**Johann Radon Institute for
Computational and Applied Mathematics
Austrian Academy of Sciences (ÖAW)**

Dual-Primal Isogeometric Tearing and Interconnecting Solvers for large-scale systems of multipatch continuous Galerkin IgA equations

C. Hofer, U. Langer

RICAM-Report 2015-44

Dual-Primal Isogeometric Tearing and Interconnecting Solvers for large-scale systems of multipatch continuous Galerkin 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. The dual-primal isogeometric tearing and interconnecting (IETI-DP) method is the adaption of the dual-primal finite element tearing and interconnecting (FETI-DP) method to isogeometric analysis of scalar elliptic boundary value problems like, e.g., diffusion problems with heterogeneous diffusion coefficients. The purpose of this paper is to extent the already existing results on condition number estimates to multi-patch domains, which consist of different geometrical mappings for each patch. Another purpose is to prove a polylogarithmic condition number bound for the preconditioned system with stiffness scaling in case of C^0 smoothness across patch interfaces. Numerical experiments validate the presented theory.

1 Introduction

Isogeometric Analysis (IgA) is a new methodology for solving partial differential equations (PDEs) numerically. IgA was introduced by Hughes, Cottrell and Bazilevs in [1], and has become a very active field of research, see also [2] for the first results on the numerical analysis of IgA, and the monograph [3] for a comprehensive presentation of the IgA. The main idea is to use the basis functions, which are used for the representation of the geometry in computer aided design (CAD) models anyway, also for the approximation of the solution of the PDE or the system of PDEs describing the physical phenomenon which we are going

to simulate. The typical choice for such basis functions are B-Splines or Non-Uniform Rational Basis Spline (NURBS). One advantage of the IgA over the more traditional finite element method (FEM) is certainly the fact that there is no need for decomposing the computational domain into finite elements. Hence, one gets rid of this geometrical error source, at least, in the class of computational domains that are produced by a CAD system. Moreover, it is much easier to build up $C^l, l \geq 1$, conforming basis functions in IgA than in the finite element (FE) case. The major drawback is the fact that the basis functions are not nodal and have a larger support. However, it is still possible to associate basis functions to the interior, the boundary and the vertices of the domain, which is crucial for the dual-primal isogeometric tearing and interconnecting (IETI-DP) method, which was introduced in [4]. The IETI-DP method is an extension of the dual-primal finite element tearing and interconnecting method (FETI-DP) to IgA. A comprehensive theoretical analysis of the FETI-DP method can be found in, e.g., [5], [6] or [7]. We here also mention the recent development of other IgA domain decomposition (DD) techniques. In particular, we want to mention isogeometric overlapping Schwarz methods, where we refer to [8], [9], [10], and isogeometric mortaring discretizations, see [11]. The analysis of Balancing Domain Decomposition by Constraints (BDDC) preconditioner, which has been done for IgA matrices in [12], also applies to the IETI-DP method due to the same spectrum (with the exception of at most two eigenvalues), see [13]. Based on the FE work in [14], a recent improvement for the IgA BDDC preconditioner with a more advanced scaling technique, the so called *deluxe scaling*, can be found in [15].

The goal of this paper is to extend the condition number estimates for BDDC preconditioners, presented in [12], to multipatch domains composed of non-overlapping patches which are images of the parameter domain by several different geometrical mappings. Moreover, we present the derivation of an improved bound for the condition number for the so-called *stiffness scaling* in the simplified case of C^0 smoothness across patch interfaces. It turns out that, in our case, the stiffness scaling provides the same quasi optimal polylogarithmic bound in terms of H/h as for the coefficient scaling.

In the present paper, we consider the following model elliptic boundary problem in a bounded Lipschitz domain $\Omega \subset \mathbb{R}^d, d \in \{2, 3\}$: find

$u : \overline{\Omega} \to \mathbb{R}$ such that

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

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 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

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

respectively.

The rest of the paper is organized as follows. In Section 2, we recall the basic definitions and properties of B-Splines as well as the main principles of IgA. The IETI-DP and the corresponding BDDC methods are explained in Section 3. In particular, the new results on the condition number estimate, as mentioned above, are presented in Subsection 3.4. Section 4 is devoted to the implementation of the IETI-DP method. The numerical examples confirming the theory are presented in Section 5. Finally, in Section 6, we draw some conclusions and discuss further issues concerning generalizations to multipatch discontinuous Galerkin IgA schemes as constructed and analysed in [16] and [17].

2 Some preliminaries on Isogeometric analysis

B-Splines and NURBS play an important role in computer aided design and computer graphics. Here we will use these splines for building our trial and test spaces for the Galerkin approximations to (2), as proposed in [3]. This section provides the definition of B-Splines in one dimension and in higher dimensions via a tensor product structure. We will give an overview of isogeometric discretization and summarize the approximation properties of these B-Splines and NURBS.

2.1 Univariate B-Splines

Definition 1. *Let $\xi_1 \leq \ldots \leq \xi_m$ be a finite, real-valued, monotonically increasing sequence of real numbers. The set $\{\xi_1, \ldots, \xi_m\}$ is called* knot vector. *An entry $\xi_i, i \in \{1, \ldots, m\}$ is called* knot *and is called an* interior knot *if $(\xi_1 < \xi_i) \wedge (\xi_i < \xi_m)$. If r knots have the same value, we say that the knot has multiplicity r, i.e., $r = |\{j \in \{1, \ldots, m\} : \xi_j = \xi_i\}|$ is the cardinal number of the set $\{j \in \{1, \ldots, m\} : \xi_j = \xi_i\}$. The interval between two knots is called* knot span. *A knot span is called* empty *if $\xi_i = \xi_{i+1}$ and is called interior if $\xi_1 < \xi_{i+1} \wedge \xi_i < \xi_m$. The knots provides a partitioning of the parameter domain into elements. If the knots are equally spaced in the domain, we call it* uniform, *otherwise* non-uniform.

Based on a knot vector, we can define the B-Spline functions recursively.

Definition 2. *Let $p \in \mathbb{N}$ and Ξ be a knot vector with multiplicity of any interior knot of at most p. Then by the* Cox-de Boor *recursion formula we can define the $M = m - p - 1$ univariate B-Spline basis functions on $[\xi_1, \xi_m]$ as follows:*

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

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

where $i = 1, \ldots, M$. If in (4) appears a $0/0$ we define it as 0. The number p is then called degree *of the B-Spline.*

Definition 3. *Let Ξ be a knot vector. We say that the knot vector is* open *if the multiplicity of the first knot and the last knot are $p + 1$ whereas the multiplicity of the other knots is p at most.*

B-Splines defined on open knot vectors are interpolatory at the beginning and the end of the parameter interval, while all other basis function are zero there. Hence it is possible to identify basis functions corresponding to the interior and the boundary. Additionally, the number of interior knot spans ν is given by $\nu = m - 1 - 2p$. Without loss of generality, we can restrict ourselves to a certain class of knot vectors by means of a suitable scaling.

Assumption 1 *We only consider knot vectors being a partition of $[0, 1]$, i.e. $\xi_1 = 0$ and $\xi_m = 1$.*

Since later on, we will only be interested in C^0 continuity across the interfaces, we restrict our analysis to open knot vectors.

Assumption 2 *We consider all knot vectors used as open knot vectors.*

At the end of this section, let us summarize some important properties of B-Splines:

1. The B-Splines basis functions $N_{i,p}$ form a partition of unity, i.e.

$$\sum_{i=1}^{M} N_{i,p}(\xi) \equiv 1$$

   for all $p = 0, 1, \ldots$.

2. The B-Spline basis functions are non negative, i.e. $\forall \xi \in [\xi_1, \xi_m] \, \forall i \in \{1, \ldots, M\} : N_{i,p}(\xi) \geq 0$.

3. The support $N_{i,p}$ is local and it holds

$$\text{supp} N_{i,p} \subseteq (\xi_i, \xi_{i+p+1}), \quad \forall i \in \{1, \ldots, M\}.$$

4. Only $p + 1$ basis functions are non zero on a knotspan (ξ_i, ξ_{i+1}) and its indices are $i - p, \ldots, i$, i.e.

$$N_{j,p}|_{(\xi_i, \xi_{i+1})} \neq 0 \Leftrightarrow j \in \{i - p, \ldots, i\},$$

   where $i \in \{1, \ldots, m - 1\}$.

5. On each knotspan the B-Spline basis functions are piecewise polynomials of degree p and, without multiple knots in the interior, C^{p-1} continuous. At a knot with multiplicity r, it has C^{p-r} continuity. Hence, the continuity is reduced in the presence of multiple knots.

6. If a knot ξ_l has the multiplicity $r = p$, then there is one basis function $N_{i,p}$, such $N_{i,p}(\xi_l) = 1$ and all other basis functions have zero value there, i.e., the basis is interpolatory at ξ_l.

2.2 Tensor product B-Splines

In order to define B-Splines in higher dimensions, we make use of the tensor product.

Definition 4. *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 $N^\iota_{i^\iota,p}(\xi^\iota)$. Then the d-dimensional tensor product B-Spline (TB-Spline) is defined by*

$$N_{(i^1, \ldots, i^d),(p^1, \ldots, p^d)}(\xi) = \prod_{\iota=1}^{d} N^\iota_{i^\iota,p^\iota}(\xi^\iota). \tag{5}$$

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

$$\mathcal{I} := \{(i^1, \ldots, i^d) : i^\iota \in \{1, \ldots, M_\iota\} \, \forall \iota \in \{1, \ldots, d\}\},$$

where M_ι are the number of B-Spline basis function for dimension ι. Due to the tensor product structure, the TB-Splines provide the same properties as the univariate B-Splines.

In case of TB-Spline, we will call a non-empty knotspan $\hat{Q}_i = (\xi_i, \xi_{i+1}), i \in \mathcal{I}_{\hat{Q}}$ also *cell*, where $\mathcal{I}_{\hat{Q}} = \{i \in \mathcal{I} | i^\iota \neq M_\iota\}$ and (ξ_i, ξ_{i+1}) is defined as

$$(\xi_i, \xi_{i+1}) := (\xi^1_{i^1}, \xi^1_{i^1+1}) \times \ldots \times (\xi^d_{i^d}, \xi^d_{i^d+1}).$$

The mesh created by these cells is denoted by $\hat{Q}_h$, i.e.

$$\hat{Q}_h := \{(\xi_i, \xi_{i+1}) | i \in \mathcal{I}_{\hat{Q}}\}.$$

2.3   B-Spline geometries and geometrical mapping

The B-Splines are used to represent a d-dimensional geometry in $\mathbb{R}^g$, where $d \leq g$. In the following, we will restrict ourselves to the case $d = g \in \{2, 3\}$.

Definition 5. *Let* $\{N_{i,p}\}_{i \in \mathcal{I}}$ *be a family of tensor B-Spline basis functions. Given* control points $P_i \in \mathbb{R}^g$, $i \in \mathcal{I}$, *the* B-Spline surface/volume *is defined by*

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

We call G *the* geometrical mapping, *the domain* $(0,1)^d$ *of* G *parameter domain* *and the image* $\Omega := G((0,1)^d) \subset \mathbb{R}^g$ *physical domain. The geometrical mapping is called* regular *if* $\det \nabla G(\xi) \neq 0$, $\forall \xi \in [0,1]^d$.

The knot vector Ξ provides a partition of the parameter domain into cells, and, by means of the geometrical mapping, we receive a partition of the physical space in cells Q_i as well, where

$$Q_i = G(\hat{Q}_i), \hat{Q}_i \in \hat{\mathcal{Q}}_h.$$

If we collect all these cells, then we get a mesh $\mathcal{Q}_h$ for the physical domain

$$\mathcal{Q}_h := \{Q = G(\hat{Q}) | \hat{Q} \in \hat{\mathcal{Q}}_h\}.$$

Definition 6. *A family of meshes* $\{\mathcal{Q}_h\}_{h \in H}$ *is called* quasi uniform, *if there exists a constant* $\theta \geq 1$ *for all* $\mathcal{Q}_h \in \{\mathcal{Q}_h\}_{h \in H}$, *such that* $\theta^{-1} \leq diam(Q)/diam(Q') \leq \theta$ *for all* $Q, Q' \in \mathcal{Q}_h$.

Assumption 3 *All considered meshes are quasi uniform.*

2.3.1   Multi-patch geometries In many practical applications, it is not possible to describe the physical computational domain Ω just with one geometrical mapping G. Therefore, we represent 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 $\overline{\Omega} = \bigcup_{k=1}^{N} \overline{\Omega}^{(k)}$. Clearly, each patch has a mesh $\mathcal{Q}_h^{(k)}$ in the physical domain and a mesh $\hat{\mathcal{Q}}_h^{(k)}$ in the parameter domain, consisting of cells $Q^{(k)}$ and $\hat{Q}^{(k)}$.

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 is denoted by $\partial\Omega$. Note that the interface Γ is sometimes called *skeleton*.

2.4   Isogeometric discretization

The key point in isogeometric analysis is the use of the same functions for representing the geometry as well as basis functions for the solution space. This motives the definition of the basis functions in the physical domain via the push-forward of the basis functions in the parameter domain, i.e.,

$$\check{N}_{i,p} := N_{i,p} \circ G^{-1}.$$

Thus, we define our discrete function space V_h by

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

The function u_h from the IgA space V_h can therefore be represented in the form

$$u_h(x) = \sum_{i\in\mathcal{I}} u_i \check{N}_{i,p}(x).$$

Hence, each function $u_h(x)$ 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.

Sometimes we also need the space S_h of spline functions in the parameter domain, i.e.,

$$S_h = \text{span}\{N_{i,p}\}_{i \in \mathcal{I}} \subset H^1((0,1)^d). \tag{7}$$

Similar to the previous subsection, we define the discrete function spaces for each patch of a multipatch by

$$V_h^{(k)} = \text{span}\{\check{N}_{i,p}^{(k)}\}_{i \in \mathcal{I}^{(k)}} \subset H^1(\Omega^{(k)}) \tag{8}$$

and functions therein by

$$u_h(x) = \sum_{i \in \mathcal{I}^{(k)}} c_i^{(k)} \check{N}_{i,p}^{(k)}(x),$$

where

$$\check{N}_{i,p}^{(k)} := N_{i,p}^{(k)} \circ G^{(k)^{-1}}$$

The discrete function spaces for the whole multipatch domain is then given by

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

The space of spline functions in the parameter domain is analogously defined as $S_h^{(k)}$.

Based on the work in [12], we can find an important splitting of the space V_h. Since we are using open knot vectors, we can identify basis function on the interface Γ and in the interior of each patch. The set of all indices of basis function having a support on the interface is denoted by $\mathcal{I}_B$ and the corresponding space is defined via

$$V_{\Gamma,h} := \text{span}\{\check{N}_{i,p}|\ i \in \mathcal{I}_B\} \subset H^1(\Omega). \tag{9}$$

For a function $u \in V_{\Gamma,h}$, we define its restriction to a single patch $\Omega^{(k)}$ by $u^{(k)} \in V_{\Gamma,h}^{(k)}$, where $V_{\Gamma,h}^{(k)} := \{u_{|\Omega^{(k)}}|u \in V_{\Gamma,h}\}$. Similarly, we define the space of all functions living in the interior of $\Omega^{(k)}$ by

$$V_{I,h}^{(k)} := V_h^{(k)} \cap H_0^1(\Omega^{(k)}). \tag{10}$$

It is easy to see that the space $V_{I,h}^{(k)}$ has also the following representation

$$V_{I,h}^{(k)} = \mathrm{span}\{\check{N}_{i,p}^{(k)}|\ \mathrm{supp}\{\check{N}_{i,p}^{(k)}\} \subset \Omega^{(k)}\}. \tag{11}$$

This leads to the decomposition

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

where $\mathcal{H}$ is the *discrete NURBS harmonic extension* defined by

$$\mathcal{H} : V_{\Gamma,h} \rightarrow V_h :$$
$$\begin{cases} \text{Find } \mathcal{H}v_B \in V_h : \\ \quad a(\mathcal{H}v_B, v^{(k)}) = 0 \qquad \forall v^{(k)} \in V_{I,h}^{(k)},\ 1 \le k \le N, \\ \quad \mathcal{H}v_{B|\partial\Omega^{(k)}} = v_{B|\partial\Omega^{(k)}} \quad 1 \le k \le N. \end{cases} \tag{12}$$

See [12] and [18] for a more sophisticated discussion.

2.4.1 Continuous Galerkin IgA Schemes

We now look for the Galerkin approximate u_h from the finite dimensional subspace $V_{D,h}$ of V_D, where $V_{D,h}$ is the set of all functions from V_h which vanish on the Dirichlet boundary Γ_D. The Galerkin IgA scheme reads as follows: find $u_h \in V_{D,h}$ such that

$$a(u_h, v_h) = \langle F, v_h \rangle \quad \forall v_h \in V_{D,h}. \tag{13}$$

A basis of the space $V_{D,h}$ are the B-Spline functions $\{\check{N}_{i,p}\}_{i \in \mathcal{I}_0}$. However, we have to exclude those basis functions which have a contribution to the value on the Dirichlet boundary, obtaining the new index set $\mathcal{I}_0$. Since we are using open knot vectors, we can identify those basis functions. By choosing such a basis and introducing a global ordering $\mathcal{I}$ of the basis function across all patches, we can rewrite the Galerkin IgA scheme (13) as a linear system of IgA equations of the form

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

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 given by the IgA isomorphism. In order to keep the notation simple, we will reuse the symbol $\mathcal{I}$ for the set $\mathcal{I}_0$ in the following.

2.4.2   Schur complement system   Introducing the bilinear form

$$s : V_{\Gamma,h} \times V_{\Gamma,h} \to \mathbb{R}$$
$$s(w_B, v_B) = a(\mathcal{H}w_B, \mathcal{H}v_B),$$

one can show that the interface component u_B of the solution to the IgA scheme (13) satisfies the variational identity

$$s(u_B, v_B) = \langle g, v_B \rangle \quad \forall v_B \in V_{\Gamma,h}, \tag{15}$$

where $g \in V_{\Gamma,h}^*$ is a suitable functional. By choosing the B-Spline basis for $V_{\Gamma,h}$, the variational identity (15) is equivalent to the linear system

$$\boldsymbol{S}\boldsymbol{u}_B = \boldsymbol{g}.$$

The matrix $\boldsymbol{S}$ is the Schur complement matrix of $\boldsymbol{K}$ with respect to the interface dofs. Suppose, we reorder the entries of the stiffness matrix $\boldsymbol{K}$ and the load vector $\boldsymbol{f}$, such that the dofs corresponding to the interface come first, i.e.,

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{BB} & \boldsymbol{K}_{BI} \\ \boldsymbol{K}_{IB} & \boldsymbol{K}_{II} \end{bmatrix} \text{ and } \quad \boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_B \\ \boldsymbol{f}_I \end{bmatrix},$$

then it can be shown that $\boldsymbol{S}$ and $\boldsymbol{g}$ are given by

$$\boldsymbol{S} = \boldsymbol{K}_{BB} - \boldsymbol{K}_{BI}(\boldsymbol{K}_{II})^{-1}\boldsymbol{K}_{IB},$$
$$\boldsymbol{g} = \boldsymbol{f}_B - \boldsymbol{K}_{BI}(\boldsymbol{K}_{II})^{-1}\boldsymbol{f}_I.$$

Once $\boldsymbol{u}_B$ is calculated, we obtain $\boldsymbol{u}_I$ as the solution of the system

$$\boldsymbol{K}_{II}\boldsymbol{u}_I = \boldsymbol{f}_I - \boldsymbol{K}_{BI}\boldsymbol{u}_B.$$

Instead of the Schur complement matrix $\boldsymbol{S}$ we will mostly use its operator representation:

$$S : V_{\Gamma,h} \to V_{\Gamma,h}^*,$$
$$\langle Sv, w \rangle = (\boldsymbol{S}\boldsymbol{v}, \boldsymbol{w}).$$

2.4.3 Approximation Properties This section recaps some important properties of the approximation power of B-Splines from [2]. First of all, we state a result about the relation of the H^m norms between the function in the physical and the parameter domain, summarized in Proposition 1 and Corrolary 1, which are proved in [2], see Lemma 3.5.

Proposition 1. *Let m be a non-negative integer, $\hat{Q} \in \hat{\mathcal{Q}}_h$ and $Q = G(\hat{Q})$. Then the equivalence inequalities*

$$|v|_{H^m(\hat{Q})} \leq C_{shape} \left\| \det \nabla G^{-1} \right\|_{L^\infty(Q)}^{1/2} \sum_{j=0}^{m} \|\nabla G\|_{L^\infty(\hat{Q})}^{j} |\check{v}|_{H^j(Q)}, \quad (16)$$

$$|\check{v}|_{H^m(Q)} \leq C_{shape} \left\| \det \nabla G \right\|_{L^\infty(\hat{Q})}^{1/2} \|\nabla G\|_{L^\infty(\hat{Q})}^{-m} \sum_{j=0}^{m} |v|_{H^j(\hat{Q})}. \quad (17)$$

hold for all $\check{v} \in H^m(Q)$ and their counterparts $v \in H^m(\hat{Q})$, where C_{shape} are positive generic constants that only depend on the shape of the geometry Ω and its parametrization.

We note that the 0-order terms in the upper bounds of Proposition 1 are not needed for $m > 0$. They are incorporated in order to give a unified presentation for $m \geq 0$. Hence, as a special case of Proposition 1, we obtain the following estimates for the L^2 norm and H^1 seminorm

Corollary 1. *Let $\hat{Q} \in \hat{\mathcal{Q}}_h$ and $Q = G(\hat{Q})$.*
For $\check{v} \in L^2(Q)$, we have

$$\|v\|_{L^2(\hat{Q})} \leq C_{shape} \left\| \det \nabla G^{-1} \right\|_{L^\infty(Q)}^{1/2} \|\check{v}\|_{L^2(Q)}, \quad (18)$$

$$\|\check{v}\|_{L^2(Q)} \leq C_{shape} \left\| \det \nabla G \right\|_{L^\infty(\hat{Q})}^{1/2} \|v\|_{L^2(\hat{Q})}, \quad (19)$$

and, for $\check{v} \in H^1(Q)$, we can write

$$|v|_{H^1(\hat{Q})} \leq C_{shape} C |\check{v}|_{H^1(Q)}, \quad (20)$$

$$|\check{v}|_{H^1(Q)} \leq C_{shape} C |v|_{H^1(\hat{Q})}, \quad (21)$$

where the generic constants C only depends on the geometrical mapping.

The second result provides the quantitative approximation power of NURBS. We mention that B-Splines are special case of NURBS. It basically states that a NURBS space has the same approximation power as a FE space of same degree. We do not give any further technical details. We refer the reader to [2] for a more comprehensive discussion of the approximation properties. In particular, the following theorem is proved in [19], see also Theorem 3.2 in [2].

Theorem 1. *Let k and l be integers with $0 \le k \le l \le p + 1$, $l \ge r$, and $Q = G(\hat{Q})$ with $\hat{Q} \in \hat{\mathcal{Q}}_h$. Then there exists a projective operator $\Pi_{V_h} : L^2(\Omega) \to V_h$ from $L^2(\Omega)$ to the NURBS space V_h such that the approximation error estimates*

$$\sum_{Q \in \mathcal{Q}_h} |v - \Pi_{V_h} v|^2_{H^k(Q)} \le C_{shape} \sum_{Q \in \mathcal{Q}_h} h_Q^{2(l-k)} \sum_{i=0}^{l} \|\nabla G\|^{2(i-l)}_{L^\infty(\hat{\bar{Q}})} |v|^2_{H^i(\overline{Q})}$$

hold for all $v \in H^l(\Omega)$, where $\overline{Q}$ denotes the so-called support extension of Q, and the constant C_{shape} only depends on the geometry and its parametrization.

3   The IsogEometric Tearing and Interconnecting method and its Analysis

The IETI method, that was introduced in [4], is the adaption of the FETI method (see, e.g., [6] or [5]) to isogeometric analysis. According to [13] based on algebraic arguments, the BDDC preconditioner and the FETI-DP method possess the same spectrum up to zeros and ones. Hence a condition number bound for BDDC implies a bound for FETI-DP and vice versa. Since the proof is based on algebraic arguments, it also holds for the IETI-DP method. It was first introduced in [4], and further analysed in [12].

In the following, let V_h be the conform IgA space which fulfils the Dirichlet boundary conditions as defined in Section 2.4.1. Furthermore, we denote by $\{\check{N}_{i,p}\}_{i \in \mathcal{I}}$ the B-Spline basis of this space. The idea of IETI is to introduce local spaces, which are independent of each other, and consider only the local problems. The coupling and the continuity across interfaces is received via additional constraints.

3.1 Local spaces and Jump operator

Let $V_h^{(k)}$ be the space of B-Spline functions defined on patch $\Omega^{(k)}$, see (8). Analogously to the splitting introduced in Section 2.4, we define the local interface space

$$W^{(k)} := \operatorname{span}\{\check{N}_{i,p}|\operatorname{supp}\{\check{N}_{i,p}\} \cap (\partial\Omega^{(k)} \cap \Gamma) \neq \emptyset, i \in \mathcal{I}\},$$

which is the restriction of $V_{\Gamma,h}$ to $\Omega^{(k)}$. In the following, in order to avoid cumbersome notation, we define the patch boundary to be just the interface part, i.e. $\partial\Omega^{(k)} := \partial\Omega^{(k)} \cap \Gamma$. Hence, we have

$$V_h^{(k)} = W^{(k)} \oplus V_{I,h}^{(k)},$$

where $V_{I,h}^{(k)}$ is defined as in (10) or (11). Furthermore, we define the space of functions, which are locally in $W^{(k)}$, by

$$W := \prod_{k=1}^{N} W^{(k)}.$$

We note that functions from W are, in general, not continuous across the interface, i.e. $W \not\subseteq C^0$. A function $w \in W^{(k)}$ has components $w := \left[w^{(k)}\right]_{k=1}^{N} \leftrightarrow \left[\boldsymbol{w}^{(k)}\right]_{k=1}^{N} =: \boldsymbol{w}$.

In order to obtain continuous functions, we introduce additional constraints which will enforce the continuity. Let $\mathcal{B}(k,l)$ be the set of all coupled indices between $\Omega^{(k)}$ and $\Omega^{(l)}$, then we enforce the following constraints

$$\boldsymbol{w}_i^{(k)} - \boldsymbol{w}_j^{(l)} = 0 \qquad \forall (i,j) \in \mathcal{B}(k,l), k > l. \tag{22}$$

The operator $B : W \to U^* := \mathbb{R}^\Lambda$, which realizes constraints (22) in the form

$$Bw = 0,$$

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.

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

3.1.1 Saddle point formulation

Due to the multipatch structure of our physical domain, we can decompose the bilinear form and the right hand side functional as follows

$$a\big(u_h, v_h\big) = \sum_{k=1}^{N} a^{(k)}\big(u_h^{(k)}, v_h^{(k)}\big),$$

$$\langle F, v_h \rangle = \sum_{k=1}^{N} \left\langle F^{(k)}, v_h^{(k)} \right\rangle,$$

where $u_h, v_h \in V_h$ and according to Section 2.4 $u_h^{(k)}, v_h^{(k)}$ denote its restriction to $\Omega^{(k)}$. By means of the B-Spline basis we can rewrite the variational problem as linear system

$$\left(\sum_{k=1}^{N} \boldsymbol{A} \boldsymbol{K}^{(k)} \boldsymbol{A}^T \right) \boldsymbol{u} = \sum_{k=1}^{N} \boldsymbol{A} \boldsymbol{f}^{(k)}, \tag{23}$$

where $\boldsymbol{A}$ is the Boolean patch assembling matrix.

Analogously to Section 2.4.2, we can reorder the entries of the patch local stiffness matrix and right-hand side

$$\boldsymbol{K}^{(k)} = \begin{bmatrix} \boldsymbol{K}_{BB}^{(k)} & \boldsymbol{K}_{BI}^{(k)} \\ \boldsymbol{K}_{IB}^{(k)} & \boldsymbol{K}_{II}^{(k)} \end{bmatrix}, \qquad \boldsymbol{f}^{(k)} = \begin{bmatrix} \boldsymbol{f}_B^{(k)} \\ \boldsymbol{f}_I^{(k)} \end{bmatrix}.$$

The equation $\boldsymbol{K}^{(k)} \boldsymbol{u}^{(k)} = \boldsymbol{f}^{(k)}$ is equivalent to

$$\boldsymbol{S}^{(k)} \boldsymbol{u}_B^{(k)} = \boldsymbol{g}^{(k)},$$

where $\boldsymbol{S}^{(k)} = \boldsymbol{K}_{BB}^{(k)} - \boldsymbol{K}_{BI}^{(k)}(\boldsymbol{K}_{II}^{(k)})^{-1}\boldsymbol{K}_{IB}^{(k)}$ and $\boldsymbol{g}^{(k)} = \boldsymbol{f}_B^{(k)} - \boldsymbol{K}_{BI}^{(k)}(\boldsymbol{K}_{II}^{(k)})^{-1}\boldsymbol{f}_I^{(k)}$. We obtain that equation (23) can be reformulated as

$$\left(\sum_{k=1}^{N} \boldsymbol{A} \boldsymbol{S}^{(k)} \boldsymbol{A}^T \right) \boldsymbol{u}_B = \sum_{k=1}^{N} \boldsymbol{A} \boldsymbol{g}^{(k)}. \tag{24}$$

Similarly, we can express (24) in operator notation as

$$\sum_{k=1}^{N} \left\langle S^{(k)} u_{B,h}^{(k)}, v_{B,h}^{(k)} \right\rangle = \sum_{k=1}^{N} \left\langle g^{(k)}, v_{B,h}^{(k)} \right\rangle \quad \forall v_{B,h} \in V_{\Gamma,h}, \tag{25}$$

where $u_{B,h} \in V_{\Gamma,h}$, $g^{(k)} \in V_{\Gamma,h}^{(k)*}$ and $S^{(k)} : V_{\Gamma,h}^{(k)} \to V_{\Gamma,h}^{(k)*}$.

Instead of defining the global Schur complement and the right-hand side functional on the conforming space $V_{\Gamma,h}$, we define an *extended* version of the Schur complement on the "discontinuous" space W, i.e.

$$S : W \to W^*$$

$$\langle Sv, w \rangle := \sum_{k=1}^{N} \left\langle S^{(k)} v^{(k)}, w^{(k)} \right\rangle \quad \text{for } v, w \in W,$$

$$g \in W^*$$

$$\langle g, w \rangle := \sum_{k=1}^{N} \left\langle g^{(k)}, w^{(k)} \right\rangle \quad \text{for } w \in W.$$

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

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

The next step is to reformulate (25) in terms of S and B in the space W. Due to the symmetry of $a(\cdot, \cdot)$, we can write (25) as minimization problem

$$u_{B,h} = \underset{v \in V_{\Gamma,h}}{\text{argmin}} \frac{1}{2} \sum_{k=1}^{N} \left(\left\langle S^{(k)} v^{(k)}, v^{(k)} \right\rangle - \left\langle g^{(k)}, v^{(k)} \right\rangle \right).$$

It is easy to see that this problem is equivalent to

$$u_{B,h} = \underset{w \in W, Bw=0}{\text{argmin}} \frac{1}{2} \langle Sw, w \rangle - \langle g, w \rangle.$$

In the following we will only work with the Schur complement system and hence, to simplify the notation, we will use u instead of $u_{B,h}$, when we consider functions in $V_{\Gamma,h}$. If there has to be made a distinction between $u_h, u_{B,h}$ and $u_{I,h}$, we will write the subscript letter.

This constraint minimization problem can be rewritten in form of the following saddle point problem: find $(u, \boldsymbol{\lambda}) \in W \times U$ such that

$$\begin{bmatrix} S & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} g \\ 0 \end{bmatrix}, \tag{26}$$

Theorem 2. *If $ker(S) \cap ker(B) = \{0\}$, then the above problem is uniquely solveable up to adding elements from $ker(B^T)$ to $\boldsymbol{\lambda}$.*

Remark 1. We note that problem (26) is uniquely solvable with respect to u (if K is regular). However, not all $S^{(k)}$ are regular, since those, which do not lie on a Dirichlet boundary correspond to pure Neumann problems. The usual strategy for solving (26) is to work with its Schur complement F, but since some blocks of S are singular, the Schur complement is not well defined. In classical FETI/IETI, one adds the basis of each local kernel to the space and regularizes the matrix. Unfortunately, this need an exact knowledge of the kernels, which is in general not trivial. The dual primal approach presented below circumvents this by restricting the space W in order to guarantee the regularity of each $S^{(k)}$.

3.2   Intermediate space and primal constraints

In order to guarantee the positive definiteness of S, we are looking for an intermediate space $\widetilde{W}$ in the sense $\widehat{W} \subset \widetilde{W} \subset W$ such that S restricted to $\widetilde{W}$ is SPD. Let $\Psi \subset V_{\Gamma,h}^*$ be a set of linearly independent *primal variables*. Then we define the spaces

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

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

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*. For the analysis, the subspace W_Π is not required. However, it brings advantages in the implementation.

If we choose Ψ, such that $\widetilde{W} \cap \ker(S) = \{0\}$, then

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

is invertible. If a set Ψ fulfils $\widetilde{W} \cap \ker(S) = \{0\}$, then we say that Ψ *controls the kernel* and in the following, we will always assume that such a set is chosen.

In the literature, there are 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: $\psi^{\mathcal{F}}(v) = \frac{1}{|\mathcal{F}|} \int_{\mathcal{F}} v\, ds$.

The typical choices for Ψ are usually called Algorithm A - C:

- Algorithm A: $\Psi = \{\psi^{\mathcal{V}}\}$,

- Algorithm B: $\Psi = \{\psi^{\mathcal{V}}\} \cup \{\psi^{\mathcal{E}}\} \cup \{\psi^{\mathcal{F}}\}$,

- Algorithm C: $\Psi = \{\psi^{\mathcal{V}}\} \cup \{\psi^{\mathcal{E}}\}$.

Moreover, one finds references to two further choices for Ψ, commonly referred to Algorithm D and E, which are aiming for a reduced set of primal variables, see, e.g. [6] Algorithm 6.28 and 6.29. This algorithms address the issue of the rapidly increasing number of primal variables.

Remark 2. For domains $\Omega \subset \mathbb{R}^2$, Algorithm A will provide a quasi optimal method for the Poisson problem. By choosing additional primal variables, the coarse problem will grow. Hence, it becomes computationally more demanding. However, it brings benefits in the condition number. For three-dimensional domains, one can show that just choosing vertex evaluation does not lead to a quasi optimal method. In such cases, additional primal variables have to be chosen.

3.2.1 IETI - DP 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

$$\widetilde{B} := B\widetilde{I} : \widetilde{W} \to U^*. \tag{27}$$

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

$$\begin{bmatrix} \widetilde{S} & \widetilde{B}^T \\ \widetilde{B} & 0 \end{bmatrix} \begin{bmatrix} u \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \widetilde{g} \\ 0 \end{bmatrix}, \tag{28}$$

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, $\widetilde{S}$ is SPD on $\widetilde{W}$. Hence, we can define the Schur complement F and the corresponding right-hand side of equation (28) as follows:

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

Hence, the saddle point system is equivalent to solving:

$$\text{find } \boldsymbol{\lambda} \in U : \quad F\boldsymbol{\lambda} = d. \tag{29}$$

By means of Brezzi's theorem, we obtain

Theorem 3. *The above problem is uniquely solvable up to adding elements from* $\ker(\widetilde{B}^T)$ *to* $\boldsymbol{\lambda}$*. The unique solution*

$$u = \widetilde{S}^{-1}(\widetilde{g} - \widetilde{B}^T\boldsymbol{\lambda})$$

satisfies $u \in \widehat{W} \equiv V_{\Gamma,h}$ *and is the unique solution of (13).*

We note that F is SPSD on U. According to [5], if we set

$$\widetilde{U} := U_{/\ker(\widetilde{B}^T)},$$
$$\widetilde{U}^* := \mathcal{R}(\widetilde{B}),$$

where $\widetilde{U}^*$ is in fact the dual of $\widetilde{U}$, then F restricted to $\widetilde{U}^*$ is SPD, i.e. $F_{|\widetilde{U}} : \widetilde{U} \to \widetilde{U}^*$. Hence, it is possible to solve (29) with PCG.

3.2.2 BDDC - Preconditioner Following the lines in [5] a different but equivalent way leads to the so called *BDDC - Preconditioner*. Here we start from the equation

$$\hat{S}\hat{u} = \hat{g}, \tag{30}$$

where the notation $\hat{\cdot}$ indicates that the operator and the functions are restricted to the continuous space $V_{\Gamma,h}$, i.e., just the standard Schur complement. Since $\hat{S}$ is SPD, we can solve system (30) by the PCG method preconditioned by the so called BDDC preconditioner M_{BDDC}^{-1}.

As in the previous section, using a set of linearly independent primal variables $\bar{\Psi}$ and the corresponding spaces, we define the BDDC preconditioner as follows

$$M_{BDDC}^{-1} := \widetilde{E}_D \widetilde{S}^{-1} \widetilde{E}_D^T,$$

where $\widetilde{E}_D^T$ is defined via the formulas

$$\begin{aligned}
\widetilde{E}_D &= E_D \widetilde{I} : \quad \widetilde{W} \to \widehat{W}, \\
E_D &= I - P_D : \quad W \to \widehat{W}, \\
P_D &= B_D^T B : \quad W \to W.
\end{aligned}$$

One can give an alternative formulation see, e.g., [12]. Here we assume, that after a change of basis, each primal variable corresponds to one degree of freedom, see, e.g., [20]. Let $\boldsymbol{K}^{(k)}$ be the stiffness matrix corresponding to $\Omega^{(k)}$. We splitting the degrees of freedom into interior (I) and interface (B) ones. Furthermore, we again split the interface degrees of freedoms into primal (Π) and dual (Δ) ones. This provides a partition of $\boldsymbol{K}^{(k)}$ into 2 x 2 and 3 x 3 block systems:

$$\boldsymbol{K}^{(k)} = \begin{bmatrix} \boldsymbol{K}_{II}^{(k)} & \boldsymbol{K}_{IB}^{(k)} \\ \boldsymbol{K}_{BI}^{(k)} & \boldsymbol{K}_{BB}^{(k)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{II}^{(k)} & \boldsymbol{K}_{I\Delta}^{(k)} & \boldsymbol{K}_{I\Pi}^{(k)} \\ \boldsymbol{K}_{\Delta I}^{(k)} & \boldsymbol{K}_{\Delta\Delta}^{(k)} & \boldsymbol{K}_{\Delta\Pi}^{(k)} \\ \boldsymbol{K}_{\Pi I}^{(k)} & \boldsymbol{K}_{\Pi\Delta}^{(k)} & \boldsymbol{K}_{\Pi\Pi}^{(k)} \end{bmatrix}.$$

In order to the define the preconditioner, we need the following restriction and interpolation operators:

$$
\begin{aligned}
R_{B\Delta} &: \widetilde{W} \to W_{\Delta}, \\
R_{B\Pi} &: \widetilde{W} \to W_{\Pi}, \\
R_{\Delta}^{(k)} &: W_{\Delta} \to W_{\Delta}^{(k)}, \\
R_{\Pi}^{(k)} &: W_{\Pi} \to W_{\Pi}^{(k)}.
\end{aligned}
\tag{31}
$$

We now derive a scaled version $R_{D,\Delta}^{(k)}$ from $R_{\Delta}^{(k)}$ by multiplying its i-th row with $\delta_i^{\dagger(k)}$, and then we define

$$
R_{D,B} := R_{B\Pi} \oplus \left(\sum_{k=1}^{N} R_{D,\Delta}^{(k)} \right) R_{B\Delta}.
\tag{32}
$$

The BDDC preconditioner is now determined by

$$
M_{BDDC}^{-1} := R_{D,B}^T \widetilde{S}^{-1} R_{D,B},
$$

where

$$
\widetilde{S}^{-1} = R_{B\Delta}^T \left(\sum_{k=1}^{N} \begin{bmatrix} 0 & R_{\Delta}^{(k)^T} \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_{II}^{(k)} & \boldsymbol{K}_{I\Delta}^{(k)} \\ \boldsymbol{K}_{\Delta I}^{(k)} & \boldsymbol{K}_{\Delta\Delta}^{(k)} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ R_{\Delta}^{(k)} \end{bmatrix} \right) R_{B\Delta} + \Phi \boldsymbol{S}_{\Pi\Pi}^{-1} \Phi^T.
$$

Here the matrices $\boldsymbol{S}_{\Pi\Pi}$ and Φ are given by

$$
\boldsymbol{S}_{\Pi\Pi} = \sum_{k=1}^{N} R_{\Pi}^{(k)^T} \left(\boldsymbol{K}_{\Pi\Pi}^{(k)} - \begin{bmatrix} \boldsymbol{K}_{\Pi I}^{(k)} & \boldsymbol{K}_{\Pi\Delta}^{(k)} \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_{II}^{(k)} & \boldsymbol{K}_{I\Delta}^{(k)} \\ \boldsymbol{K}_{\Delta I}^{(k)} & \boldsymbol{K}_{\Delta\Delta}^{(k)} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{K}_{I\Pi}^{(k)} \\ \boldsymbol{K}_{\Delta\Pi}^{(k)} \end{bmatrix} \right) R_{\Pi}^{(k)}
$$

and

$$
\Phi = R_{B\Pi}^T - R_{B\Delta}^T \sum_{k=1}^{N} \left(\begin{bmatrix} 0 & R_{\Delta}^{(k)^T} \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_{II}^{(k)} & \boldsymbol{K}_{I\Delta}^{(k)} \\ \boldsymbol{K}_{\Delta I}^{(k)} & \boldsymbol{K}_{\Delta\Delta}^{(k)} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{K}_{I\Pi}^{(k)} \\ \boldsymbol{K}_{\Delta\Pi}^{(k)} \end{bmatrix} \right) R_{\Pi}^{(k)},
$$

respectively.

3.3 Preconditioning

The existent theory for FETI provides us with the following estimate for the condition number of F, see, e.g., [5],

Theorem 4. *Let $H^{(k)}$ be the diameter of patch k. Then, under suitable assumptions imposed on the mesh, we have*

$$\kappa(F_{|\tilde{U}}) \leq C \left(\max_k \frac{H^{(k)}}{h^{(k)}} \right),$$

where the constant positive C is dependent on α.

Since this condition number bound is not optimal in comparison with other precondition strategies like multigrid, there is a need for additional preconditioning of F. This can be done by the so called *Dirichlet preconditioner* or its scaled versions. For finite elements, the Dirichlet preconditioner provides a quasi optimal condition number bound, which is not robust with respect to the diffusion coefficient. The *Dirichlet preconditioner* M_D^{-1} is defined as

$$M_D^{-1} = BSB^T.$$

We note that this preconditioning just uses the block diagonal version of the Schur complement. Hence, the application can be done in parallel.

In order to receive robustness with respect to the diffusion coefficient α, we use the scaled version the of Dirichlet preconditioner, the so called *scaled Dirichlet preconditioner*. The scaling is incorporated in the application of the jump operator. Therefore, we define the *scaled* jump operator B_D, such that the operator enforces the constraints:

$$\delta_j^{\dagger (l)} \boldsymbol{w}_i^{(k)} - \delta_i^{\dagger (k)} \boldsymbol{w}_j^{(l)} = 0 \qquad \forall (i,j) \in \mathcal{B}(k,l), k > l,$$

$$\text{with } \delta_i^{\dagger (k)} = \frac{\rho_i^{(k)}}{\sum_l \rho_{j_l}^{(l)}},$$

where j_l is the corresponding coefficient index on the neighbouring patch $\Omega^{(l)}$. The scaled Dirichlet preconditioner has the following form

$$M_{sD}^{-1} = B_D S B_D^T. \tag{33}$$

Typical choices for $\rho_i^{(k)}$ are

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

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

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

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 too large, one should use the other scalings to obtain robustness.

Theorem 5. *Let $H^{(k)}$ be the diameter and $h^{(k)}$ the local mesh size of $\Omega^{(k)}$ and let M_{sD}^{-1} be the scaled Dirichlet preconditioner.*

Then, under suitable assumption imposed on the mesh, we have

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

where the positive constant C is independent of h and H.

In the case of IgA, a more general proof in the sense, that not only C^0 smoothness across patch interfaces is allowed but also C^l, $l \geq 0$ smoothness, can be found in [12]. However, the proof is restricted to the case of a domain decomposition, which is obtained by subdividing a single patch, i.e. performing a decomposition of the parameter domain. Hence, always the same geometrical mapping G is used. Furthermore, due to the C^l, $l \geq 0$, smoothness across interfaces, only a condition number bound of $O\left((1 + \log H/h)H/h\right)$ could be proven for stiffness scaling. In the proceeding section, we will extend the proof given in [12] to multipatch domains, which consists of different geometrical mappings $G^{(k)}$ for each patch. Additionally, for $l = 0$, we again obtain quasi-optimal condition number bounds also for stiffness scaling.

3.4 Analysis of BDDC-Preconditiner

In this section we rephrase the results and notations established in [12], and extend them to multipatch domains, consisting of a different geometrical mapping $G^{(k)}$ for each patch. However, we only allow C^0 smoothness across the patch interfaces.

3.4.1   General Results Let $\check{z}$ be a function from V_h. Its restriction to a patch $\Omega^{(k)}$ belongs to $V_h^{(k)}$, and can be written as

$$\check{z}^{(k)}(x) := \check{z}(x)_{|\Omega^{(k)}} = \sum_{i \in \mathcal{I}^{(k)}} c_i^{(k)} \check{N}_{i,p}^{(k)}(\xi),$$

where in $\mathcal{I}^{(k)}$ all indices, where the basis functions have a support on the Dirichlet boundary in the physical space $\Omega^{(k)}$, are excluded. The corresponding spline function in the parameter space is denoted by $z^{(k)}(\xi) \in S_h^{(k)}$. It is important to note, that the geometrical map G and its inverse G^{-1} are independent of h, since it is fixed on a coarse discretization. When the domain becomes refined, G stays the same. Clearly, the same applies for the gradients and it can be assumed, that $G^{(k)} \in W^{1,\infty}((0,1)^d)$ for all $k \in \{1,\dots,N\}$.

Let $\hat{Q} \in \hat{\mathcal{Q}}_h^{(k)}$ be a cell in the parameter domain $(0,1)^d$, and $Q \in \mathcal{Q}_h^{(k)}$ be a cell in the physical domain $\Omega^{(k)}$. Then we denote all indices of basis functions, which have a support on $\hat{Q}$ and Q, respectively, by

$$\mathcal{I}^{(k)}(\hat{Q}) := \{i \mid \hat{Q} \subseteq \operatorname{supp}\{N_{i,p}^{(k)}\}\} = \{i \mid Q \subseteq \operatorname{supp}\{\check{N}_{i,p}^{(k)}\}\}. \quad (34)$$

We will now define a local discrete norm based on control points c_i.

Definition 7. *Let $\hat{Q} \in \hat{\mathcal{Q}}_h$, $Q = G^{(k)}(\hat{Q})$, $\check{z} \in V_h^{(k)}$ and z its counterpart in the parameter domain. We define*

$$|z|_{\hat{Q}}^2 := \left(\max_{i \in \mathcal{I}^{(k)}(\hat{Q})} |c_i^{(k)}|^2 \right) h_{\hat{Q}}^2,$$

$$|\check{z}|_Q^2 := \left(\max_{i \in \mathcal{I}^{(k)}(\hat{Q})} |c_i^{(k)}|^2 \right) h_Q^2,$$

where $h_Q = \|\nabla G\|_{L^\infty(\hat{Q})} \, h_{\hat{Q}}$.

Remark 3. Note that the two discrete norms are obviously equivalent, since

$$
\begin{aligned}
|z|_{\hat{Q}}^2 &= \left(\max_{i \in \mathcal{I}^{(k)}(\hat{Q})} |c_i|^2 \right) h_{\hat{Q}}^2 = \left(\max_{i \in \mathcal{I}^{(k)}(\hat{Q})} |c_i|^2 \right) \|\nabla G\|_{L^\infty(\hat{Q})}^{-2} h_Q^2 \\
&= \|\nabla G\|_{L^\infty(\hat{Q})}^{-2} |\check{z}|_Q^2.
\end{aligned}
\quad (35)
$$

Lemma 1. *Let $\hat{Q} \in \hat{\mathcal{Q}}_h$, $Q = G^{(k)}(\hat{Q})$, $\check{z} \in V_h^{(k)}$ and z its counterpart in the parameter domain, then*

$$\|z\|_{L^2(\hat{Q})} \approx |z|_{\hat{Q}}. \tag{36}$$

Proof. See [12]. □

From Lemma 1 and Corrolary 1, we obtain

Corollary 2. *Let $\hat{Q} \in \hat{\mathcal{Q}}_h$, $Q = G^{(k)}(\hat{Q})$, $\check{z} \in V_h^{(k)}$ and z its counterpart in the parameter domain. Then we have*

$$\|\check{z}\|_{L^2(Q)} \approx |\check{z}|_Q^2. \tag{37}$$

Proof. Indeed, it is easy to see that

$$\|\check{z}\|_{L^2(Q)}^2 \leq \|\det \nabla G\|_{L^\infty(\hat{Q})} \|z\|_{L^2(\hat{Q})}^2 \approx \|\det \nabla G\|_{L^\infty(\hat{Q})} |z^{(k)}|_{\hat{Q}}^2,$$

and

$$|z^{(k)}|_{\hat{Q}}^2 \approx \|z\|_{L^2(\hat{Q})}^2 \leq \left\|\det \nabla G^{-1}\right\|_{L^\infty(Q)} \|\check{z}\|_{L^2(Q)}^2,$$

which yield (37). □

It immediately follows that the norm in the physical space $\|\check{z}\|_{L^2(Q)}$ is also equivalent to $|z|_{\hat{Q}}^2$, since $\|\check{z}\|_{L^2(Q)} \approx |\check{z}|_Q^2 \approx |z|_{\hat{Q}}^2$. Now we define a global patchwise norm, which will be again equivalent to the L^2-norm in the parameter domain.

Definition 8. *Let $\check{z} \in V_h^{(k)}$ and z its counterpart in the parameter domain. The global discrete norm is defined by*

$$|z|_\Box^2 := \sum_{i \in \mathcal{I}^{(k)}} |c_i^{(k)}|^2 h_{\hat{Q}}^2. \tag{38}$$

Lemma 2. *Let $\check{z} \in V_h^{(k)}$ and z its counterpart in the parameter domain. Then we have the norm equivalence*

$$|z|_\Box^2 \approx \|z\|_{L^2((0,1)^d)}^2. \tag{39}$$

Proof. See [12].   □

Again, it follows that the norm is also equivalent to the corresponding one in the physical space.

Corollary 3. *Let $\check{z} \in V_h^{(k)}$ and z its counterpart in the parameter domain. Then we have the norm equivalence*

$$\|\check{z}\|_{L^2(\Omega^{(k)})} \approx |z|^2_{\square}.$$

Proof. The proof is identical to that of Corrolary 2 with the same constants.   □

The next step is to define a discrete H^1 seminorm based on the coefficients c_i. We denote by $c_{i,i^\iota - j}$ the coefficient corresponding to basis function $N_{(i^1,\ldots,i^\iota - j,\ldots,i^d),p}$, see (5). The derivative with respect to ξ^ι of a B-Spline z can be written as

$$\frac{\partial z}{\partial \xi^\iota} = \sum_{i \in \mathcal{I}_\iota} \left(\frac{c_{i,i^\iota} - c_{i,i^\iota - 1}}{\Delta_i^\iota} \right) N_{i,(p,p^\iota - 1)}, \tag{40}$$

where $\mathcal{I}_\iota$ is the set of admissible indices such that each summand is well defined, $\Delta_i^\iota = \xi_{i+p}^\iota - \xi_i^\iota$, and $N_{i,(p,p^\iota - 1)}$ is a tensor B-Spline of degree p, where its degree in dimension ι is reduced by one.

With this definition at hand, we are able to define a discrete seminorm $|\cdot|_{\xi^\iota, \hat{Q}}$ and $|\cdot|_{\xi^\iota}$ in the parameter domain as follows:

Definition 9. *Let $\hat{Q} \in \hat{\mathcal{Q}}_h$, $Q = G^{(k)}(\hat{Q})$, $\check{z} \in V_h^{(k)}$, and z its counterpart in the parameter domain. Then we define the local discrete seminorm by*

$$|z|^2_{\xi^\iota, \hat{Q}} := \max_{i \in \mathcal{I}_\iota^{(k)}(\hat{Q})} |c_{i,i^\iota}^{(k)} - c_{i,i^\iota - 1}^{(k)}|^2, \tag{41}$$

where $\mathcal{I}_\iota^{(k)}(\hat{Q}) := \{i \in \mathcal{I}^{(k)}(\hat{Q}) \mid \hat{Q} \subseteq supp\{N_{i,(p,p^\iota-1)}\}$. The global patchwise counterpart is then defined by

$$|z|^2_{\xi^\iota} := \sum_{i \in \mathcal{I}_\iota^{(k)}} |c_{i,i^\iota}^{(k)} - c_{i,i^\iota - 1}^{(k)}|^2. \tag{42}$$

This seminorms are again equivalent to the standard Sobolev seminorms.

Lemma 3. *Let $\hat{Q} \in \hat{\mathcal{Q}}_h$, $Q = G^{(k)}(\hat{Q})$, $\check{z} \in V_h^{(k)}$ and z its counterpart in the parameter domain. Then we have the equivalence*

$$|z|^2_{\xi^\iota, \hat{Q}} \approx \left\| \frac{\partial z}{\partial \xi^\iota} \right\|^2_{L^2(\hat{Q})}. \tag{43}$$

Proof. See [12]. $\qquad \square$

We are now in the position to define the "full" discrete seminorm

Definition 10. *Let $\check{z} \in V_h^{(k)}$, and z its counterpart in the parameter domain. Then we define the "full" discrete seminorm by the formula*

$$|z|^2_\nabla := \sum_{\iota=1}^d |z|^2_{\xi^\iota}. \tag{44}$$

Lemma 3 and Corrolary 1 immediately yield the following proposition.

Proposition 2. *Let $\check{z} \in V_h^{(k)}$ and z its counterpart in the parameter domain. Then the following equivalences hold:*

$$|z|^2_\nabla \approx |z|^2_{H^1((0,1)^d)} \approx |\check{z}|^2_{H^1(\Omega^{(k)})}. \tag{45}$$

The next step is to provide properties in the local index spaces. Since we consider only the two dimensional problem, we can interpret the control points $(c_i)_{i\in\mathcal{I}}$ as entries of a matrix $C = (c_i)_{i=1}^{M_1,M_2}$. Before doing that we will provide abstract results for an arbitrary matrix.

We define the seminorm

$$\|C\|^2_\nabla := \sum_{\iota=1}^2 \sum_{\substack{i=1 \\ i^\iota=2}}^{M_\iota} |c_{i,i^\iota} - c_{i,i^\iota-1}|^2 \tag{46}$$

for a real valued $M_1 \times M_2$ matrix $C = (c_i)_{i=1}^{M_1,M_2} \in \mathbb{R}^{M_1 \times M_2}$. The entries of the matrix C can be interpreted as values on a uniform grid $\mathcal{T}$.

This motivates the definition of an operator, which evaluates a continuous function on the grid points $(x_i) = (x_{i^1 i^2})$:

$$
\begin{aligned}
(\cdot)_I : C([0,1]^2) &\to \mathbb{R}^{M_1 \times M_2}, \\
f &\mapsto f_I : (f_I)_i = f(x_i),
\end{aligned}
\tag{47}
$$

and an operator, that provides a piecewise bilinear interpolation of the given grid values

$$
\begin{aligned}
\chi : \mathbb{R}^{M_1 \times M_2} &\to \mathcal{Q}_1(\mathcal{T}) \subset H^1([0,1]^2), \\
\boldsymbol{C} &\mapsto \chi(\boldsymbol{C}) : \chi(\boldsymbol{C})(x_i) = c_i,
\end{aligned}
\tag{48}
$$

where $\mathcal{Q}_1(\mathcal{T})$ is the space of piecewise bilinear functions on $\mathcal{T}$.

Furthermore, given values on an edge on $[0,1]^2$, we need to define its linear interpolation and a discrete harmonic extension to the interior. In order to do so, let e be an edge on $[0,1]^2$, and let us denote all indices of grid points x_i associated to e by $\mathcal{I}(e)$. Additionally, let $\mathcal{P}_1(\mathcal{T}_{|e})$ be the space of piecewise linear spline functions on $\mathcal{T}_{|e}$.

We define the interpolation of values on $\mathcal{I}(e)$ by the restriction of the operator χ to e:

$$
\begin{aligned}
\chi_e : \mathbb{R}^{M_\iota} &\to H^1(e) \\
\boldsymbol{v} &\mapsto \chi_e(\boldsymbol{v}) \in \mathcal{P}_1(\mathcal{T}_{|e}) : \chi_e(\boldsymbol{v})(x_i) = v_{i^\iota}, \quad i \in \mathcal{I}(e), 1 \le i^\iota \le M_\iota,
\end{aligned}
$$

where e corresponds to dimension ι, $\iota \in \{1, 2\}$.

This leads to a definition of a seminorm for grid points on an edge e via the interpolation to functions from $\mathcal{P}_1(\mathcal{T}_{|e})$:

Definition 11. *Let e be an edge of $[0,1]^2$ along dimension ι and $\boldsymbol{v}$ be a vector in $\mathbb{R}^{M_\iota}$. Then we define the following seminorm*

$$
\|\boldsymbol{v}\|_e := |\chi_e(\boldsymbol{v})|_{H^{1/2}(e)}, \quad \forall \boldsymbol{v} \in \mathbb{R}^{M_\iota}.
\tag{49}
$$

It remains to define the interpolation operator for the whole boundary, which will be the interpolation of all four edges.

Definition 12. *Let*

$$\mathcal{I}(\partial) := \{i : x_i \in \partial[0,1]^2\}. \tag{50}$$

be all indices i such that x_i lies on the boundary $\partial[0,1]^2$. For $\boldsymbol{b} \in \mathbb{R}^{|\mathcal{I}(\partial)|}$, let $\boldsymbol{b}_{|e}$ be all the values which are associated with e. The interpolation operator χ_∂ is then defined by

$$\begin{aligned} \chi_\partial &: \mathbb{R}^{|\mathcal{I}(\partial)|} \to H^1(\partial[0,1]^2) \\ \boldsymbol{b} &\mapsto \chi_\partial(\boldsymbol{b}) : \chi_\partial(\boldsymbol{b})_{|e} = \chi_e(\boldsymbol{b}_{|e}). \end{aligned} \tag{51}$$

Remark 4. For the interpolation operator χ_e defined on an edge e, the equations $\chi(\boldsymbol{C})_{|e} = \chi_e(\boldsymbol{C}_{|e})$ and $\left\|\left|\boldsymbol{C}_{|e}\right|\right\|_e = |\chi_e(\boldsymbol{C}_{|e})|_{H^{1/2}(e)} = |\chi(\boldsymbol{C})_{|e}|_{H^{1/2}(e)}$ are obviously valid.

Finally, we are able to define the discrete harmonic extension in $\mathbb{R}^{M_1 \times M_2}$.

Definition 13. *Let $\mathcal{H}_{\mathcal{Q}_1}$ be the standard discrete harmonic extension into the piecewise bilinear space $\mathcal{Q}_1$. This defines the lifting operator $\boldsymbol{H}$ by*

$$\begin{aligned} \boldsymbol{H} &: \mathbb{R}^{|\mathcal{I}(\partial)|} \to \mathbb{R}^{M_1 \times M_2} \\ \boldsymbol{b} &\mapsto \boldsymbol{H}(\boldsymbol{b}) := (\mathcal{H}_{\mathcal{Q}_1}(\chi_\partial(\boldsymbol{b})))_I. \end{aligned} \tag{52}$$

Theorem 6. *Let e be a particular side on the boundary of $[0,1]^2$ and the constant $\beta \in \mathbb{R}^+$ such that*

$$\beta^{-1}M_2 \leq M_1 \leq \beta M_2.$$

Then the following hold:

– *For all $\boldsymbol{b} \in \mathbb{R}^{2M_1+2M_2-1}$ that vanish on the four components corresponding to the four corners, it holds*

$$\left\|\left|\boldsymbol{H}(\boldsymbol{b})\right|\right\|_\nabla^2 \leq c(1 + \log^2 M_1) \sum_{e \in \partial[0,1]^2} \left\|\left|\boldsymbol{b}_{|e}\right|\right\|_e^2,$$

where the constant c depends only on β.

– *It holds*

$$\left\|\left|\boldsymbol{C}\right|\right\|_\nabla \geq c \left\|\left|\boldsymbol{C}_{|e}\right|\right\|_e, \quad \forall \boldsymbol{C} \in \mathbb{R}^{M_1 \times M_2},$$

where the constant c depends only on β.

Proof. See [12]. ∎

3.4.2 Condition number estimate The goal of this section to establish a condition number bound for $P = M_{BDDC}^{-1}\hat{S}$. Following [12], we assume that the mesh is quasi-uniform on each subdomain and the diffusion coefficient is globally constant. We focus now on a single patch $\Omega^{(k)}$, $k \in \{1, \dots, N\}$. For notational simplicity, we assume that the considered patch $\Omega^{(k)}$ does not touch the boundary $\partial\Omega$.

We define the four edges of the parameter domain $(0, 1)^2$ by E_r, and their images by $\check{E}_r = G^{(k)}(E_r)$, $r = 1, 2, 3, 4$. In order to represent functions $\check{w}^{(k)} \in W^{(k)}$ as the corresponding vectors, we need the coefficients corresponding to the basis functions on the four edges, namely

$$
\begin{aligned}
\mathcal{I}(\check{E}_1^{(k)}) &= \{i \mid i^1 = 1,\ i^2 = 1, 2, \dots M_2^{(k)}\}, \\
\mathcal{I}(\check{E}_2^{(k)}) &= \{i \mid i^1 = M_1^{(k)},\ i^2 = 1, 2, \dots M_2^{(k)}\}, \\
\mathcal{I}(\check{E}_3^{(k)}) &= \{i \mid i^1 = 1, 2, \dots M_1^{(k)},\ i^2 = 1\}, \\
\mathcal{I}(\check{E}_4^{(k)}) &= \{i \mid i^1 = 1, 2, \dots M_1^{(k)},\ i^2 = M_2^{(k)}\}, \\
\mathcal{I}(\Gamma^{(k)}) &= \bigcup_{1 \le r \le 4} \mathcal{I}(\check{E}_r^{(k)}).
\end{aligned}
\tag{53}
$$

Let $\check{z}^{(k)} \in V_h^{(k)}$, i.e., $\check{z}^{(k)} = \sum_{i \in \mathcal{I}^{(k)}} c_i^z \check{N}_{i,p}^{(k)}$. Hence, $\check{z}^{(k)}$ is determined by its coefficients $c_i^z = (c_{i^1,i^2}^z)_{i^1,i^2=1}^{M_1^{(k)},M_2^{(k)}}$, which can be interpreted as a $M_1^{(k)} \times M_2^{(k)}$ matrix $\boldsymbol{C}^z$, i.e.,

$$
V_h^{(k)} \ni \check{z}^{(k)} = \sum_{i \in \mathcal{I}^{(k)}} c_i^z \check{N}_{i,p}^{(k)} \iff \{c_i^z\}_{i \in \mathcal{I}^{(k)}} \iff \boldsymbol{C}^z = (c_{i^1,i^2}^z) \in \mathbb{R}^{M_1^{(k)} \times M_2^{(k)}}.
$$

Moreover, we can identify functions on the trace space $W^{(k)}$ as follows:

$$
W^{(k)} \ni \check{w}^{(k)} = \sum_{i \in \mathcal{I}(\Gamma^{(k)})} c_i^w \check{N}_{i,p}^{(k)} \iff \{c_i^w\}_{i \in \mathcal{I}^{(k)}(\Gamma^{(k)})},
$$

$$
W_{|E_r^{(k)}}^{(k)} \ni \check{w}^{(k)} = \sum_{i \in \mathcal{I}(\check{E}_r^{(k)})} c_i^w \check{N}_{i,p}^{(k)} \iff \{c_i^w\}_{i \in \mathcal{I}(\check{E}_r^{(k)})} \quad \text{for } r = 1, 2, 3, 4.
$$

Similar to the definition of the global discrete NURBS harmonic extension $\mathcal{H}$ in (12), we define the local patch version

$$
\mathcal{H}^{(k)} : W^{(k)} \to V_h^{(k)}
\tag{54}
$$

such that, for any $\check{w}^{(k)} \in W^{(k)}$, $\check{z}^{(k)} := \mathcal{H}^{(k)}(\check{w}^{(k)})$ is the unique function in V_h which minimizes the H^1 energy on $\Omega^{(k)}$ and $c_i^z = c_i^w$ $\forall i \in \mathcal{I}(\Gamma^{(k)})$. Finally, let $W_\Delta^{(k)} \subset W^{(k)}$ be the space of spline functions which vanish on the primal variables, i.e., in the corner points. The following theorem provides an abstract estimate of the condition number using the coefficient scaling:

Theorem 7. *Let the counting function $\delta^{\dagger(k)}$ be chosen accordingly to the coefficient scaling strategy. Assume that there exist two positive constants c_*, c^* and a boundary seminorm $|\cdot|_{W^{(k)}}$ on $W^{(k)}$, $k = 1, \ldots, N$, such that*

$$|\check{w}^{(k)}|_{W^{(k)}}^2 \leq c^* s^{(k)}(\check{w}^{(k)}, \check{w}^{(k)}) \quad \forall \check{w}^{(k)} \in W^{(k)}, \tag{55}$$

$$|\check{w}^{(k)}|_{W^{(k)}}^2 \geq c_* s^{(k)}(\check{w}^{(k)}, \check{w}^{(k)}) \quad \forall \check{w}^{(k)} \in W_\Delta^{(k)}, \tag{56}$$

$$|\check{w}^{(k)}|_{W^{(k)}}^2 = \sum_{r=1}^{4} |\check{w}^{(k)}_{|E_r^{(k)}}|_{W_r^{(k)}} \quad \forall \check{w}^{(k)} \in W^{(k)}, \tag{57}$$

where $|\check{w}^{(k)}_{|E_r^{(k)}}|_{W_r^{(k)}}$ is a seminorm associated to the edge spaces $W^{(k)}_{|E_r^{(k)}}$, with $r = 1, 2, 3, 4$. Then the condition number of the preconditioned BDDC operator P satisfies the bound

$$\kappa(M_{BDDC}^{-1} \hat{S}) \leq C(1 + c_*^{-1} c^*), \tag{58}$$

where the constant C is independent of h and H.

Proof. See [12] or [21]. $\qquad\square$

Using this abstract framework, we obtain the following condition number estimate for the BDDC preconditioner.

Theorem 8. *There exists a boundary seminorm such that the constants c_* and c^* of Theorem 7 are bounded by*

$$c^* \leq C_1, \tag{59}$$

$$c_*^{-1} \leq C_2 \max_{1 \leq k \leq N} \left(1 + \log^2\left(\frac{H^{(k)}}{h^{(k)}}\right)\right), \tag{60}$$

where the constants C_1 and C_2 are independent of H and h. Therefore, the condition number of the isogeometric preconditioned BDDC operator is bounded by

$$\kappa(M_{BDDC}^{-1}\hat{S}) \leq C \max_{1 \leq k \leq N} \left(1 + \log^2\left(\frac{H^{(k)}}{h^{(k)}}\right)\right), \qquad (61)$$

where the constant C is independent of H and h.

Proof. The proof essentially follows the lines of the proof given in [12] with a minor modification due to the different geometrical mappings $G^{(k)}$. We note that we only consider C^0 continuity across the patch interfaces, which makes the proof less technical.

The first step is to appropriately define the seminorm $|\check{w}^{(k)}|^2_{W^{(k)}}$ in $W^{(k)}$:

$$|\check{w}^{(k)}|^2_{W^{(k)}} := \sum_{r=1}^{4} |\check{w}^{(k)}_{|E_r^{(k)}}|_{W_r^{(k)}},$$

$$|\check{w}^{(k)}_{|E_1^{(k)}}|_{W_1^{(k)}} := \left\|\!\left\|\check{w}^{(k)}_{|E_1^{(k)}}\right\|\!\right\|_e^2 + \sum_{i^2=1}^{M_2^{(k)}-1} |c^w_{(1,i^2+1)} - c^w_{(1,i^2)}|^2,$$

$$|\check{w}^{(k)}_{|E_2^{(k)}}|_{W_2^{(k)}} := \left\|\!\left\|\check{w}^{(k)}_{|E_2^{(k)}}\right\|\!\right\|_e^2 + \sum_{i^2=1}^{M_2^{(k)}-1} |c^w_{(M_1,i^2+1)} - c^w_{(M_1,i^2)}|^2, \qquad (62)$$

$$|\check{w}^{(k)}_{|E_3^{(k)}}|_{W_3^{(k)}} := \left\|\!\left\|\check{w}^{(k)}_{|E_3^{(k)}}\right\|\!\right\|_e^2 + \sum_{i^1=1}^{M_1^{(k)}-1} |c^w_{(i^1+1,1)} - c^w_{(i^1,1)}|^2,$$

$$|\check{w}^{(k)}_{|E_4^{(k)}}|_{W_4^{(k)}} := \left\|\!\left\|\check{w}^{(k)}_{|E_4^{(k)}}\right\|\!\right\|_e^2 + \sum_{i^1=1}^{M_1^{(k)}-1} |c^w_{(i^1+1,M_2)} - c^w_{(i^1,M_2)}|^2,$$

where $M_\iota^{(k)}$ denotes the number of basis functions on patch k in direction ι. Furthermore, we define $\left\|\!\left\|\check{w}^{(k)}_{|E_r^{(k)}}\right\|\!\right\|_e^2 := \|v\|_e$, where v are the values $(c_i^w)_{i \in \mathcal{I}(\check{E}_r^{(k)})}$ written as a vector.

Let $\check{z}^{(k)} \in V_h^{(k)}$ be the NURBS harmonic extension of $w^{(k)} = \{c_i^w\} \in W^{(k)}$, and $z^{(k)}$ its representation in the parameter domain. Additionally, let e be any edge of the parameter domain of $\Omega^{(k)}$. Due to the fact that

$$c_i^w = c_i^z \quad \forall i \in \mathcal{I}(\Gamma^{(k)}),$$

and denoting $\boldsymbol{C^{(k)}} = (c_i^z)_{i \in \mathcal{I}^{(k)}}$, we obtain $\left\| \check{w}_{|e}^{(k)} \right\|_e^2 = \left\| \boldsymbol{C}_{|e}^{(k)} \right\|_e^2 \leq c \left\| \boldsymbol{C}^{(k)} \right\|_{\nabla}^2$ by means of Theorem 6. From the definition of $\left\| \boldsymbol{C}^{(k)} \right\|_{\nabla}^2$ and the definition of $|\check{w}_{|E_r^{(k)}}^{(k)}|_{W_r^{(k)}}^2$, we get

$$|\check{w}_{|e}^{(k)}|_{W_r^{(k)}}^2 \leq c \left\| \boldsymbol{C}^{(k)} \right\|_{\nabla}^2. \tag{63}$$

Furthermore, we have

$$|\check{w}_{|e}^{(k)}|_{W_r^{(k)}}^2 \leq c \left\| \boldsymbol{C}^{(k)} \right\|_{\nabla}^2 \leq c|z^{(k)}|_{\nabla}^2 \leq c|z^{(k)}|_{H^1((0,1)^d)} \leq c|\check{z}^{(k)}|_{H^1(\Omega^{(k)})}.$$

Since $|\check{z}^{(k)}|_{H^1(\Omega^{(k)})} = |\mathcal{H}^{(k)}\left(\check{w}^{(k)}\right)|_{H^1(\Omega^{(k)})}^2 = s^{(k)}(\check{w}^{(k)}, \check{w}^{(k)})$, we arrive at the estimate

$$|\check{w}_{|e}^{(k)}|_{W_r^{(k)}}^2 \leq c\, s^{(k)}(\check{w}^{(k)}, \check{w}^{(k)}).$$

This estimates hold for all edges of $\Omega^{(k)}$. Hence, it follows that

$$|\check{w}^{(k)}|_{W^{(k)}}^2 \leq c^* s^{(k)}(\check{w}^{(k)}, \check{w}^{(k)}) \quad \forall \check{w} \in W^{(k)},$$

where the constant does not depend on h and H. This proofs the upper bound of the estimate, i.e., (55).

Let be $\check{w}^{(k)} \in W_{\Delta}^{(k)}$, $w^{(k)}$ its representation in the parameter domain, and $(c_i^w)_{i \in \mathcal{I}(\Gamma^{(k)})}$ its coefficient representation. We now apply the lifting operator $\boldsymbol{H}^{(k)}$ to $(c_i^w)_{i \in \mathcal{I}(\Gamma^{(k)})}$, and obtain a matrix with entries $\boldsymbol{H}^{(k)}(w^{(k)}) = (h_i^{(k)})_{i \in \mathcal{I}^{(k)}}$. These entries define a spline function

$$z^{(k)} := \sum_{i \in \mathcal{I}^{(k)}} c_i^z N_{i,p}^{(k)}$$

$$c_i^z := h_i^{(k)} \quad \forall i \in \mathcal{I}^{(k)}.$$

Now we obtain the estimate

$$\left\| \boldsymbol{H}^{(k)}(w^{(k)}) \right\|_{\nabla}^2 = |z^{(k)}|_{\nabla}^2 \geq c|z^{(k)}|_{H^1((0,1)^d)}^2 \geq c|\check{z}^{(k)}|_{H^1(\Omega^{(k)})}^2 \geq c|\mathcal{H}\left(\check{w}^{(k)}\right)|_{H^1(\Omega^{(k)})}^2,$$

where the last inequality holds, due to the face that the discrete NURBS harmonic extension minimizes the energy among functions with given boundary data $\check{w}$. The constant c does not depend on h or H.

Recalling the definition of $|\check{w}^{(k)}|^2_{W^{(k)}}$ and using Theorem 6, we arrive at the estimates

$$\left\| \boldsymbol{H}^{(k)}(w^{(k)}) \right\|^2_\nabla \leq c \left(1 + \log^2 M^{(k)} \right) \sum_{e \in \partial[0,1]^2} \left\| w^{(k)}_{|e} \right\|^2_e \leq c \left(1 + \log^2 M^{(k)} \right) |\check{w}^{(k)}|^2_{W^{(k)}}.$$

Due to the mesh regularity, we have $M^{(k)} \approx H^{(k)}/h^{(k)}$, and, hence, we obtain

$$s^{(k)}(\check{w}^{(k)}, \check{w}^{(k)}) = |\mathcal{H}\left(\check{z}^{(k)} \right)|^2_{H^1(\Omega^{(k)})} \leq c \left(1 + \log^2 \left(\frac{H^{(k)}}{h^{(k)}} \right) \right) |\check{w}^{(k)}|^2_{W^{(k)}},$$

which provides

$$c_* \leq c \max_{1 \leq k \leq N} \left(1 + \log^2 \left(\frac{H^{(k)}}{h^{(k)}} \right) \right),$$

where the constant c is again independent of h and H. $\qquad\square$

The next theorem provides a the corresponding estimates for the stiffness scaling.

Theorem 9. *Let the counting functions be chosen according to the stiffness scaling strategy. Assume that there exist two positive constants c_*, c^* and a boundary seminorm $|\cdot|_{W^{(k)}}$ on $W^{(k)}$, $k = 1, \ldots, N$, such that the three conditions of Theorem 7 hold. Moreover, we assume that it exits a constant c^*_{STIFF} such that*

$$|\check{w}^{(k)}|_{W^{(k)}} \leq c^*_{STIFF} s(\delta\check{w}^{(k)}, \delta\check{w}^{(k)}) \quad \forall \check{w}^{(k)} \in W^{(k)}_\Delta, \qquad (64)$$

where the coefficients of $\delta\check{w}^{(k)}$ are given by $c_i^{(k)}\delta_i^{(k)}$. Then the condition number of the preconditioned BDDC operator $M^{-1}_{BDDC}\hat{S}$ satisfies the bound

$$\kappa(M^{-1}_{BDDC}\hat{S}) \leq c(1 + c_*^{-1}c^* + c_*^{-1}c^*_{STIFF}) \qquad (65)$$

for some constant c which is independent of h and H.

Proof. See [12]. $\qquad\square$

According to [12], we apply a modified version of the stiffness scaling where we use one representative of the values $\delta_i^{(k)}$. This is reasonable, since these values are very similar on one patch $\delta_i^{(k)} \approx \delta_j^{(k)}$, which arises from the tensor product structure of B-Splines and the constant material value on a patch.

Lemma 4. *The bound (64) holds with*

$$c_{STIFF}^* \leq 1. \tag{66}$$

Hence, the condition number of the BDDC preconditioned system in the case of stiffness scaling is bounded by

$$\kappa(M_{BDDC}^{-1}\hat{S}) \leq C \max_{1 \leq k \leq N} \left(1 + \log^2 \left(\frac{H^{(k)}}{h^{(k)}} \right) \right), \tag{67}$$

where the constant C is independent of H and h.

Proof. The inequality

$$|\check{w}^{(k)}|_{W^{(k)}} \leq c_{STIFF}^* s(\delta \check{w}^{(k)}, \delta \check{w}^{(k)}) \quad \forall \check{w} \in W_\Delta^{(k)}$$

is equivalent to

$$|\delta^\dagger \check{w}^{(k)}|_{W^{(k)}} \leq c_{STIFF}^* s(\check{w}^{(k)}, \check{w}^{(k)}) \quad \forall \check{w} \in W_\Delta^{(k)}. \tag{68}$$

We have already proven that

$$|\check{w}^{(k)}|_{W^{(k)}} \leq c_{STIFF}^* s(\check{w}^{(k)}, \check{w}^{(k)}) \quad \forall \check{w} \in W_\Delta^{(k)} \subset W^{(k)}.$$

Hence, it is enough to show the inequality

$$|\delta^\dagger \check{w}^{(k)}|_{W^{(k)}} \leq c_{h,H} |\check{w}^{(k)}|_{W^{(k)}} \quad \forall \check{w} \in W_\Delta^{(k)},$$

where the constant $c_{h,H}$ may depend on $h^{(k)}$ and $H^{(k)}$. Recalling the definition of $|\delta^\dagger \check{w}^{(k)}|_{W^{(k)}}$ as the sum of $|\delta^\dagger \check{w}_{|E_r^{(k)}}^{(k)}|_{W_r^{(k)}}$, $r = 1, 2, 3, 4$, see (62), we have only to estimate the parts, e.g., $|\check{w}_{|E_1^{(k)}}^{(k)}|_{W_1^{(k)}}$. The other three terms follow analogously. From the fact that $\delta_i^{\dagger(k)} \leq 1$ and $\delta_i^{\dagger(k)} = \delta_{i+1}^{\dagger(k)}$

for all $k \in \{1, \ldots, N\}$ and $i \in \mathcal{I}(\check{E}_1^{(k)})$, c.f. Section 6.2. in [12] , it follows that

$$\left\|\left|\delta^\dagger \check{w}_{|E_1^{(k)}}^{(k)}\right|\right\|_e = \delta^\dagger \left\|\left|\check{w}_{|E_1^{(k)}}^{(k)}\right|\right\|_e \leq \left\|\left|\check{w}_{|E_1^{(k)}}^{(k)}\right|\right\|_e$$

and

$$\sum_{i^2=1}^{M_2^{(k)}-1} |\delta_{(1,i^2+1)}^\dagger c_{(1,i^2+1)}^w - \delta_{(1,i^2)}^\dagger c_{(1,i^2)}^w|^2 = \sum_{i^2=1}^{M_2^{(k)}-1} \delta_i^\dagger |c_{(1,i^2+1)}^w - c_{(1,i^2)}^w|^2$$

$$\leq \sum_{i^2=1}^{M_2^{(k)}-1} |c_{(1,i^2+1)}^w - c_{(1,i^2)}^w|^2.$$

These estimates provide the inequalities

$$|\delta^\dagger \check{w}_{|E_1^{(k)}}^{(k)}|_{W_1^{(k)}} \leq |\check{w}_{|E_1^{(k)}}^{(k)}|_{W_1^{(k)}}, \tag{69}$$

and, finally,

$$|\delta^\dagger \check{w}^{(k)}|_{W^{(k)}} \leq |\check{w}^{(k)}|_{W^{(k)}}. \tag{70}$$

This concludes the proof with $c_{STIFF}^* \leq 1$, and the desired condition number bound. $\qquad\square$

4 Implementation

Since $\boldsymbol{F}$ is symmetric and at least positive semi definite and positive definite on $\widetilde{U}$, we can solve the linear system $\boldsymbol{F}\boldsymbol{\lambda} = \boldsymbol{d}$ of the algebraic equations by means of the PCG algorithm, where we use M_{sD}^{-1} as preconditioner:

It is very expensive to build up the matrices $\boldsymbol{F}$ and M_{sD}^{-1}. Fortunately, the PCG algorithm only requires the application of the matrix on a vector. Hence, we want to present a way to efficiently apply $\boldsymbol{F}$ and M_{sD}^{-1} without calculating its matrix form. In order to do so, we need the following realization:

– Application of $\widetilde{S}^{-1} : \widetilde{W}^* \to \widetilde{W}$ and $S : W \to W^*$,

---

Algorithm 1 PCG method

$\boldsymbol{\lambda}_0$ given
$r_0 = \boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_0, \quad k = 0, \quad \beta_{-1} = 0$
repeat
$\quad s_k = M_{sD}^{-1} r_k$
$\quad \beta_{k-1} = \frac{(r_k, s_k)}{(r_{k-1}, s_{k-1})}$
$\quad p_k = s_k + \beta_{k-1} p_{k-1}$
$\quad \alpha_k = \frac{(r_k, s_k)}{(\boldsymbol{F}p_k, p_k)}$
$\quad \boldsymbol{\lambda}_{k+1} = r_k + \alpha_k p_k$
$\quad r_{k+1} = r_k - \alpha_k \boldsymbol{F} p_k$
$\quad k = k + 1$
until stopping criterion fulfilled

- Realization and a basis of $\widetilde{W} = \widetilde{W}_\Pi \oplus \prod \widetilde{W}_\Delta^{(k)}$,

- Representation of $w \in \widetilde{W}$ as $\{\boldsymbol{w}_\Pi, \{w_\Delta^{(k)}\}_k\}$,

- Representation of $f \in \widetilde{W}^*$ as $\{\boldsymbol{f}_\Pi, \{f_\Delta^{(k)}\}_k\}$,

- Application of $\widetilde{B}$ and $\widetilde{B}^T$.

4.1 Choosing a basis for $\widetilde{W}_\Pi$

The first step is to provide an appropriate space $\widetilde{W}_\Pi$ and a basis $\{\widetilde{\phi}_j\}_j^{n_\Pi}$ for $\widetilde{W}_\Pi$, where n_Π is the dimension of $\widetilde{W}_\Pi$, i.e., the number of primal variables. We request from the basis that it has to be nodal with respect to the primal variables, i.e.,

$$\psi_i(\widetilde{\phi}_j) = \delta_{i,j}, \quad \forall i, j \in \{1, \ldots, n_\Pi\}.$$

Additionally, we require that

$$\widetilde{\phi}_{j|\Omega^{(k)}} = 0 \quad \text{if } \psi_j \text{ is not associated to } \Omega^{(k)},$$

i.e., the basis has a local support in a certain sense.

There are many choices for the subspace $\widetilde{W}_\Pi$. Following the approach presented in [5], we will choose that one which is orthogonal to $\widetilde{W}_\Delta$ with respect to S. Such a subspace exits since $\widetilde{S}$ is SPD on $\widetilde{W}$. Hence, we can define $\widetilde{W}_\Pi := \widetilde{W}_\Delta^{\perp_S}$, i.e.,

$$\langle Sw_\Pi, w_\Delta \rangle = 0, \quad \forall w_\Pi \in \widetilde{W}_\Pi, w_\Delta \in \widetilde{W}_\Delta.$$

This choice, which will simplify the application of $\widetilde{S}^{-1}$ significantly, is known as *energy minimizing primal subspace* in the literature, cf. [5],[22].

Now we need to find a nodal local basis for this space. In order to do that, we define the constraint matrix $C^{(k)} : W^{(k)} \to \mathbb{R}^{n_\Pi^{(k)}}$ for each patch $\Omega^{(k)}$ which realizes the primal variables:

$$C^{(k)} : W^{(k)} \to \mathbb{R}^{n_\Pi^{(k)}}$$
$$(C^{(k)}v)_j = \psi_{i(k,j)}(v) \quad \forall v \in W \, \forall j \in \{1, \dots, n_\Pi^{(k)}\},$$

where $n_\Pi^{(k)}$ are the number of primal variables associated with $\Omega^{(k)}$ and $i(k,j)$ the global index of the j-th primal variable on $\Omega^{(k)}$. Note that a function $w_\Delta^{(k)} \in \widetilde{W}_\Delta^{(k)}$ is in the kernel of $C^{(k)}$, i.e., $C^{(k)}w_\Delta^{(k)} = 0$.

For each patch k, the basis functions $\{\widetilde{\phi}_j^{(k)}\}_{j=1}^{n_\Pi^{(k)}}$ of $\widetilde{W}_\Pi^{(k)}$ are the solution of the system

$$\begin{bmatrix} S^{(k)} & C^{(k)T} \\ C^{(k)} & 0 \end{bmatrix} \begin{bmatrix} \widetilde{\phi}_j^{(k)} \\ \widetilde{\boldsymbol{\mu}}_j^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{e}_j^{(k)} \end{bmatrix}, \quad \forall j \in \{1, \dots, n_\Pi^{(k)}\}, \qquad (71)$$

where $\boldsymbol{e}_j^{(k)} \in \mathbb{R}^{n_\Pi^{(k)}}$ is the j-th unit vector. The Lagrange multipliers $\widetilde{\boldsymbol{\mu}}_j^{(k)} \in \mathbb{R}^{n_\Pi^{(k)}}$ will become important later on. Equation (71) has a unique solution $\{\widetilde{\phi}_j^{(k)}, \widetilde{\boldsymbol{\mu}}_j^{(k)}\}$. Due to $\ker(S) \cap \widetilde{W} = \{0\}$ and $\ker(C^{(k)}) = W_\Delta^{(k)}$, it follows that $\ker(S^{(k)}) \cap \ker(C^{(k)}) = \{0\}$. Furthermore, Ψ is linearly independent, and, hence, $\ker(C^{(k)T}) = \{0\}$. Both properties provide the existence and uniqueness of the solution.

Due to the fact that building the Schur complement $S^{(k)}$ is not efficient, we use an equivalent formulation by means of $K^{(k)}$:

$$\begin{bmatrix} K_{BB}^{(k)} & K_{BI}^{(k)} & C^{(k)T} \\ K_{IB}^{(k)} & K_{II}^{(k)} & 0 \\ C^{(k)} & 0 & 0 \end{bmatrix} \begin{bmatrix} \widetilde{\phi}_j^{(k)} \\ \cdot \\ \widetilde{\boldsymbol{\mu}}_j^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{e}_j^{(k)} \end{bmatrix}. \qquad (72)$$

For each patch k, the LU factorization of this matrix is computed and stored.

4.2 Application of $\widetilde{S}^{-1}$

Assume that $f := \{\boldsymbol{f}_\Pi, \{f_\Delta^{(k)}\}\} \in \widetilde{W}^*$ is given. We are now looking for $w := \{\boldsymbol{w}_\Pi, \{w_\Delta^{(k)}\}\} \in \widetilde{W}$ with

$$w = S^{-1}f. \tag{73}$$

The idea to provide a formula for the application of $\widetilde{S}^{-1}$ is via a block LDL^T factorization of $\widetilde{S}$ with respect to the splitting of $\widetilde{W} = \widetilde{W}_\Pi \oplus \prod \widetilde{W}_\Delta^{(k)}$. Let $S_{\Pi\Pi}, S_{\Delta\Pi}, S_{\Pi\Delta}, S_{\Delta\Delta}$ be the restrictions of S to the corresponding subspaces, i.e.,

$$\langle S_{\Pi\Pi} v_\Pi, w_\Pi \rangle = \langle S v_\Pi, w_\Pi \rangle \qquad \text{for } v_\Pi, w_\Pi \in \widetilde{W}_\Pi,$$
$$\langle S_{\Pi\Delta} v_\Pi, w_D \rangle = \langle S v_\Pi, w_D \rangle \qquad \text{for } v_\Pi \in \widetilde{W}_\Pi, w_D \in \widetilde{W}_\Delta,$$
$$\langle S_{\Delta\Delta} v_D, w_D \rangle = \langle S v_D, w_D \rangle \qquad \text{for } v_D, w_D \in \widetilde{W}_\Delta,$$

and $S_{\Delta\Pi} = S_{\Pi\Delta}^T$. We note that $S_{\Delta\Delta}$ can be seen as a block diagonal operator, i.e., $S_{\Delta\Delta} = \mathrm{diag}(S_{\Delta\Delta}^{(k)})$. Based on this splitting, we have the block form

$$\widetilde{S} = \begin{bmatrix} S_{\Pi\Pi} & S_{\Pi\Delta} \\ S_{\Delta\Pi} & S_{\Delta\Delta} \end{bmatrix}.$$

A formal block LDL^T factorization of S yields

$$\widetilde{S}^{-1} = \begin{bmatrix} \boldsymbol{I} & 0 \\ -S_{\Delta\Delta}^{-1} S_{\Delta\Pi} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{T}_\Pi^{-1} & 0 \\ 0 & S_{\Delta\Delta}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & -S_{\Pi\Delta} S_{\Delta\Delta}^{-1} \\ 0 & \boldsymbol{I} \end{bmatrix}, \tag{74}$$
$$\boldsymbol{T}_\Pi^{-1} = S_{\Pi\Pi} - S_{\Pi\Delta} S_{\Delta\Delta}^{-1} S_{\Delta\Pi}. \tag{75}$$

Due to our special choice $\widetilde{W}_\Pi := \widetilde{W}_\Delta^{\perp_S}$, we have $S_{\Delta\Pi} = S_{\Pi\Delta} = 0$. This simplifies the expression (74) and (75). Hence, we obtain

$$\widetilde{S}^{-1} = \begin{bmatrix} S_{\Pi\Pi}^{-1} & 0 \\ 0 & S_{\Delta\Delta}^{-1} \end{bmatrix}.$$

Therefore, the application of $\widetilde{S}^{-1}$ reduces to an application of one global coarse problem involving $\boldsymbol{S}_{\Pi\Pi}^{-1}$ and N local problems involving $S_{\Delta\Delta}^{(k)}{}^{-1}$

$$\boldsymbol{w}_\Pi = \boldsymbol{S}_{\Pi\Pi}^{-1} \boldsymbol{f}_\Pi,$$
$$w_\Delta^{(k)} = S_{\Delta\Delta}^{(k)}{}^{-1} f_\Delta^{(k)} \quad \forall k = 1, \dots, N.$$

Application of $S_{\Delta\Delta}^{(k)}{}^{-1}$:

The application of $S_{\Delta\Delta}^{(k)}{}^{-1}$ corresponds to solving a local Neumann problem in the space $\widetilde{W}_\Delta$, i.e.,

$$S^{(k)}w^{(k)} = f_\Delta^{(k)},$$

with the constraint $C^{(k)}w^{(k)} = 0$.

This problem can be rewritten as a saddle point problem in the form

$$\begin{bmatrix} S^{(k)} & C^{(k)^T} \\ C^{(k)} & 0 \end{bmatrix} \begin{bmatrix} w^{(k)} \\ \cdot \end{bmatrix} = \begin{bmatrix} f_\Delta^{(k)} \\ 0 \end{bmatrix}.$$

The same method as used in (71) for rewriting this equation in terms of K applies here and the LU factorization of the matrix is already available.

Application of $\boldsymbol{S}_{\Pi\Pi}^{(k)}{}^{-1}$:

The matrix $\boldsymbol{S}_{\Pi\Pi}$ can be assembled from the patch local matrices $\boldsymbol{S}_{\Pi\Pi}^{(k)}$.

Let $\{\widetilde{\phi}_j^{(k)}\}_{j=1}^{n_\Pi^{(k)}}$ be the basis of $\widetilde{W}_\Pi^{(k)}$. In order to assemble $\boldsymbol{S}_{\Pi\Pi}^{(k)}$, in general, we have to compute

$$\left(\boldsymbol{S}_{\Pi\Pi}^{(k)}\right)_{i,j} = \left\langle S^{(k)}\widetilde{\phi}_i^{(k)}, \widetilde{\phi}_j^{(k)} \right\rangle, \quad i,j \in \{1,\ldots,n_\Pi^{(k)}\}.$$

The special choice of $\widetilde{W}_\Pi := \widetilde{W}_\Delta^{\perp_S}$ provides us with the following property

$$\left(\boldsymbol{S}_{\Pi\Pi}^{(k)}\right)_{i,j} = \left\langle S^{(k)}\widetilde{\phi}_i^{(k)}, \widetilde{\phi}_j^{(k)} \right\rangle = -\left\langle C^{(k)^T}\widetilde{\boldsymbol{\mu}}_i^{(k)}, \widetilde{\phi}_j^{(k)} \right\rangle = -\left\langle \widetilde{\boldsymbol{\mu}}_i^{(k)}, C^{(k)}\widetilde{\phi}_j^{(k)} \right\rangle$$

$$= -\left\langle \widetilde{\boldsymbol{\mu}}_i^{(k)}, \boldsymbol{e}_j^{(k)} \right\rangle = -\left(\widetilde{\boldsymbol{\mu}}_i^{(k)}\right)_j.$$

Therefore, we can reuse the Lagrange multipliers $\widetilde{\boldsymbol{\mu}}_i^{(k)}$ obtained in (72), and can assemble $\boldsymbol{S}_{\Pi\Pi}^{(k)}$ from them. Once $\boldsymbol{S}_{\Pi\Pi}$ is assembled, the LU factorization can be calculated and stored.

4.3   Summary of the algorithm for $F = \widetilde{B}S^{-1}\widetilde{B}^T$

Let us mention that the implementation of the embedding operator $\widetilde{I}$ and assembling operator $\widetilde{I}^T$ is explained in detail in [5] and is omitted here.

Algorithm 2 Algorithm for the calculation of $\nu = F\boldsymbol{\lambda}$ for given $\boldsymbol{\lambda} \in U$

Application of B^T : $\{f^{(k)}\}_{k=1}^N = B^T\boldsymbol{\lambda}$

Application of $\widetilde{I}^T$: $\{\boldsymbol{f}_\Pi, \{f_\Delta^{(k)}\}_{k=1}^N\} = \widetilde{I}^T\left(\{f^{(k)}\}_{k=1}^N\right)$

Application of S^{-1} :

  – $\boldsymbol{w}_\Pi = \boldsymbol{S}_{\Pi\Pi}^{-1}\boldsymbol{f}_\Pi$

  – $w_\Delta^{(k)} = S_{\Delta\Delta}^{(k)}{}^{-1}f_\Delta^{(k)}$ $\forall k = 1, \dots, N$

Application of $\widetilde{I}$: $\{w^{(k)}\}_{k=1}^N = \widetilde{I}\left(\{\boldsymbol{w}_\Pi, \{w_\Delta^{(k)}\}_{k=1}^N\}\right)$

Application of B : $\nu = B\left(\{w^{(k)}\}_{k=1}^N\right)$

4.4   Application of the preconditioner

The application of the preconditioner $M_{sD}^{-1} = B_D S B_D^T$ is basically the application of S:

$$S = \operatorname{diag}(S^{(k)}),$$
$$S^{(k)} = K_{BB}^{(k)} - K_{BI}^{(k)}(K_{II}^{(k)})^{-1}K_{IB}^{(k)}.$$

The calculation of $v^{(k)} = S^{(k)}w^{(k)}$ consists of 2 steps:

1. Solve: $K_{II}^{(k)}x^{(k)} = -K_{IB}^{(k)}w^{(k)}$ (*Dirichlet problem*).

2. $v^{(k)} = K_{BB}^{(k)}w^{(k)} + K_{BI}^{(k)}x^{(k)}$.

Again, a LU factorization of $K_{II}^{(k)}$ can be computed in advance and stored.

5   Numerical examples

We test the implemented IETI-DP algorithm for solving large scale systems arising from the IgA discretization of (1) on the domains illustrated in Figure 1. 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. Each subdomain has a diameter of H and an associated mesh size of h. The degree of the B-Splines is chosen as $p = 4$. In order to solve the linear system (29), a PCG algorithm with the scaled Dirichlet preconditioner (33) is performed. We use 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.

We use the C++ library G+SMO[1], for describing the geometry and performing the numerical tests.

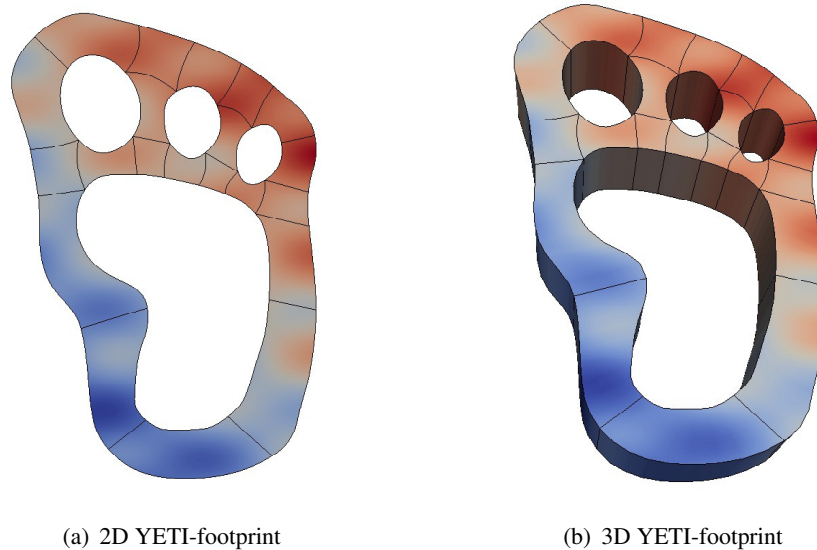(a) 2D YETI-footprint   (b) 3D YETI-footprint

Fig. 1. Illustration of the computational domain in 2D and 3D.

5.1 The case of homogeneous diffusion coefficient

We present numerical tests for problem (1) with globally constant diffusion coefficient α. The 2D results are summarized in Table 1, whereas the 3D results are presented in Table 2. The results confirm that the preconditioned systems with coefficient scaling as well as the stiffness scaling provide a quasi optimal condition number bound according to Theorem 8 and Theorem 9.

---

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

ALG. A		unprec. F		coeff. scal		stiffness scal.	
#dofs	H/h	κ	It.	κ	It.	κ	It.
2364	9	45	50	9	21	9	20
4728	13	73	46	11	22	11	22
11856	21	133	57	15	24	14	24
35712	37	265	68	18	25	18	25
ALG. C							
2364	9	28	44	1.8	11	1.8	11
4728	13	22	39	2	12	2	12
11856	21	18	39	2.4	14	2.4	14
35712	37	17	38	2.8	15	2.8	15

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

ALG. A		unprec. F		coeff. scal		stiffness scal.	
#dofs	H/h	κ	It.	κ	It.	κ	It.
7548	5	3254	393	63	33	63	33
14368	7	3059	356	86	37	86	37
38100	10	2170	317	196	45	196	46
142732	16	7218	397	467	64	468	65
ALG. B							
7548	5	2751	341	1.6	10	1.6	10
14368	7	2860	397	1.7	11	1.7	11
38100	10	1697	333	2.0	12	2.3	13
142732	16	1261	333	2.3	13	3.1	16

Table 2. 3D example with $p = 4$ homogeneous diffusion coefficient. Dependence of the condition number κ and the number It. of iterations on H/h for the unpreconditioned system and 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).

5.2   The case of jumping diffusion coefficient

We investigate numerical numerical examples with patchwise constant diffusion coefficient α, with a jumping pattern according to Figure 2. The values of the diffusion coefficient are 10^{-3} (blue) and 10^3 (red). The 2D results are summarized in Table 3 and the 3D results in Table 4. We observe a quasi optimal condition number bound which is clearly independent of the diffusion coefficient and its jumps across subdomain interfaces.
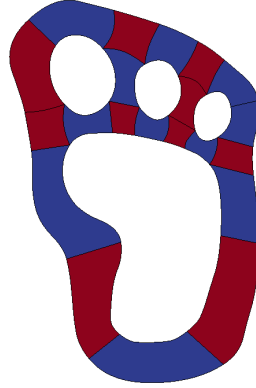
Fig. 2. Pattern of the jumping diffusion coefficient

ALG. A		unprec. F		coeff. scal		stiffness scal.	
#dofs	H/h	κ	It.	κ	It.	κ	It.
2364	9	1.4e07	317	5.6	13	5.3	13
4728	13	1.5e07	297	7.0	13	6.4	13
11856	21	2.4e07	397	8.7	15	7.8	13
35712	37	4.0e07	434	10.6	16	9.3	14
ALG. C							
2364	9	1.5e07	261	1.8	7	1.7	7
4728	13	1.1e07	267	2.2	8	2	7
11856	21	9.8e06	291	2.6	8	2.3	8
35712	37	9.0e06	310	3.0	10	2.7	10

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

5.3 Dependence on p

We want to examine the dependence of the condition number on the B-Spline degree p, although the presented theory in Section 3.4 does not cover the dependence of IETI-DP preconditioned system on p. We note that in our implementation the degree elevation yields an increase in the multiplicity of the knots within each step, resulting in C^1 smoothness on each patch. The computational domain is chosen as the 2D YETI-footprint presented in Figure 1 and the diffusion coefficient is chosen to

ALG. A		unprec. F		coeff. scal		stiffness scal.	
#dofs	H/h	κ	It.	κ	It.	κ	It.
7548	5	>1.e16	>1000	47	20	47	18
14368	7	>1.e16	>1000	69	20	65	19
38100	10	>1.e16	>1000	165	32	152	29
142732	16	>1.e16	>1000	405	38	368	34
ALG. B							
7548	5	>1.e16	>1000	1.7	7	1.6	7
14368	7	>1.e16	>1000	1.8	7	1.7	7
38100	10	>1.e16	>1000	2.1	8	2.3	8
142732	16	>1.e16	>1000	4.4	9	3.2	11

Table 4. 3D example with $p = 4$ jumping diffusion coefficient. Dependence of the condition number κ and the number It. of iterations on H/h for the unpreconditioned system and 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).

be globally constant. The results are summarized in Table 5, where we observe a possibly logarithmic dependence of the condition number on the polynomial degree in case of the coefficient scaling as well as of the above mention version of the stiffness scaling. The numerical experiments depict a linear dependence in case of the regular stiffness scaling (3.3), see Figure 3.

ALG. C		unprec. F		coeff. scal		stiffness scal.		stiff. scal. modif.	
#dofs	degree	κ	It.	κ	It.	κ	It.	κ	It.
800	2	3.24	16	1.65	8	1.64	8	1.63	8
2364	3	8.08	28	1.71	10	1.69	10	1.7	10
4728	4	24.2	51	1.83	11	1.88	12	1.83	11
7892	5	82.8	86	2.03	13	2.16	12	2.01	12
11856	6	296	140	2.23	13	2.42	14	2.18	13
16620	7	1082	230	2.41	14	2.66	14	2.34	14
22184	8	4021	371	2.57	15	2.88	15	2.49	15
28548	9	15034	594	2.72	15	3.09	16	2.63	15
35712	10	56773	968	2.87	16	3.28	16	2.75	15

Table 5. 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 unpreconditioned system and preconditioned system with coefficient and stiffness scaling. Choice of primal variables: vertex evaluation and edge averages.
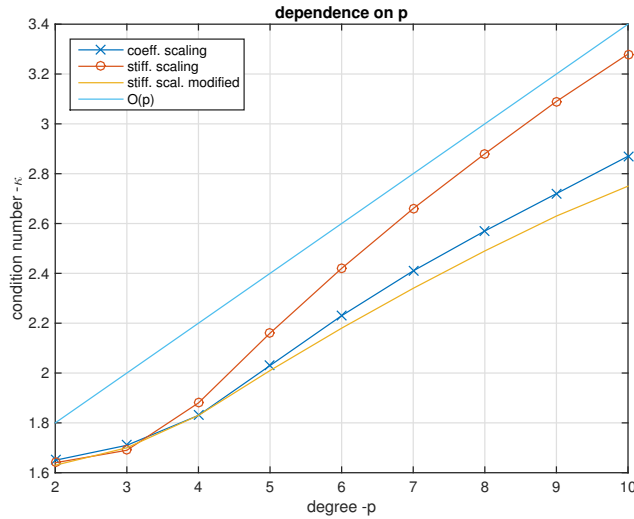
Fig. 3. 2D example with fixed initial mesh and homogeneous diffusion coefficient. Condition number κ as a function of polynomial degree. Choice of primal variables: vertex evaluation and edge averages.

5.4 Performance

The algorithm was tested on a Desktop PC with an Intel(R) Xeon(R) CPU E5-1650 v2 @ 3.50GHz and 16 GB main memory. As already mentioned at the beginning of this section, we used the open source library G+SMO for the materialization of the code. Moreover, we make use of the Sparse-LU factorization of the open source library "Eigen"[2] for the local solvers. The timings presented in Table 6 are obtained from a sequential implementation of the code. We choose the same setting as presented in Table 3 with Algorithm C. However, we do one more refinement steps and obtain 121824 total degrees of freedom, 1692 Lagrange multipliers, and on each patch approximate 4900 local degrees of freedom. We select a run with coefficient scaling and obtain a condition number of $\kappa = 3.53$ and 11 iterations.

We remark that about $90\,\%$ of the runtime is used for the assembling part of the program including the Schur complement computations. In more detail, most of the time is spent for calculating the LU-factorizations of

---

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

the local matrices. This indicates the importance of replacing the direct solver with inexact solvers on each patch, see, e.g., [23],[24]. Furthermore, we want to note, that especially in 3D, an additional bottleneck is the memory demand of the direct solvers.

	Wall-clock time	relative time in %
Preparing the bookkeeping	0.011 s	0.03
Assembling all patch local $\boldsymbol{K}^{(k)}$	6.2 s	15.42
Partitioning w.r.t. B and I	0.087 s	0.22
Assembling C	0.016 s	0.04
Calculating LU -fact. of $\boldsymbol{K}_{II}$	15 s	37.31
Calculating LU -fact. of $\begin{bmatrix} K^{(k)} & C^{(k)T} \\ C^{(k)} & 0 \end{bmatrix}$	15 s	37.31
Assembling and LU-fact of $\boldsymbol{S}_{\Pi\Pi}$	0.46 s	1.14
Assemble rhs.	0.094 s	0.23
Total assembling	37 s	92.04
One PCG iteration	0.22 s	-
Solving the system	2.5 s	6.22
Calculating the solution $\boldsymbol{u}$	0.5 s	1.24
Total spent time	40.2 s	100.00

Table 6. Timings of the 2D example with coefficient scaling, Algorithm C and one more refinement compared to Table 3. The discrete problem consists of 121824 total degrees of freedom, 1692 Lagrange multipliers, and on each patch approximate 4900 local degrees of freedom. Middle column presents the absolute spent time and right column the relative one.

6 Conclusions

We have derived condition number estimates for the IETI-DP method and extended the existent theory to domains which cannot be represented by a single geometrical mapping. Due to the fact, that we only considered open knot vectors, we could identify basis function on the interface and on the interior. This assumption implies that the discrete solution is only C^0 smooth across patch interfaces. However, under this assumption, we were able to find an improved condition number bound of the IETI-DP method using the Dirichlet preconditioner with stiffness scaling. Numerical examples with two and three dimensional domains, different choices of primal variables and different scaling methods confirmed the theoretical results presented in Section 3. Additionally, we investigated the

B-Spline degree dependence of the preconditioned system. Moreover, the numerical results indicate the robustness with respect to jumping diffusion coefficients across the interfaces. We have obtained similar numerical results for solving multipatch discontinuous Galerkin (dG) IgA schemes, proposed and investigated in [17], by means of IETI-DP methods following the approach developed by [25] for a composite finite element and dG method. Our results in the IETI-DP versions for solving multipatch dG IgA equations will be published in a forthcoming paper.

References

1. 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.

2. 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.

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

4. 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.

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

6. A. Toselli and O. B. Widlund. *Domain decomposition methods – algorithms and theory.* Berlin: Springer, 2005.

7. T. Mathew. *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations (Lecture Notes in Computational Science and Engineering)*, volume 61. Springer Publishing Company, Incorporated, 1 edition, 2008.

8. 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.

9. 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.

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

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

12. 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.

13. J. Mandel, C. R. Dohrmann, and R. Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. *Appl. Numer. Math.*, 54(2):167–193, 2005.

14. 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.

15. 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.

16. U. Langer and S. E. Moore. Discontinuous Galerkin isogeometric analysis of elliptic PDEs on surfaces. NFN Technical Report 12, Johannes Kepler University Linz, NFN Geometry and Simulation, Linz, 2014. also available at `arxiv.org/abs/1402.1185` and accepted for publication in the DD22 proceedings.
17. 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.
18. L. L. Schumaker. *Spline functions: basic theory.* Cambridge: Cambridge University Press, 2007.
19. A. Tagliabue, L. Dedé, and A. Quarteroni. Isogeometric analysis and error estimates for high order partial differential equations in fluid dynamics. *Computers & Fluids*, 102:277–303, 2014.
20. A. Klawonn and O. B. Widlund. Dual-primal FETI methods for linear elasticity. *Commun. Pure Appl. Math.*, 59(11):1523–1572, 2006.
21. L. Beirão da Veiga, C. Chinosi, C. Lovadina, and L. F. Pavarino. Robust BDDC preconditioners for Reissner-Mindlin plate bending problems and MITC elements. *SIAM J. Numer. Anal.*, 47(6):4214–4238, 2010.
22. C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, 25(1):246–258, 2003.
23. J. Li and O. B. Widlund. On the use of inexact subdomain solvers for BDDC algorithms. *Comput. Methods Appl. Mech. Eng.*, 196(8):1415–1428, 2007.
24. A. Klawonn and O. Rheinbach. Inexact FETI-DP methods. *Int. J. Numer. Methods Eng.*, 69(2):284–307, 2007.
25. 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.