Dirichlet preconditioner works well for the IgA setting too.

Recall the definition of $S_e = \text{diag}(S_e^{(k)})_{k=1}^N$, we define the scaled Dirichlet preconditioner M_{sD}^{-1} as

$$M_{sD}^{-1} = B_D S_e B_D^T, \tag{41}$$

where B_D is a scaled version of the jump operator B. The scaled jump operator B_D is defined such that the operator enforces the constraints

$$\delta^\dagger{}_j^{(l)}(\boldsymbol{u}^{(k)})_i^{(k)} - \delta^\dagger{}_i^{(k)}(\boldsymbol{u}^{(l)})_j^{(k)} = 0 \quad \forall (i,j) \in B_e(k,l), \ \forall l \in \mathcal{I}_{\mathcal{F}}^{(k)}, \tag{42}$$

and

$$\delta^\dagger{}_j^{(l)}(\boldsymbol{u}^{(k)})_i^{(l)} - \delta^\dagger{}_i^{(k)}(\boldsymbol{u}^{(l)})_j^{(l)} = 0 \quad \forall (i,j) \in B_e(l,k), \ \forall l \in \mathcal{I}_{\mathcal{F}}^{(k)}, \tag{43}$$

where for $(i, j) \in B_e(k, l)$

$$\delta^{\dagger (k)}_i = \frac{\rho^{(k)}_i}{\sum_{l \in \mathcal{I}^{(k)}_{\mathcal{F}}} \rho^{(l)}_j}$$

is an appropriate scaling. Typical choices for $\rho^{(k)}_i$ are

- Multiplicity Scaling: $\rho^{(k)}_i = 1$,

- Coefficient Scaling: If $\alpha(x)_{|\Omega^{(k)}} = \alpha^{(k)}$, choose $\rho^{(k)}_i = \alpha^{(k)}$,

- Stiffness Scaling: $\rho^{(k)}_i = \boldsymbol{K}^{(k)}_{e_{i,i}}$.

If the diffusion coefficient α is constant and identical on each patch, then the multiplicity and the coefficient scaling are the same. If there is only a little variation in α, then the multiplicity scaling provides good results. If the variation is really large, then one should use the other scalings to obtain robustness with respect to the jumps in the diffusion coefficient across the patch interfaces.

In order to realizes the method, one can use the same procedure as for the continuous IETI-DP method, where one has to use the corresponding definitions described in this paper. One possibility for the implementation can be found in [21], where the Schur complement is denoted by S and $\widetilde{S}$.

In [15] and [16] it is proven for FE that the condition number behaves like the condition number of the preconditioned system for the continuous FETI-DP method, see also [14] for dG-BDDC FE preconditioners. From [21] and [6], we know that the condition number of the continuous IETI-DP and BDDC-IgA operators is also quasi-optimal with respect to the patch and mesh sizes. Therefore, we expect that the condition number of the dG-IETI-DP operator behaves as

$$\kappa(M^{-1}_{sD} F_{\widetilde{U}}) \leq C \max_k \left(1 + \log \left(\frac{H^{(k)}}{h^{(k)}} \right) \right)^2,$$

where $H^{(k)}$ and $h^{(k)}$ are the patch size and mesh size, respectively, and the positive constant C is independent of $H^{(k)}$, $h^{(k)}$, $h^{(k)}/h^{(l)}$, and α. Our numerical results presented in the next section insistently confirm this behaviour.

4 Numerical examples

In this section, we present some numerical results documenting the numerical behaviour of the implemented dG-IETI-DP algorithm for solving large-scale linear systems arising from higher-order IgA discretizations of (1) in the domains illustrated in Figure 1(a) and Figure 1(b). The computational domain consists of 21 subdomains in both 2D and 3D. In both cases, one side of a patch boundary has inhomogeneous Dirichlet conditions, whereas all other sides have homogeneous Neumann conditions. We consider the case of non-matching meshes, i.e. two neighbouring patches may have different mesh sizes $h^{(k)}$ and $h^{(l)}$. Due to our implementation of the dG formulation, we only consider nested meshes on the interface, i.e. the B-Spline spaces on the interfaces are nested. However, we note that the presented algorithm does not rely on this assumption. Each subdomain has a diameter of $H^{(k)}$ and an associated mesh size of $h^{(k)}$. In the following, we use the abbreviation $H/h = \max_k H^{(k)}/h^{(k)}$. We consider B-Splines, where its

degree is chosen as $p = 2$ and $p = 4$. In all numerical examples when increasing the degree from 2 to 4, we keep the smoothness of the space, i.e. increasing the multiplicity of the knots on the coarsest mesh. In order to solve the linear system (40), a PCG algorithm with the scaled Dirichlet preconditioner (41) is performed. We use a zero initial guess, and a reduction of the initial residual by a factor of 10^{-6} as stopping criterion. The numerical examples illustrate the dependence of the condition number of the IETI-DP preconditioned system on jumps in the diffusion coefficient α, patch size H, mesh size h and the degree p. In Section 4.3, we investigate the special case of increasing $h^{(k)}/h^{(l)}$ and its influence on the condition number. In all other tests we consider a fixed the ratio $h^{(k)}/h^{(l)}$.

We use the C++ library G+SMO[1] for describing the geometry and performing the numerical tests, see also [25] and [30]. The stars "$\ast$" in Tables 2, 4 and 5 mean that the problem size doesn't fit into our Desktop PC with an Intel(R) Xeon(R) CPU E5-1650 v2 @ 3.50GHz and 16 GB main memory, on which we performed all numerical experiments.
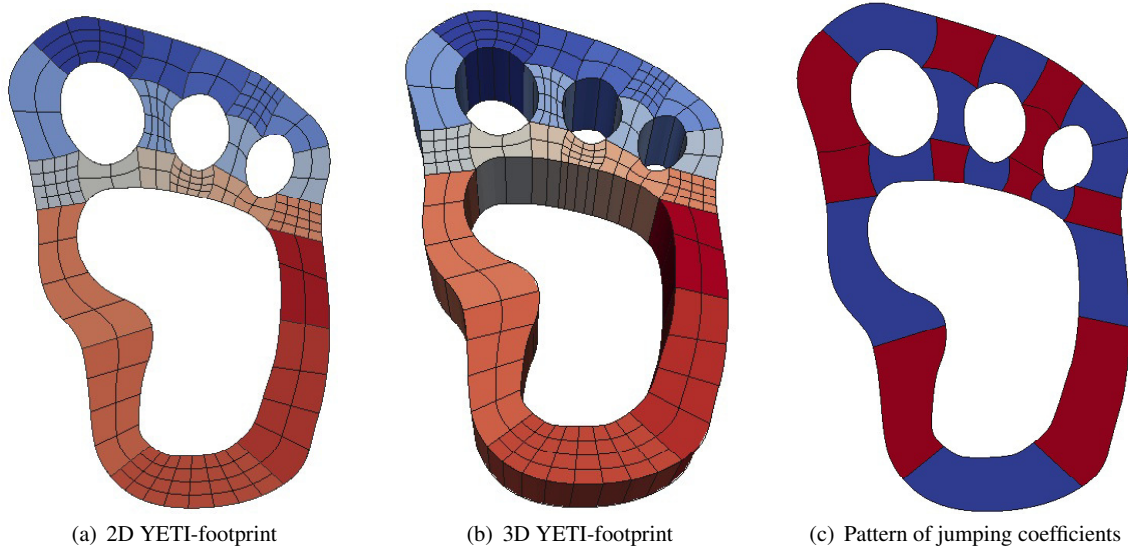

(a) 2D YETI-footprint   (b) 3D YETI-footprint   (c) Pattern of jumping coefficients

Fig. 1. Figure (a) and (b) show an illustration of the computational domain and its initial mesh in 2D and 3D, respectively. Figure (c) presents the pattern of the jumping diffusion coefficient.

4.1 The case of homogeneous diffusion coefficient

We first consider the case of homogeneous diffusion coefficient, i.e. $\alpha = 1$ on Ω. The 2D results are summarized in Table 1, whereas the 3D results are presented in Table 2. We observe, that the condition number of the preconditioned system grows logarithmically with respect to H/h. Moreover, the numerical results indicate a dependence of the condition number on the degree p, which will be investigated in more detail in Section 4.4.

4.2 The case of inhomogeneous diffusion coefficient

In this subsection, we investigate the case of patchwise constant diffusion coefficient, but with jumps across the patch interfaces. The diffusion coefficient takes values $\alpha^{(k)} \in \{10^{-4}, 10^4\}$,

---

[1] https://ricamsvn.ricam.oeaw.ac.at/trac/gismo/wiki/WikiStart

	Degree $p=2$						Degree $p=4$				
	ALG. A		coeff. scal.		stiff. scal.		ALG. A		coeff. scal.		stiff. scal.
#dofs	H/h	κ	It.	κ	It.	#dofs	H/h	κ	It.	κ	It.
1610	8	3.07	17	3.16	17	4706	8	4.56	20	4.75	20
4706	16	4.06	19	4.22	19	9370	16	5.64	22	5.89	22
15602	32	5.22	21	5.45	21	23402	32	7.01	23	7.35	23
56210	64	6.55	23	6.86	23	70282	64	8.56	24	9.00	24
212690	128	8.04	24	8.45	24	239306	128	10.3	26	10.8	26
	ALG. C						ALG. C				
1610	8	1.35	9	1.34	9	4706	8	1.78	11	1.79	12
4706	16	1.64	11	1.64	11	9370	16	2.11	13	2.11	13
15602	32	1.99	12	1.99	12	23402	32	2.54	15	2.53	15
56210	64	2.41	14	2.41	14	70282	64	3.03	16	3.00	16
212690	128	2.88	16	2.88	16	239306	128	3.57	17	3.54	18

Table 1. 2D example with $p=2$ (left) and $p=4$ (right), and homogeneous diffusion coefficient. Dependence of the condition number κ and the number It. of iterations on H/h for the preconditioned system with coefficient and stiffness scaling. Choice of primal variables: vertex evaluation (upper table), vertex evaluation and edge averages (lower table).

	Degree $p=2$						Degree $p=4$				
	ALG. A		coeff. scal.		stiff. scal.		ALG. A		coeff. scal.		stiff. scal.
#dofs	H/h	κ	It.	κ	It.	#dofs	H/h	κ	It.	κ	It.
2800	3	49.9	44	49.3	44	22204	3	203	77	199	79
9478	6	72.3	48	70.3	49	42922	6	248	82	240	83
42922	12	169	70	165	69	116110	12	506	104	488	104
244594	25	376	91	368	92	443926	25	*	*	*	*
	ALG. B						ALG. B				
2800	3	1.49	9	1.45	9	22204	3	23.3	33	21.5	34
9478	6	15.8	17	14.7	17	42922	6	26.7	34	24.7	34
42922	12	19.8	30	18.4	29	116110	12	31.3	42	29.1	41
244594	25	24	37	22.4	36	443926	25	*	*	*	*

Table 2. 3D example with $p=2$ (left) and $p=4$ (right), and homogeneous diffusion coefficient. Dependence of the condition number κ and the number It. of iterations on H/h for the preconditioned system with coefficient and stiffness scaling. Choice of primal variables: vertex evaluation (upper table), vertex evaluation and edge averages and face averages (lower table).

with a jumping pattern according to Figure 1 (c). The 2D results are summarized in Table 3, and the 3D results are presented in Table 4. First of all, one clearly sees the robustness with respect to jumping coefficients of the considered method and the quasi optimal dependence of the condition number on H/h. The dependence of the degree will again be studied in Section 4.4.

	Degree $p=2$						Degree $p=4$				
	ALG. A		coeff. scal.		stiff. scal.		ALG. A		coeff. scal.		stiff. scal.
#dofs	H/h	κ	It.	κ	It.	#dofs	H/h	κ	It.	κ	It.
1610	8	3.82	12	4.02	12	4706	8	5.72	14	6.16	14
4706	16	5.11	13	5.47	13	9370	16	7.08	14	7.7	15
15602	32	6.58	15	7.12	15	23402	32	8.77	15	9.64	17
56210	64	8.23	15	9	16	70282	64	10.7	18	11.8	18
212690	128	10.1	17	11.1	18	239306	128	12.8	18	14.2	18
	ALG. C						ALG. C				
1610	8	1.4	7	1.43	7	4706	8	1.85	8	1.94	8
4706	16	1.7	7	1.78	7	9370	16	2.17	8	2.32	8
15602	32	2.06	8	2.19	8	23402	32	2.58	9	2.81	9
56210	64	2.46	8	2.65	8	70282	64	3.05	9	3.36	9
212690	128	2.9	9	3.18	9	239306	128	3.55	10	3.97	10

Table 3. 2D example with $p=2$ (left) and $p=4$ (right), and jumping diffusion coefficient. Dependence of the condition number κ and the number It. of iterations on H/h for the preconditioned system with coefficient and stiffness scaling. Choice of primal variables: vertex evaluation (upper table), vertex evaluation and edge averages (lower table).

Degree $p = 2$					Degree $p = 4$					
ALG. A		coeff. scal.		stiff. scal.	ALG. A		coeff. scal.		stiff. scal.	
#dofs	H/h	κ	It.	κ	It.	#dofs	H/h	κ	It.	κ	It.
2800	3	50.3	28	57.9	25	22204	3	203	44	236	45
9478	6	72.2	29	83.4	29	42922	6	250	43	290	42
42922	12	176	43	203	42	116110	12	520	58	605	57
244594	25	400	52	463	58	443926	25	*	*	*	*
ALG. B					ALG. B					
2800	3	2.11	11	2.17	11	22204	3	17.7	15	20.7	15
9478	6	12.6	17	14.6	18	42922	6	20.5	17	23.9	17
42922	12	15.7	22	18.2	24	116110	12	24	19	28	21
244594	25	18.9	28	22	30	443926	25	*	*	*	*

Table 4. 3D example with $p = 2$ (left) and $p = 4$ (right), and jumping diffusion coefficient. Dependence of the condition number κ and the number It. of iterations on H/h for the preconditioned system with coefficient and stiffness scaling. Choice of primal variables: vertex evaluation (upper table), vertex evaluation and edge averages and face averages (lower table).

4.3 Dependence in $h^{(k)}/h^{(l)}$

In this subsection, we deal with dependence of the condition number on the ratio $h^{(k)}/h^{(l)}$ of mesh sizes corresponding to neighbouring patches, The initial domain is the same as given in Figure 1, but without the additional refinements in certain patches, i.e. $h^{(k)}/h^{(l)} = 1$. Then we consequently perform uniform refinement in the considered patches and obtain $h^{(k)}/h^{(l)} = 2^{-r}$, where r is the number of refinements. In the numerical tests, we only consider the cases $\alpha \in \{10^{-4}, 10^4\}$ and $p = 4$. The results for 2D and 3D are summarized in Table 5 and indicate that the condition number is independent of the ratio $h^{(k)}/h^{(l)}$ for 2D and 3D, as also predicted for FE in [15]. We note that the increasing condition number and number of iterations come along with the increased ratio H/h and comparing the numbers of this test with the corresponding ones from Table 4 we observe an agreement. Thus, it is noteworthy that, although in 2D the ratio H/h is increasing, the condition number stays constant.

dim = 2							dim = 3						
ALG. A			coeff. scal.		stiff. scal.		ALG. C			coeff. scal.		stiff. scal.	
#dofs	$h^{(k)}/h^{(l)}$	H/h	κ	It.	κ	It.	#dofs	$h^{(k)}/h^{(l)}$	H/h	κ	It.	κ	It.
1816	1	2	4.92	13	5.21	14	9362	1	1	14.8	21	14.9	21
2134	2	4	4.93	13	5.36	14	11902	2	3	17.7	23	22.1	24
2962	4	8	4.93	13	5.55	14	20426	4	6	29.2	24	37.5	27
5386	8	16	4.93	13	5.69	14	56626	8	12	52.2	26	67.2	27
13306	16	32	4.93	13	5.71	14	345268	16	25	*	*	*	*
41434	32	64	4.92	13	5.66	14	1758004	32	50	*	*	*	*
ALG. C							ALG. B						
1816	1	2	1.67	7	1.72	7	9362	1	1	14.8	15	16.6	15
2134	2	4	1.67	7	1.77	7	11902	2	3	19.5	15	28.7	17
2962	4	8	1.67	7	1.81	7	20426	4	6	29.2	16	37.5	18
5386	8	16	1.67	7	1.85	7	56626	8	12	52.2	17	67.2	18
13306	16	32	1.67	7	1.85	7	345268	16	25	*	*	*	*
41434	32	64	1.67	7	1.84	7	1758004	32	50	*	*	*	*

Table 5. 2D (left) and (3D)example with $p = 4$, and jumping diffusion coefficient. Dependence of the condition number κ and the number It. of iterations on the ratio $h^{(k)}/h^{(l)}$ for the preconditioned system with coefficient and stiffness scaling. Choice of primal variables: in 2D vertex evaluation (upper table), vertex evaluation and edge averages (lower table), in 3D vertex evaluation and edge averages (upper table), vertex evaluation, edge averages and face averages (lower table).

4.4   Dependence on p

In this subsection, we study the dependence of the condition number on the degree p of the B-Spline space. There are two ways to dealing with degree elevation. One method is to keep the smoothness of the space, i.e., the multiplicity of the knots is increased in each step. The other way keeps the multiplicity of the knots, while increasing the smoothness of the B-Spline. The first method retains the support of the B-Spline basis small, with the drawback of a larger number of dofs, while the second method does it vice versa, i.e., increasing the support of the B-Spline, while having a smaller number of dofs. The aim of this section is to investigate the effect of the two different elevation techniques on the condition number.

We choose the computational domain as the 2D and 3D YETI-footprint presented in Figure 1 and the diffusion coefficient is chosen to be globally constant. The results are summarized in Table 6 and in Table 7 for the 2D and 3D domain, respectively. The numerical results indicate a at most linear dependence of the condition number of the preconditioned system on the B-Spline degree p. When considering the 2D domain, the dependence on the degree seems to be also logarithmic, see Figure 2. One observes a significant increase of the condition number, when increasing the degree from 2 to 3 in 3D as illustrated in Figure 2 (b).

Increasing the multiplicity						Increasing the smoothness					
ALG. C		coeff. scal		stiff. scal.		ALG. C		coeff. scal		stiff. scal.	
#dofs	degree	κ	It.	κ	It.	#dofs	degree	κ	It.	κ	It.
1610	2	1.36	9	1.37	9	1610	2	1.36	9	1.37	9
4706	3	1.68	11	1.69	11	2006	3	1.55	10	1.57	11
9370	4	1.95	12	1.96	13	2444	4	1.74	11	1.77	12
15602	5	2.19	13	2.2	14	2924	5	1.88	12	1.93	12
23402	6	2.4	15	2.4	15	3446	6	2.03	13	2.09	13
32770	7	2.59	15	2.59	16	4010	7	2.14	14	2.22	14
43706	8	2.77	16	2.76	16	4616	8	2.27	14	2.36	14
56210	9	2.93	17	2.92	17	5264	9	2.36	15	2.47	15
70282	10	3.08	17	3.06	17	5954	10	2.48	15	2.59	15

Table 6. 2D example with fixed initial mesh and homogeneous diffusion coefficient. Dependence of the condition number κ and the number It. of iterations on H/h for the preconditioned system with coefficient and stiffness scaling. Choice of primal variables: vertex evaluation and edge averages.

Increasing the multiplicity						Increasing the smoothness					
ALG. B		coeff. scal		stiff. scal.		ALG. B		coeff. scal		stiff. scal.	
#dofs	degree	κ	It.	κ	It.	#dofs	degree	κ	It.	κ	It.
2800	2	1.49	9	1.45	9	2800	2	1.49	9	1.45	9
9478	3	17.9	20	16.6	20	4864	3	17.1	18	16.4	18
22204	4	23.3	33	21.5	34	7714	4	21.7	32	21	33
42922	5	29.0	39	26.9	39	11476	5	27.0	45	26.3	46
73576	6	34.4	51	32.0	49	16276	6	31.9	47	31.3	47
116110	7	40.2	54	37.6	55	22240	7	37.5	51	36.9	50

Table 7. 3D example with fixed initial mesh and homogeneous diffusion coefficient. Dependence of the condition number κ and the number It. of iterations on H/h for the preconditioned system with coefficient and stiffness scaling. Choice of primal variables: vertex evaluation, edge averages and face averages.
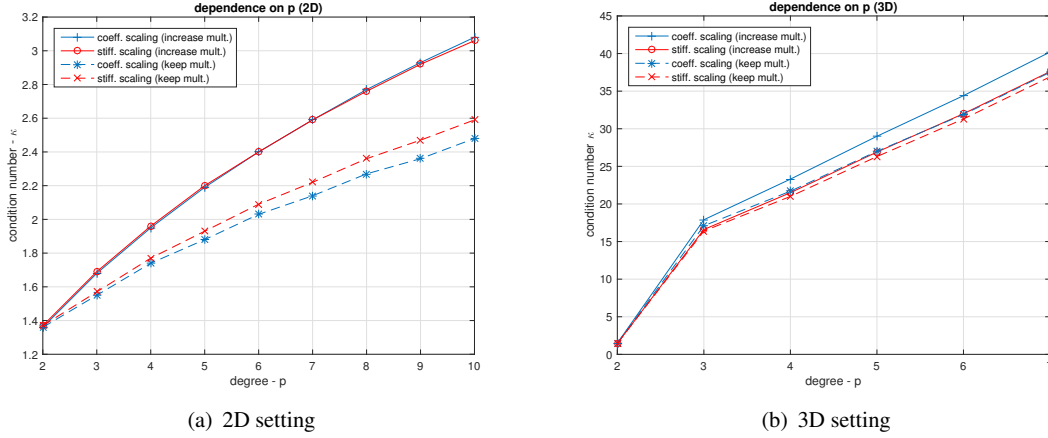
(a) 2D setting   (b) 3D setting

Fig. 2. Dependence of the condition number on the B-Spline degree p for the 2D and 3D domain. We compare the influence of the considered scaling strategy and method for increasing the degree.

4.5   Performance and OpenMP

As already explained in the beginning of this section, we use the open source C++ library G+SMO for materialising the code. The LU-factorizations for the local solvers were performed by means of the PARDISO 5.0.0 Solver Project, see [27]. In Table 8, we investigate the runtime of a serial implementation and compare it with the timings of the cG-IETI-DP algorithm as presented in [21]. In order to compare the timings for the continuous IETI-DP and discontinuous IETI-DP method, we use the setting as described in [21]. More precisely, the computational domain is the 2D example from Figure 1, but with fully matching patches, i.e. no additional refinements on selected patches. We observe from Table 8 that the dG-IETI-DP method shows a very similar performance. The increased number of primal variables and the extended version of the stiffness matrix K_e leads to a slightly larger runtime. As already mentioned, we use the PARDISO Solver Project instead of the SparseLU factorization of the open source library "Eigen"[2] as in Subsection 5.4 in [21]. This change led to a significant speed-up in the computation time for the LU-factorization. Small deviations in the timings compared to Table 6 in [21] might be due to some changes regarding the parallel implementation.

The IETI-DP method is well suited for a parallel implementation, since most of the computations are independent of each other. Only the assembly and application of the Schur complement $S_{\Pi\Pi}$ corresponding to the primal variables requires communication with neighbouring patches. Although the structure of the algorithm perfectly suits the framework of distributed memory models, hence, using MPI, we first implemented a version using OpenMP. We want to mention that the numerical examples in Section 4.1 and Section 4.2 are performed by means of the parallel implementation. The major problem for obtaining a scaleable method is the unequally distributed workload. This arises from the fact that in IgA the partition of the domain is mostly based on geometric aspect, where as in FE the mesh is partitioned in such a way, that each patch has a similar number of dofs. Especially, the cases considered in Section 4.1 and Section 4.2, where we have non-matching meshes, lead to very unequally distributed workloads. To summarize, in order to achieve a scalable IETI type solver one has to spend some time in finding an equally distributed workload for each thread, e.g. performing further subdivisions of certain patches and optimal assignment of patches to threads.

---

[2] `http://eigen.tuxfamily.org/index.php?title=Main_Page`

	Wall-clock time		relative time in %	
	cG-IETI-DP	dG-IETI-DP	cG-IETI-DP	dG- IETI-DP
Preparing the bookkeeping	0.012 s	0.02 s	0.06	0.11
Assembling all patch local $\boldsymbol{K}^{(k)}$	6.4 s	6.8 s	35.56	35.79
Partitioning w.r.t. B and I	0.085 s	0.12 s	0.48	0.63
Assembling C	0.017 s	0.034 s	0.09	0.18
Calculating LU -fact. of $\boldsymbol{K}_{II}^{(k)}$	2.5 s	2.5 s	13.89	13.16
Calculating LU -fact. of $\begin{bmatrix} K^{(k)} & C^{(k)^T} \\ C^{(k)} & 0 \end{bmatrix}$	3.9 s	4 s	21.67	21.05
Assembling and LU-fact of $\boldsymbol{S}_{\Pi\Pi}$	0.78 s	1.1 s	4.33	5.79
Assemble rhs.	0.13 s	0.13 s	0.72	0.68
Total assembling	14 s	15 s	77.78	78.95
One PCG iteration	0.34 s	0.36 s	-	-
Solving the system	3.4 s	3.6 s	18.89	18.95
Calculating the solution $\boldsymbol{u}$	0.33 s	0.33 s	1.83	1.74
Total spent time	18 s	19 s	100.00	100.00

Table 8. Serial computation times of the 2D example with coefficient scaling and Algorithm C. The discrete problem consists of 121824 total degrees of freedom, 1692 Lagrange multipliers, and on each patch approximate 4900 local degrees of freedom according to the setting in [21], Section 5. Column 2 and 3 present the absolute spent time, whereas column 4 and 5 present the relative one for the cG-IETI-DP and dG-IETI-DP method.

5   Conclusions and outlook

In this paper, we investigated an adaption of the IETI-DP method to dG-IgA equations, i.e. we used dG techniques to couple non-matching meshes across patch interfaces. The numerical examples in Section 4 indicate the same quasi-optimal behaviour of the condition number of the dG-IETI-DP operator with respect to H/h, and show robustness with respect to jumps in the diffusion coefficient. Additionally, the condition number in 2D and 3D seems to be independent of the ratio $h^{(k)}/h^{(l)}$. Moreover, we examined the dependence of the condition number on the B-Spline degree. We found that in 2D the dependence is quite weak, while in 3D one observes a more significant increase. As illustrated in Figure 2, the dependence seems to be linear or even logarithmic in 2D, while it is clearly linear in 3D. Finally, we investigated the performance of the dG-IETI-DP in comparison with the cG-IETI-DP method, which turns out to be very similar. The theoretical analysis, that was done for cG-IETI-DP in [21], is more technical for dG-IETI-DP, cf. [15] for dG-FETI-DP. The dG-IETI-DP can be generalized to dG-IgA schemes on segmentations with non-matching interfaces (segmentation crimes) studied in [22] and [23].

References

1. D. N. Arnold.   An interior penalty finite element method with discontinuous elements.   *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.
2. 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, 2002.
3. Y. Bazilevs, L. Beirão da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. *Math. Models Methods Appl. Sci.*, 16(7):1031–1090, 2006.
4. L. Beirão da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Mathematical analysis of variational isogeometric methods. *Acta Numerica*, 23:157–287, 2014.
5. L. Beirão da Veiga, D. Cho, L. F. Pavarino, and S. Scacchi. Overlapping Schwarz methods for isogeometric analysis. *SIAM J. Numer. Anal.*, 50(3):1394–1416, 2012.
6. L. Beirão Da Veiga, D. Cho, L. F. Pavarino, and S. Scacchi. BDDC preconditioners for isogeometric analysis. *Math. Models Methods Appl. Sci.*, 23(6):1099–1142, 2013.
7. L. Beirão da Veiga, D. Cho, L. F. Pavarino, and S. Scacchi. Isogeometric Schwarz preconditioners for linear elasticity systems. *Comput. Methods Appl. Mech. Eng.*, 253:439–454, 2013.

8. L. Beirão Da Veiga, L. F. Pavarino, S. Scacchi, O. B. Widlund, and S. Zampini. Isogeometric BDDC preconditioners with deluxe scaling. *SIAM J. Sci. Comput.*, 36(3):a1118–a1139, 2014.

9. M. Bercovier and I. Soloveichik. Overlapping non matching meshes domain decomposition method in isogeometric analysis. *arXiv preprint arXiv:1502.03756*, 2015.

10. J. A. Cotrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis, Toward Integration of CAD and FEA*. John Wiley and Sons, 2009.

11. D. A. Di Pietro and A. Ern. *Mathematical aspects of discontinuous Galerkin methods.* Berlin: Springer, 2012.

12. C. R. Dohrmann and O. B. Widlund. Some recent tools and a BDDC algorithm for 3D problems in H(curl). In *Domain Decomposition Methods in Science and Engineering XX*, pages 15–25. Springer, 2013.

13. M. Dryja. On discontinuous Galerkin methods for elliptic problems with discontinuous coefficients. *Comput. Methods Appl. Math.*, 3(1):76–85, 2003.

14. M. Dryja, J. Galvis, and M. Sarkis. BDDC methods for discontinuous Galerkin discretization of elliptic problems. *J. Complexity*, 23(4-6):715–739, 2007.

15. M. Dryja, J. Galvis, and M. Sarkis. A FETI-DP preconditioner for a composite finite element and discontinuous Galerkin method. *SIAM J. Numer. Anal.*, 51(1):400–422, 2013.

16. M. Dryja and M. Sarkis. 3-d feti-dp preconditioners for composite finite element-discontinuous galerkin methods. In *Domain Decomposition Methods in Science and Engineering XXI*, pages 127–140. Springer, 2014.

17. J. A. Evans and T. J. R. Hughes. Explicit trace inequalities for isogeometric analysis and parametric hexahedral finite elements. *Numer. Math.*, 123(2):259–290, 2013.

18. C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: the truncated basis for hierarchical splines. *Comput. Aided Geom. Design*, 29, 2012.

19. C. Giannelli, B. Jüttler, and H. Speleers. Strongly stable bases for adaptively refined multilevel spline spaces. *Advances in Computational Mathematics*, 40:459–490, 2014.

20. C. Hesch and P. Betsch. Isogeometric analysis and domain decomposition methods. *Comput. Methods Appl. Mech. Eng.*, 213-216:104–112, 2012.

21. C. Hofer and U. Langer. Dual-primal isogeometric tearing and interconecting solvers for large-scale systems of multipatch continuous Galerkin IgA equations. NFN-Technical Report No. 39, http://www.gs.jku.at/pubs/NFNreport39.pdf, Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, 2015.

22. C. Hofer, U. Langer, and I. Toulopoulos. Discontinuous Galerkin Isogeometric Analysis of Elliptic Diffusion Problems on Segmentations with Gaps. RICAM-Technical and NFN-Technical Report No. 38, http://www.gs.jku.at/pubs/NFNreport38.pdf, Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, 2015. available also at: http://arxiv.org/abs/1511.05715.

23. C. Hofer and I. Toulopoulos. Discontinuous Galerkin Isogeometric Analysis of Elliptic Problems on Segmentations with Non-Matching Interfaces. RICAM-Technical Report No. 46, https://www.ricam.oeaw.ac.at/publications/reports/15/rep15-46.pdf, Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, 2015.

24. T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Engrg.*, 194:4135–4195, 2005.

25. B. Juettler, U. Langer, A. Mantzaflaris, S. E. Moore, and W. Zulehner. Geometry + simulation modules: Implementing isogeometric analysis. *Proc. Appl. Math. Mech.*, 14(1):961–962, 2014. Special Issue: 85th Annual Meeting of the Int. Assoc. of Appl. Math. and Mech. (GAMM), Erlangen 2014.

26. S. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. IETI–isogeometric tearing and interconnecting. *Computer Methods in Applied Mechanics and Engineering*, 247:201–215, 2012.

27. A. Kuzmin, M. Luisier, and O. Schenk. Fast methods for computing selected elements of the greens function in massively parallel nanoelectronic device simulations. In F. Wolf, B. Mohr, and D. Mey, editors, *Euro-Par 2013 Parallel Processing*, volume 8097 of *Lecture Notes in Computer Science*, pages 533–544. Springer Berlin Heidelberg, 2013.

28. U. Langer, A. Mantzaflaris, S. E. Moore, and I. Toulopoulos. Multipatch discontinuous Galerkin isogeometric analysis. In B. Jüttler and B. Simeon, editors, *Isogeometric Analysis and Applications IGAA 2014*, Lecture Notes in Computer Science, Heidelberg, 2015. Springer. to appear, also available as Technical Report no. 18 at http://www.gs.jku.at and at http://arxiv.org/abs/1411.2478.

29. U. Langer and I. Toulopoulos. Analysis of multipatch discontinuous Galerkin IgA approximations to elliptic boundary value problems. RICAM Reports 2014-08, Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Linz, 2014. also available at ArXiv 1408.0182.

30. A. Mantzaflaris, C. Hofer, et al. G+smo (geometry plus simulation modules) v0.8.1. http://gs.jku.at/gismo, 2015.

31. A. Mantzaflaris, B. Jüttler, B. Khoromskij, and U. Langer. Low rank tensor isogeometric analysis. Technical Report 2015-xx, Austrian Academy of Sciences, Johann Radon Institute for Computational and Applied Mathematics, 2015.

32. A. Mantzaflaris, B. Jüttler, B. Khoromskij, and U. Langer. Matrix generation in isogeometric analysis by low rank tensor approximation. In M. Floater, T. Lyche, M.-L. M. K. Moerken, and L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2015. accepted for publication, also available as Technical Report no. 19 at www.gs.jku.at.

33. J. Nitsche. über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36:9–15, 1971. (in German).

34. C. Pechstein. *Finite and boundary element tearing and interconnecting solvers for multiscale problems.* Berlin: Springer, 2013.

35. B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations. Theory and implementation.* Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2008.
36. A. Toselli and O. B. Widlund. *Domain decomposition methods – algorithms and theory.* Berlin: Springer, 2005.