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

IETI - Isogeometric Tearing and Interconnecting

S.K. Kleiss, C. Pechstein, B. Jüttler, S. Tomar

RICAM-Report 2012-01

IETI - Isogeometric Tearing and Interconnecting

Stefan K. Kleiss*, Clemens Pechstein[b,*], Bert Jüttler[c], Satyendra Tomar[a]

[a] *Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Altenberger Straße 69, A-4040 Linz, Austria*
[b] *Institute of Computational Mathematics, Johannes Kepler University Linz, Altenberger Straße 69, A-4040 Linz, Austria*
[c] *Institute of Applied Geometry, Johannes Kepler University Linz, Altenberger Straße 69, A-4040 Linz, Austria*

Abstract

Finite Element Tearing and Interconnecting (FETI) methods are a powerful approach to designing solvers for large-scale problems in computational mechanics. The numerical simulation problem is subdivided into a number of independent sub-problems, which are then coupled in appropriate ways. NURBS- (Non-Uniform Rational B-spline) based isogeometric analysis (IGA) applied to complex geometries requires to represent the computational domain as a collection of several NURBS geometries. Since there is a natural decomposition of the computational domain into several subdomains, NURBS-based IGA is particularly well suited for using FETI methods.

This paper proposes the new IsogEometric Tearing and Interconnecting (IETI) method, which combines the advanced solver design of FETI with the exact geometry representation of IGA. We describe the IETI framework for two classes of simple model problems (Poisson and linearized elasticity) and discuss the coupling of the subdomains along interfaces (both for matching interfaces and for interfaces with T-joints, i.e. hanging nodes). Special attention is paid to the construction of a suitable preconditioner for the iterative linear solver used for the interface problem. We report several computational experiments to demonstrate the performance of the proposed IETI method.

Keywords: Isogeometric analysis, NURBS, domain decomposition, FETI

1. Introduction

Isogeometric Analysis (IGA), introduced by Hughes et al. [20], is a promising concept that establishes a close link between the technologies of CAD (computer aided design) and numerical simulation via finite element analysis (FEA). In the IGA framework, the same function spaces, which are used for the geometric representation of the computational domain, are used for the approximation (e.g. Galerkin) of the problem unknowns. There are several computational geometry technologies that could serve as a basis for IGA. However, Non-Uniform Rational B-Splines (NURBS) are the most widely used and well established computational technique in CAD, which we shall persue here. The main advantage consists in the fact that an *exact* representation of the computational domain is available at all times during the (possibly adaptive) simulation process. The need for geometric approximation processes, known as mesh generation, is thus eliminated.

Despite its short history, IGA has already attracted enormous interest which is documented by a substantial number of publications. IGA has been successfully applied to various simulation problems, including linear elasticity, simulation in electromagnetics, and flow simulations, e.g., [7, 15, 39]. It was shown that IGA not only eliminates the geometrical errors introduced by the approximation of the physical domain by a computational mesh but also significantly reduces the number of unknowns (when compared to standard FEA) needed to achieve a certain accuracy in practice.

On the theoretical side, various issues such as error estimates, convergence rates, stability issues, and numerical quadrature rules have been analyzed thoroughly [2, 5, 21, 47]. Many of these topics are already well understood and it was demonstrated that most of the theoretical background of FEA can be adapted and extended to cover the IGA framework. A notable exception, which is the subject of several on-going studies, is the search for generalizations of tensor-product spline spaces providing the possibility of local refinement. This is needed for adaptive simulation methods. Several concepts are currently being explored, including T-splines, hierarchical splines, and polynomial splines over hierarchical T-meshes [3, 6, 13, 28, 31, 38, 51].

Another new problem, which is located right at the crossroads between numerical simulation and geometric design, is the challenge of domain parameterization for IGA: Given a boundary representation of the computational domain by a collection of NURBS patches, extend it to a volume parameterization that is suitable for IGA. Several methods for designing such parameterizations and for creating them from boundary data exist [1, 8, 19, 37, 52]. For the numerical simulation in real-world applications it

*Corresponding author, Tel.: +43 732 2468 5229, Fax: +43 732 2468 5212

will be essential to create and to use multi-patch parameterizations [36], where the computational domain is represented by several NURBS volumes. These structures, which combine a highly regular structure on subdomains with topological flexibility to describe complex geometries, open new perspectives for the design of solvers in numerical simulation, which will be explored in this paper.

To do so, we will extend the *finite element tearing and interconnecting (FETI)* method to the isogeometric context. FETI methods are powerful solvers for large-scale finite element systems in computational mechanics. They were introduced by Farhat and Roux [16] and belong to the class of iterative substructuring methods (also called non-overlapping domain decomposition methods), see e.g. [49]. The computational domain is subdivided into non-overlapping subdomains, and each subdomain gets its own set of equations derived from the global equation. To ensure the equivalence to the global equation, continuity conditions are introduced at the interfaces between subdomains using Lagrangian multipliers.

In this paper, we combine the ideas of IGA and FETI and propose a new method called *isogeometric tearing and interconnecting*. Following the nomenclature of FETI and BETI (boundary element tearing and interconnecting) methods, we abbreviate the proposed method as *IETI* (to be pronounced ['jɛtɪ], like Yeti). The FETI method, however, is not only a coupling method but provides a powerful solver design. By (carefully) eliminating the original variables from the resulting saddle point problem, one obtains a symmetric and positive-definite system in the Lagrangian multipliers (i.e. only on the interface). The solution of the original problem can be easily computed from the solution of this interface problem. For generalizations to boundary element discretizations see e.g. [29, 30]. For generalizations to spectral element discretizations see e.g. [23].

Since the number of Lagrange multipliers is typically large, the interface problem is usually solved iteratively by FETI preconditioned conjugate gradients [43]. Suitable preconditioners have been proposed in [16, 25, 26]. The analyses in [26, 34] show that under suitable conditions the condition number of the preconditioned system is bounded by $C(1+\log(H/h)^\gamma)$, $\gamma \leq 3$, where H is the subdomain diameter and h is the mesh size. This results in quasi-optimal complexity of the overall method. We mention that the classical FETI method involves the solution of a coarse system. An efficient alternative is the dual-primal FETI (FETI-DP) method, see [17, 24, 27, 35], which is followed in this paper.

The remainder of this paper is organized as follows: In Section 2, we define the model problems considered herein and recall the definition of NURBS geometry mappings. In Section 3, we describe multi-patch NURBS mappings and multi-patch NURBS discretizations. In Section 4, we formulate the IETI method and investigate the solver and preconditioner aspects. Local refinement options which are introduced by the IETI method are described in Section 5. In Section 6, the performance of the IETI method is demonstrated in numerical examples. Finally, a conclusion is drawn in Section 7.

2. Preliminaries

We first briefly define the model problems considered in this paper, and then recall the definitions of B-spline basis functions and NURBS geometry mappings.

2.1. The model problems

Let $\Omega \subset \mathbb{R}^2$ be an open, bounded and connected Lipschitz domain with boundary $\partial\Omega$. Let the boundary $\partial\Omega$ be composed of two disjoint sets, the Dirichlet boundary Γ_D and the Neumann boundary Γ_N. Let $\vec{n}$ denote the outer unit normal vector to $\partial\Omega$, and let f, g_D and g_N be given functions. We shall consider two model problems:

(I) The scalar diffusion problem:
Find the scalar function $u : \overline{\Omega} \to \mathbb{R}$ such that

$$\begin{aligned} -\operatorname{div}(\alpha\nabla u) &= f && \text{in } \Omega, \\ u &= g_D && \text{on } \Gamma_D, \\ \alpha\frac{\partial u}{\partial \vec{n}} &= g_N && \text{on } \Gamma_N. \end{aligned}$$

(II) The linearized elasticity problem:
Find the displacement field $u : \overline{\Omega} \to \mathbb{R}^2$ such that

$$\begin{aligned} -\operatorname{div}(\sigma(u)) &= f && \text{in } \Omega, \\ u &= g_D && \text{on } \Gamma_D, \\ \sigma(u)\vec{n} &= g_N && \text{on } \Gamma_N. \end{aligned}$$

In problem (I), α denotes the diffusion coefficient. In problem (II), $\sigma(u) = C\varepsilon(u)$, where C is the fourth-order stiffness tensor and $\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T)$ is the linearized strain tensor.

The *variational forms* of these problems can be written as follows:

Find $u \in V_g$ such that

$$a(u,v) \;=\; f(v) \quad \forall v \in V_0. \tag{1}$$

The space $V_0 \subset (H^1(\Omega))^\beta$ contains test functions which vanish at the Dirichlet boundary. In problem (I), we have $\beta = 1$, and in problem (II), $\beta = 2$. The set $V_g \subset (H^1(\Omega))^\beta$ consists of functions which fulfill the Dirichlet boundary conditions on Γ_D.

We assume that the problem data is such that the bilinear form $a(\cdot,\cdot)$ is bounded, symmetric and positive definite, and that $f(\cdot)$ is a bounded linear functional. Then, the problem (1) can be reformulated as the following minimization problem:

Find u such that

$$u \;=\; \operatorname*{arg\,min}_{v \in V_g} \tfrac{1}{2}a(v,v) - f(v). \tag{2}$$

2.2. NURBS geometry mappings

To provide an overview, and to introduce notations, we briefly recall the definition of B-spline basis functions and NURBS mappings. Let p be a non-negative *degree* and let $s = (s_1, \dots, s_m)$ be a *knot vector* with $s_k \le s_{k+1}$ for all k. We consider only *open knot vectors*, i.e. knot vectors s where the multiplicity of a knot is at most p, except for the first and last knot which we require to have multiplicity $p+1$. For simplicity, we assume that $s_1 = 0$ and $s_m = 1$, which can be easily achieved by a suitable scaling. The $n_s = m - p - 1$ univariate *B-spline basis functions* $B^s_{k,p} : (0,1) \to \mathbb{R}$, $k = 1, \dots, n_s$, are defined recursively as follows:

$$\begin{aligned} B^s_{k,0}(\xi) &= \begin{cases} 1 & \text{for} \quad s_k \le \xi < s_{k+1} \\ 0 & \text{else} \end{cases} \\ B^s_{k,p}(\xi) &= \frac{\xi - s_k}{s_{k+p} - s_k} B^s_{k,p-1}(\xi) \\ &\quad + \frac{s_{k+p+1} - \xi}{s_{k+p+1} - s_{k+1}} B^s_{k+1,p-1}(\xi). \end{aligned}$$

Whenever a zero denominator appears in the definition above, the corresponding function $B^s_{k,p}$ is zero, and the whole term is considered to be zero. For open knot vectors, the first and last basis function are interpolatory at the first and the last knot, respectively.

Let $\{B^s_{k,p}\}_{k=1}^{n_s}$ and $\{B^t_{\ell,q}\}_{\ell=1}^{n_t}$ be two families of B-spline basis functions defined by the degrees p and q, and the open knot vectors

$$s = (s_1, \dots, s_{n_s+p+1}), \; t = (t_1, \dots, t_{n_t+q+1}),$$

respectively. We denote the set of all double-indices (k, ℓ) of NURBS basis functions by

$$\mathcal{R} = \{(k,\ell) : \; k \in \{1, \dots, n_s\}, \ell \in \{1, \dots, n_t\}\}.$$

Let $w_{(k,\ell)}$, $(k,\ell) \in \mathcal{R}$, be positive *weights*. We define the *bivariate NURBS basis functions* as follows:

$$R_{(k,\ell)}(\xi_1, \xi_2) = \frac{R^N_{(k,\ell)}(\xi_1, \xi_2)}{R^D(\xi_1, \xi_2)}$$

where the numerator $R^N_{(k,\ell)}$ and the denominator R^D are given by

$$\begin{aligned} R^N_{(k,\ell)}(\xi_1, \xi_2) &= B^s_{k,p}(\xi_1) \, B^t_{\ell,q}(\xi_2) \, w_{(k,\ell)}, \text{ and} \\ R^D(\xi_1, \xi_2) &= \sum_{(k,\ell) \in \mathcal{R}} B^s_{k,p}(\xi_1) \, B^t_{\ell,q}(\xi_2) \, w_{(k,\ell)} \\ &= \sum_{(k,\ell) \in \mathcal{R}} R^N_{(k,\ell)}(\xi_1, \xi_2). \end{aligned}$$

Given a *control net* of *control points* $P_{(k,\ell)} \in \mathbb{R}^2$, where again $(k,\ell) \in \mathcal{R}$, the two-dimensional *NURBS-surface* $G : Q \to \Omega$ is defined by

$$G(\xi_1, \xi_2) = \sum_{(k,\ell) \in \mathcal{R}} R_{(k,\ell)}(\xi_1, \xi_2) \, P_{(k,\ell)}, \tag{3}$$

where $Q = (0,1)^2$. We refer to Q as the *parameter domain* and to $\Omega = G(Q)$ as the *physical domain*.

For a detailed discussion of NURBS mappings and B-spline basis functions, and the properties of these functions we refer e.g. to [4, 9, 10, 20] and the references therein. Note that NURBS basis functions of degree p are, in general, globally C^{p-1}-continuous. However, in the presence of multiple knots, the continuity reduces according to the multiplicity. If a knot appears i times, the continuity of a NURBS basis function of degree p at that knot is C^{p-i}.

For our purposes, we assume that the geometry mapping is continuous and bijective (i.e. not self-penetrating), which are natural assumptions for CAD-applications.

For better readability, we collapse the double index (k,ℓ) into one multi-index $\boldsymbol{k}$ in the following. Hence, instead of (3), we write

$$G(\xi_1, \xi_2) = \sum_{\boldsymbol{k} \in \mathcal{R}} R_{\boldsymbol{k}}(\xi_1, \xi_2) \; P_{\boldsymbol{k}}.$$

3. Multi-patch geometry mappings

3.1. Single-patch NURBS discretization

In isogeometric analysis, the NURBS basis functions of the geometry mapping are also used to represent the discrete solution. The scalar H^1-conforming finite element space in a single-patch setting is the following:

$$V_h = \text{span}\{ \; \widehat{R}_{\boldsymbol{k}} \; \}_{\boldsymbol{k} \in \mathcal{R}} \subset H^1(\Omega), \tag{4}$$

where

$$\widehat{R}_{\boldsymbol{k}} = R_{\boldsymbol{k}} \circ G^{-1}. \tag{5}$$

For problems of type (II), we need to use the corresponding vector-valued space $V_h \subset H^1(\Omega)^2$. For simplicity, the following notation is only oriented on the scalar case.

We assume that the prescribed Dirichlet data g_D is such that there exists $\widehat{g} \in V_h : \; g_D = \widehat{g}|_{\Gamma_D}$ (otherwise, we project the Dirichlet data g_D to V_h). We define

$$\begin{aligned} V_{0h} &= \{v \in V_h : \; v|_{\Gamma_D} = 0\} \text{ and} \\ V_{gh} &= \widehat{g} + V_{0h} = \{v \in V_h : \; v|_{\Gamma_D} = g_D\}. \end{aligned}$$

A function $u_h \in V_h$ is represented as

$$u_h(x) = \sum_{\boldsymbol{k} \in \mathcal{R}} u_{\boldsymbol{k}} \; \widehat{R}_{\boldsymbol{k}}(x).$$

The real-valued coefficients $u_{\boldsymbol{k}}$ are referred to as *control variables* or *degrees of freedom (DOF)*.

3.2. Multi-patch NURBS discretization

Assume that the physical domain $\Omega \subset \mathbb{R}^2$ is represented by N single-patch NURBS geometry mappings $G^{(i)}$, $i = 1, \dots, N$, each of which maps the parameter domain $Q = (0,1)^2$ to an open physical subdomain

$$\Omega^{(i)} = G^{(i)}(Q) \subset \Omega, \; i = 1, \dots, N,$$

such that

$$\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega}^{(i)} \text{ and } \Omega^{(i)} \cap \Omega^{(j)} = \emptyset \text{ for } i \neq j.$$

We use the superscript (i) to indicate that knot vectors, degrees, NURBS basis functions, index sets, DOF, etc. are associated with a mapping $G^{(i)}$. For example, we denote the set of NURBS basis functions of the geometry mapping $G^{(i)}$ by $\{R_{\boldsymbol{k}}^{(i)}\}_{\boldsymbol{k}\in\mathcal{R}^{(i)}}$. For each subdomain $\Omega^{(i)}$, the local function space (for $\beta = 1$) is defined analogously to (4) and (5) as

$$V_h^{(i)} = \text{span}\{\ \widehat{R}_{\boldsymbol{k}}^{(i)}\ \}_{\boldsymbol{k}\in\mathcal{R}^{(i)}} \subset H^1(\Omega^{(i)}),$$

where

$$\widehat{R}_{\boldsymbol{k}}^{(i)} = R_{\boldsymbol{k}}^{(i)} \circ {G^{(i)}}^{-1}. \tag{6}$$

We denote the space of functions that are locally in $V_h^{(i)}$ by

$$\begin{aligned} \Pi V_h =& \{v \in L^2(\Omega)^\beta : v|_{\Omega^{(i)}} \in V_h^{(i)}, \forall\, i = 1, \dots, N\} \\ \cong& \prod_{i=1}^{N} V_h^{(i)}. \end{aligned} \tag{7}$$

As mentioned in Section 2.1, we have $\beta = 1$ for problem (I), and $\beta = 2$ for problem (II). Functions in ΠV_h are not necessarily continuous. We choose the following subsets of continuous functions in ΠV_h:

$$\begin{aligned} V_h =& \Pi V_h \cap C(\Omega)^\beta, \\ V_{0h} =& \{v \in V_h :\ v|_{\Gamma_D} = 0\}, \\ V_{gh} =& \widehat{g} + V_{0h}. \end{aligned} \tag{8}$$

A function $u_h \in V_h$ is represented subdomain-wise by:

$$u_h\big|_{\Omega^{(i)}} = \sum_{\boldsymbol{k}\in\mathcal{R}^{(i)}} u_{\boldsymbol{k}}^{(i)}\ \widehat{R}_{\boldsymbol{k}}^{(i)}. \tag{9}$$

We denote the *interface* of two subdomains $\Omega^{(i)}$ and $\Omega^{(j)}$ (see Fig. 1) by

$$\Gamma^{(i,j)} = \partial\Omega^{(i)} \cap \partial\Omega^{(j)}.$$

We collect the index-tupels of all interfaces that are non-empty

$$\mathcal{C}_\Gamma = \{(i,j) :\ \Gamma^{(i,j)} \neq \emptyset\}, \tag{10}$$

with $i, j \in \{1, \dots, N\}$. For $(i,j) \in \mathcal{C}_\Gamma$, we call $\Gamma^{(i,j)}$ a *subdomain vertex* (or *vertex* for brevity) if it consists of a single point, otherwise we call it an *edge*. For $(i,j) \in \mathcal{C}_\Gamma$, we collect the indices of those basis functions in $\Omega^{(i)}$ whose support intersects the interface $\Gamma^{(i,j)}$:

$$\mathcal{B}(i,j) \ = \ \{\boldsymbol{k} \in \mathcal{R}^{(i)} :\ \text{supp}\, \widehat{R}_{\boldsymbol{k}}^{(i)} \cap \Gamma^{(i,j)} \neq \emptyset\}. \tag{11}$$

If $\boldsymbol{k} \in \mathcal{B}(i,j)$, we say that the DOF $u_{\boldsymbol{k}}^{(i)}$ is associated with the interface $\Gamma^{(i,j)}$. In case of a subdomain vertex, we have $\#\mathcal{B}(i,j) = \#\mathcal{B}(j,i) = 1$, where $\#$ indicates the cardinality.

Let $\Gamma^{(i,j)}$ be an edge. We say that the subdomains $\Omega^{(i)}$ and $\Omega^{(j)}$ are *fully matching*, if the following two conditions are fulfilled (see Fig. 1 for an illustration):

(i) The interface $\Gamma^{(i,j)}$ is the image of an entire edge of the respective parameter domains.

(ii) For each index $\boldsymbol{k} \in \mathcal{B}(i,j)$, there must be a unique index $\boldsymbol{\ell} \in \mathcal{B}(j,i)$, such that

$$\widehat{R}_{\boldsymbol{k}}^{(i)}|_{\Gamma^{(i,j)}} = \widehat{R}_{\boldsymbol{\ell}}^{(j)}|_{\Gamma^{(i,j)}}. \tag{12}$$

This is the case, if the two knot vectors are affinely related and the corresponding weights and degrees are equal.

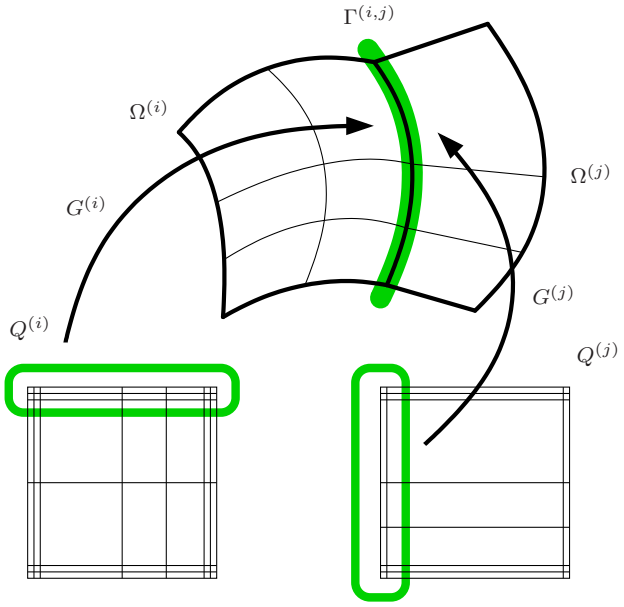


Figure 1: Fully matching subdomains $\Omega^{(i)}$ and $\Omega^{(j)}$:
All weights equal to 1,
$p^{(i)} = 2$, $s^{(i)} = \{0,0,0,0.5,0.75,1,1,1\}$,
$q^{(i)} = 2$, $t^{(i)} = \{0,0,0,0.5,1,1,1\}$,
$p^{(j)} = 1$, $s^{(j)} = \{0,0,1,1\}$,
$q^{(j)} = 2$, $t^{(j)} = \{0,0,0,0.25,0.5,1,1,1\}$.

For an illustration of two fully matching subdomains, see Fig. 1: The interface $\Gamma^{(i,j)}$ is the image of the entire northern edge of $Q^{(i)}$ under the mapping $G^{(i)}$, and the image of the entire western edge of $Q^{(j)}$ under $G^{(j)}$. Furthermore, $p^{(i)} = q^{(j)} = 2$. The knot vectors $s^{(i)}$ and $t^{(j)}$ are not equal, but due to the way they are mapped to $\Gamma^{(i,j)}$, condition (ii) is fulfilled. Hence, $\Omega^{(i)}$ and $\Omega^{(j)}$ are fully matching.

The tensor-product structure of the NURBS basis functions is very convenient for collecting/identifying the DOF associated with an interface, i.e. the index-set $\mathcal{B}(i,j)$. In particular, in combination with condition (i) for the fully

matching case, one only has to know which side (north, east, south or west) of the parameter domain defines the interface to identify the associated DOF. For example, in Fig. 1, using the double-index notation, we have

$$\mathcal{B}(i,j) \;=\; \{(k,\ell)\}_{\substack{k=1,\dots,5\\ \ell=4}} \;\subset\; \mathcal{R}^{(i)},$$
$$\mathcal{B}(j,i) \;=\; \{(k,\ell)\}_{\substack{k=1\\ \ell=1,\dots,5}} \;\subset\; \mathcal{R}^{(j)}.$$

Assume that the linear form $f(\cdot)$ can be assembled from contributions $f^{(i)}(\cdot)$ on $\Omega^{(i)}$, and let $a^{(i)}(\cdot,\cdot)$ denote the restriction of $a(\cdot,\cdot)$ to $\Omega^{(i)}$. Then, we can discretize the original minimization problem (2) as a sum of local contributions:

Find u_h such that

$$u_h = \operatorname*{arg\,min}_{v_h \in V_{gh}} \sum_{i=1}^{N} \Big(\tfrac{1}{2} a^{(i)}(v_h, v_h) - f^{(i)}(v_h) \Big). \qquad (13)$$

Condition (ii) for the fully matching setting implies that each DOF $u^{(i)}_{\boldsymbol{k}}$, $\boldsymbol{k} \in \mathcal{B}(i,j)$, can be identified with a DOF $u^{(j)}_{\boldsymbol{\ell}}$, $\boldsymbol{\ell} \in \mathcal{B}(j,i)$, such that the corresponding basis functions match as in (12). If we identify the DOF corresponding to these matching basis functions, then, together with the remaining DOF, we get a representation of the space V_h in (8). Employing a suitable numbering for the identified DOF, we can rewrite (13) in the form

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \qquad (14)$$

where $\mathbf{u}$ is the coefficient vector corresponding to the DOF which are not on the Dirichlet boundary, $\mathbf{K}$ is the stiffness matrix and $\mathbf{f}$ is the load vector. This system can be solved by standard sparse LU-factorization [11]. However, it is well known that for large problem size, the memory requirement and the runtime complexity of direct solvers are inefficient. Alternatively, one can use efficient iterative solvers such as conjugate gradient methods with appropriate preconditioners [43]. In the case of standard FEM discretizations, such preconditioners have been well studied in the literature, see e.g. [50] for geometric and algebraic multigrid methods, and see [49] for domain decomposition methods.

It is important to note that by assembling the system matrix $\mathbf{K}$ from the subdomain contributions, the structural (subdomain-wise) properties of the problem are lost, which are hard to regain from $\mathbf{K}$ alone. To alleviate this difficulty, in the following section we present a solver (inspired by the FETI methods [14, 16, 17, 18, 40, 49]) which inherently uses the local structure of (13). This approach is very suitable for parallelization, and moreover, since it mainly uses solvers on the local subdomains, their tensor product structure can be exploited (e.g. using wavelets or FFT).

4. Solver design

In the IETI method, we work in the space

$$\Pi V_h = \prod_{i=1}^{N} V_h^{(i)}$$

as defined in (7). Since these functions are, in general, discontinuous across subdomain interfaces, we need to impose the continuity conditions separately. In the following, let $v^{(i)}$ denote the i-th component of a function $v \in \Pi V_h$.

4.1. Continuity constraints

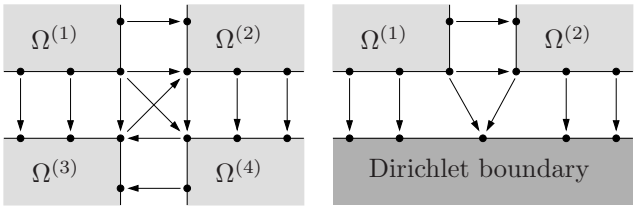
Note that in $\mathcal{C}_\Gamma$, as defined in (10), each edge $\Gamma^{(i,j)}$ is multiply represented, because $(i,j) \in \mathcal{C}_\Gamma$ and $(j,i) \in \mathcal{C}_\Gamma$. For each pair $(i,j) \in \mathcal{C}_\Gamma$, we choose a *master subdomain* and a *slave subdomain*. We express this by putting the master subdomain index first. Let $\mathcal{C} \subset \mathcal{C}_\Gamma$ be such that for each $(i,j) \in \mathcal{C}_\Gamma$ either $(i,j) \in \mathcal{C}$ or $(j,i) \in \mathcal{C}$, but not both. For all $(i,j) \in \mathcal{C}$ and a fixed index $\boldsymbol{k} \in \mathcal{B}(i,j)$, we can rewrite (12) in the following general form:

$$\widehat{R}^{(i)}_{\boldsymbol{k}}\big|_{\Gamma^{(i,j)}} = \sum_{\boldsymbol{\ell} \in \mathcal{B}(j,i)} a^{(i,j)}_{\boldsymbol{k},\boldsymbol{\ell}}\, \widehat{R}^{(j)}_{\boldsymbol{\ell}}\big|_{\Gamma^{(i,j)}}.$$

For a fixed $\boldsymbol{k} \in \mathcal{B}(i,j)$, all coefficients $a^{(i,j)}_{\boldsymbol{k},\boldsymbol{\ell}}$ are zero, except for one coefficient which is 1. We can collect the coefficients $a^{(i,j)}_{\boldsymbol{k},\boldsymbol{\ell}}$ in the permutation matrix

$$\mathbf{A}^{(i,j)} = (a^{(i,j)}_{\boldsymbol{k},\boldsymbol{\ell}})_{\substack{\boldsymbol{k} \in \mathcal{B}(i,j)\\ \boldsymbol{\ell} \in \mathcal{B}(j,i)}}. \qquad (15)$$

In the case of a subdomain vertex, where $\#\mathcal{B}(i,j) = 1$, the matrix $\mathbf{A}^{(i,j)}$ has only one entry which is 1.



(a) Fully redundant coupling at a subdomain vertex.

(b) Incorporation of Dirichlet boundary conditions by coupling to virtual neighbour subdomains.

Figure 2: Illustration of fully redundant coupling and all floating setting. Arrows indicate coupling conditions and point from master subdomain to slave subdomain. Note that the choice of master and slave is arbitrary in the fully matching case.

Note that, in the fully matching case, we can choose either $(i,j) \in \mathcal{C}$ or $(j,i) \in \mathcal{C}$. However, as we shall discuss in Section 5, the choice of the master subdomain for cases which are not fully matching is not arbitrary. Fig. 2(a) illustrates the coupling conditions between subdomains at a subdomain vertex.

To guarantee the continuity of u_h across all interfaces, we impose the following condition on the DOF associated with interfaces, i.e. for all $(i,j) \in \mathcal{C}$:

$$u_{\boldsymbol{k}}^{(i)} - \sum_{\boldsymbol{\ell} \in \mathcal{B}(j,i)} a_{\boldsymbol{k},\boldsymbol{\ell}}^{(i,j)} u_{\boldsymbol{\ell}}^{(j)} = 0 \quad \forall \boldsymbol{k} \in \mathcal{B}(i,j). \tag{16}$$

The incorporation of Dirichlet bondary conditions is done in a similar way, see the all-floating BETI method [40] and the total FETI method [14]. We denote the interface between $\partial\Omega^{(i)}$ and the Dirichlet boundary by

$$\Gamma^{(i,D)} = \partial\Omega^{(i)} \cap \Gamma_D.$$

Similar to (11), we collect the DOF of $u^{(i)}$ that are associated with the Dirichlet boundary, i.e. with the interface $\Gamma^{(i,D)}$, in the following index set:

$$\mathcal{D}(i) \;=\; \{\boldsymbol{k} \in \mathcal{R}^{(i)} :\; \operatorname{supp} \widehat{R}_{\boldsymbol{k}}^{(i)} \cap \Gamma^{(i,D)} \neq \emptyset\}.$$

Recall from Section 3.1 that we assume that there exists a function $\widehat{g} \in V_h$ with $\widehat{g}|_{\Gamma_D} = g_D|_{\Gamma_D}$. Hence, we can write

$$g_D\big|_{\Gamma^{(i,D)}} = \sum_{\boldsymbol{k} \in \mathcal{D}(i)} g_{\boldsymbol{k}}^{(i)} \widehat{R}_{\boldsymbol{k}}^{(i)}\big|_{\Gamma^{(i,D)}}$$

with real-valued coefficients $g_{\boldsymbol{k}}^{(i)}$. We incorporate the Dirichlet boundary conditions by imposing the following constraints on the DOF that are associated to the Dirichlet boundary, i.e. for all i such that $\Gamma^{(i,D)} \neq \emptyset$:

$$u_{\boldsymbol{k}}^{(i)} = g_{\boldsymbol{k}}^{(i)} \quad \forall \boldsymbol{k} \in \mathcal{D}(i). \tag{17}$$

The Dirichlet conditions can be thought of as continuity between the physical subdomains and a virtual neighbour subdomain (see Fig. 2(b) for an illustration).

Let J denote the total number of constraints of the form (16) and (17):

$$J = \sum_{(i,j)\in\mathcal{C}} \#\mathcal{B}(i,j) + \sum_{i=1}^{N} \#\mathcal{D}(i).$$

We assume a fixed numbering of these constraints and introduce the following notation. For a vector $y \in \mathbb{R}^J$, $y_{\boldsymbol{k}}^{(i,j)}$ denotes the component corresponding to the constraint (16), and $y_{\boldsymbol{k}}^{(i,D)}$ denotes the component corresponding to the constraint (17). We define the *jump operator* B as follows:

$$\begin{aligned} B:\; & \Pi V_h \to \mathbb{R}^J \\ (Bu)_{\boldsymbol{k}}^{(i,j)} =\; & u_{\boldsymbol{k}}^{(i)} - \sum_{\boldsymbol{\ell} \in \mathcal{B}(j,i)} a_{\boldsymbol{k},\boldsymbol{\ell}}^{(i,j)} u_{\boldsymbol{\ell}}^{(j)} \qquad (18) \\ (Bu)_{\boldsymbol{k}}^{(i,D)} =\; & u_{\boldsymbol{k}}^{(i)}. \qquad (19) \end{aligned}$$

Hence, the conditions for C^0-continuity (16) and the incorporation of Dirichlet boundary conditions (17) read

$$Bu = \mathbf{b}, \tag{20}$$

where the entry of the vector $\mathbf{b} \in \mathbb{R}^J$ is zero when corresponding to an interface condition (18), and it equals $g_{\boldsymbol{k}}^{(i)}$ when corresponding to a Dirichlet boundary condition (19). For $\widehat{g} \in V_h$ with $\widehat{g}|_{\Gamma_D} = g_D|_{\Gamma_D}$, we have $\mathbf{b} = B\widehat{g}$. Note that the linear operator B can be represented by a signed Boolean matrix. With (20), we obtain the following restricted minimization problem which is equivalent to (13):

Find u such that

$$u = \operatorname*{arg\,min}_{\substack{v \in \Pi V_h \\ Bv = \mathbf{b}}} \sum_{i=1}^{N} \Big(\tfrac{1}{2} a^{(i)}(v^{(i)}, v^{(i)}) - f^{(i)}(v^{(i)}) \Big). \tag{21}$$

4.2. Saddle point formulation

Using the local basis of each subdomain space $V_h^{(i)}$, each function $v^{(i)} \in V_h^{(i)}$ is uniquely represented by a vector $\mathbf{v}^{(i)}$. Correspondingly, each function $v \in \Pi V_h$ has a representation as a vector $\mathbf{v}$ of the form

$$\mathbf{v} = (\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(N)}) \tag{22}$$

whose blocks are the local vectors $\mathbf{v}^{(i)}$. Let $\mathbf{K}^{(i)}$ denote the stiffness matrix corresponding to the local bilinear form $a^{(i)}(\cdot,\cdot)$ and define

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}^{(1)} & & 0 \\ & \ddots & \\ 0 & & \mathbf{K}^{(N)} \end{pmatrix}.$$

Analogously, let $\mathbf{f}$ denote the load vector whose blocks $\mathbf{f}^{(i)}$ correspond to the local subdomain load vectors, and let $\mathbf{B}$ be the matrix representation of the jump operator B.

The minimization problem (21) is then equivalent to the following saddle point problem:

Find $u \in \Pi V_h$ with the vector representation $\mathbf{u}$ as in (22) and Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^J$, such that

$$\begin{pmatrix} \mathbf{K} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{b} \end{pmatrix}. \tag{23}$$

We note that even though $\boldsymbol{\lambda}$ is only unique up to an element from $\ker \mathbf{B}^T$, the solution $\mathbf{u}$ is unique.

The common strategy of FETI-type methods is to reduce (23) to an equation that involves only $\boldsymbol{\lambda}$. This is not straightforward, since in the case of our model problems (see Section 2.1) the local matrices $\mathbf{K}^{(i)}$ are not invertible. More precisely, in the scalar elliptic case (I) the kernel is spanned by the constant functions, and for the two-dimensional linearized elasticity problem (II), the kernel is spanned by three rigid body modes.

In the classical FETI methods [16] and the total FETI method [14], additional unknowns are introduced that span the kernel. In this paper, however, we will follow the dual-primal FETI (FETI-DP) method, see [17] and [49, Sect. 6.4], thus obtaining IETI-DP.

4.3. Dual-primal formulation

Recall that we only consider NURBS geometry mappings with open knot vectors. Therefore, at every vertex of the parameter domain Q, there is exactly one index $\boldsymbol{k}_0 \in \mathcal{R}^{(i)}$, such that $R^{(i)}_{\boldsymbol{k}_0}$ at this vertex is 1, while all other basis functions are zero. Hence, we can distinguish DOF that are associated with the vertices of the parameter domain. Such DOF that are associated with the vertices of a parameter domain are called *primal* DOF. All other DOF are referred to as *non-primal* or *remaining* DOF.

We define the following subset of ΠV_h

$$\widetilde{W}_h = \{v \in \Pi V_h : \ v \text{ is continuous at all vertices}\}.$$

To achieve the continuity above, we identify all the primal DOF that are associated with a common point in the physical domain, and we fix a *global numbering* of these primal DOF. Then, a function $v \in \widetilde{W}_h$ can be uniquely represented by a vector $\widetilde{\mathbf{v}}$ of the form

$$\widetilde{\mathbf{v}} = (\widetilde{\mathbf{v}}_P, \widetilde{\mathbf{v}}_R)^T = (\widetilde{\mathbf{v}}_P, \widetilde{\mathbf{v}}_R^{(1)}, \ldots, \widetilde{\mathbf{v}}_R^{(N)})^T, \tag{24}$$

where the subscripts P and R refer to *primal* and *remaining* DOF, respectively. Note that $\mathbf{v}$ in (22) and $\widetilde{\mathbf{v}}$ in (24) are different vector representations of the same function $v \in \widetilde{W}_h \subset \Pi V_h$.

Let $\widetilde{\mathbf{B}}$ denote the jump operator on $\widetilde{W}_h$ defined by

$$\widetilde{\mathbf{B}}\widetilde{\mathbf{u}} = Bu,$$

where $\widetilde{\mathbf{u}}$ is the representation of u in the form of (24). Analogously to (24), we can write

$$\widetilde{\mathbf{B}} = (\widetilde{\mathbf{B}}_P, \widetilde{\mathbf{B}}_R) = (\widetilde{\mathbf{B}}_P, \widetilde{\mathbf{B}}_R^{(1)}, \ldots, \widetilde{\mathbf{B}}_R^{(N)}).$$

4.3.1. Setting up the global system

Since the solution $u \in \widetilde{W}_h$, we can replace the space ΠV_h in (23) by $\widetilde{W}_h$. In the following, we will derive an equivalent saddle point formulation.

By rearranging the local DOF $\mathbf{u}^{(i)}$ in such a way that the primal DOF come first, the subdomain stiffness matrix $\mathbf{K}^{(i)}$ and the load vector $\mathbf{f}^{(i)}$ take the form

$$\widetilde{\mathbf{K}}^{(i)} = \begin{pmatrix} \widetilde{\mathbf{K}}^{(i)}_{PP} & \widetilde{\mathbf{K}}^{(i)}_{PR} \\ \widetilde{\mathbf{K}}^{(i)}_{RP} & \widetilde{\mathbf{K}}^{(i)}_{RR} \end{pmatrix}, \quad \widetilde{\mathbf{f}}^{(i)} = \begin{pmatrix} \widetilde{\mathbf{f}}^{(i)}_{P} \\ \widetilde{\mathbf{f}}^{(i)}_{R} \end{pmatrix}. \tag{25}$$

From these local contributions, we obtain the global stiffness matrix $\widetilde{\mathbf{K}}$ and the global load vector $\widetilde{\mathbf{f}}$

$$\widetilde{\mathbf{K}} = \begin{pmatrix} \widetilde{\mathbf{K}}_{PP} & \widetilde{\mathbf{K}}_{PR} \\ \widetilde{\mathbf{K}}_{RP} & \widetilde{\mathbf{K}}_{RR} \end{pmatrix}, \quad \widetilde{\mathbf{f}} = \begin{pmatrix} \widetilde{\mathbf{f}}_{P} \\ \widetilde{\mathbf{f}}_{R} \end{pmatrix}. \tag{26}$$

Note that, due to the identification of the primal DOF, the components carrying a subscript P in (26) are *assembled* from local contributions. $\widetilde{\mathbf{K}}_{RR}$ and $\widetilde{\mathbf{f}}_R$ have the form

$$\widetilde{\mathbf{K}}_{RR} = \begin{pmatrix} \widetilde{\mathbf{K}}^{(1)}_{RR} & & 0 \\ & \ddots & \\ 0 & & \widetilde{\mathbf{K}}^{(N)}_{RR} \end{pmatrix}, \ \widetilde{\mathbf{f}}_{RR} = \begin{pmatrix} \widetilde{\mathbf{f}}^{(1)}_{R} \\ \vdots \\ \widetilde{\mathbf{f}}^{(N)}_{R} \end{pmatrix}.$$

Note that $\widetilde{\mathbf{K}}_{PP}$ is not block-diagonal.

The coupling conditions of the form (18) can be neglected at the primal DOF, but it has no effect on the algorithm if they are included. Depending on the implementation of the jump operator, one might decide to keep them or not.

With the same steps as before, we arrive at the following saddle point problem which is equivalent to (23):

Find $u \in \widetilde{W}_h$, represented by $\widetilde{\mathbf{u}}$ as in (24), and Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^J$, such that

$$\begin{pmatrix} \widetilde{\mathbf{K}} & \widetilde{\mathbf{B}}^T \\ \widetilde{\mathbf{B}} & 0 \end{pmatrix} \begin{pmatrix} \widetilde{\mathbf{u}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \widetilde{\mathbf{f}} \\ \mathbf{b} \end{pmatrix}. \tag{27}$$

Note that the matrix $\widetilde{\mathbf{K}}$ in (27) is singular, because the space $\widetilde{W}_h$ has no restrictions on the Dirichlet boundary for our model problems. In the next section, we will incorporate Dirichlet boundary conditions at those primal DOF that are associated with the Dirichlet boundary. This incorporation will lead to a regular matrix.

4.3.2. Dirichlet boundary conditions

We distinguish between *Dirichlet* primal DOF associated with the Dirichlet boundary, and *floating* primal DOF that are in the interior of Ω or associated with the Neumann boundary. We indicate this with the subscripts d and f, respectively. We assume for simplicity that in the vector $\widetilde{\mathbf{u}}$ the Dirichlet primal DOF are listed first, i.e.

$$\widetilde{\mathbf{u}} = (\widetilde{\mathbf{u}}_d, \widetilde{\mathbf{u}}_f, \widetilde{\mathbf{u}}_R)^T = (\widetilde{\mathbf{u}}_d, \widetilde{\mathbf{u}}_f, \widetilde{\mathbf{u}}_R^{(1)}, \ldots, \widetilde{\mathbf{u}}_R^{(N)})^T \tag{28}$$

and

$$\widetilde{\mathbf{K}} = \begin{pmatrix} \widetilde{\mathbf{K}}_{dd} & \widetilde{\mathbf{K}}_{df} & \widetilde{\mathbf{K}}_{dR} \\ \widetilde{\mathbf{K}}_{fd} & \widetilde{\mathbf{K}}_{ff} & \widetilde{\mathbf{K}}_{fR} \\ \widetilde{\mathbf{K}}_{Rd} & \widetilde{\mathbf{K}}_{Rf} & \widetilde{\mathbf{K}}_{RR} \end{pmatrix}, \quad \widetilde{\mathbf{f}} = \begin{pmatrix} \widetilde{\mathbf{f}}_{d} \\ \widetilde{\mathbf{f}}_{f} \\ \widetilde{\mathbf{f}}_{R} \end{pmatrix},$$
$$\widetilde{\mathbf{B}} = \begin{pmatrix} \widetilde{\mathbf{B}}_d & \widetilde{\mathbf{B}}_f & \widetilde{\mathbf{B}}_R \end{pmatrix}.$$

Let $\widetilde{\mathbf{g}}_d$ be the vector whose entries are the values of g_D at the Dirichlet primal DOF. Since $\widetilde{\mathbf{u}}_d = \widetilde{\mathbf{g}}_d$, the saddle point problem (27) is equivalent to the following problem:

Find $u \in \widetilde{W}_h$, represented by $\widetilde{\mathbf{u}}$ as in (28), and Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^J$, such that

$$\begin{pmatrix} \overline{\mathbf{K}} & \overline{\mathbf{B}}^T \\ \overline{\mathbf{B}} & 0 \end{pmatrix} \begin{pmatrix} \widetilde{\mathbf{u}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \overline{\mathbf{f}} \\ \overline{\mathbf{b}} \end{pmatrix}, \tag{29}$$

where

$$\overline{\mathbf{K}} = \begin{pmatrix} \mathbf{I} & 0 & 0 \\ 0 & \widetilde{\mathbf{K}}_{ff} & \widetilde{\mathbf{K}}_{fR} \\ 0 & \widetilde{\mathbf{K}}_{Rf} & \widetilde{\mathbf{K}}_{RR} \end{pmatrix}, \tag{30}$$

$$\overline{\mathbf{f}} = \begin{pmatrix} \widetilde{\mathbf{g}}_d \\ \widetilde{\mathbf{f}}_f - \widetilde{\mathbf{K}}_{fd}\widetilde{\mathbf{g}}_d \\ \widetilde{\mathbf{f}}_R - \widetilde{\mathbf{K}}_{Rd}\widetilde{\mathbf{g}}_d \end{pmatrix}, \tag{31}$$

$$\overline{\mathbf{B}} = \begin{pmatrix} 0 & \widetilde{\mathbf{B}}_f & \widetilde{\mathbf{B}}_R \end{pmatrix}, \tag{32}$$

$$\overline{\mathbf{b}} = \mathbf{b} - \widetilde{\mathbf{B}}_d\widetilde{\mathbf{g}}_d \tag{33}$$

and $\mathbf{I}$ is the identity matrix. We see that each entry of $\overline{\mathbf{b}}$ corresponding to a multiplier acting on a Dirichlet primal DOF vanishes. Also, these multipliers are superfluous as they do not influence the solution $\widetilde{\mathbf{u}}$ and can be left out completely.

As before, we denote the components of $\overline{\mathbf{K}}$, $\overline{\mathbf{f}}$, and $\overline{\mathbf{B}}$ in (30)–(32) that correspond to primal and remaining DOF with the subscripts P and R, respectively. Hence,

$$\begin{aligned}
\overline{\mathbf{K}}_{PP} &= \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \widetilde{\mathbf{K}}_{ff} \end{pmatrix}, & \overline{\mathbf{K}}_{PR} &= \begin{pmatrix} 0 \\ \widetilde{\mathbf{K}}_{fR} \end{pmatrix}, \\
\overline{\mathbf{K}}_{RP} &= \begin{pmatrix} 0 & \widetilde{\mathbf{K}}_{Rf} \end{pmatrix}, & \overline{\mathbf{K}}_{RR} &= \widetilde{\mathbf{K}}_{RR}, \\
\overline{\mathbf{f}}_P &= \begin{pmatrix} \widetilde{\mathbf{g}}_d \\ \widetilde{\mathbf{f}}_f - \widetilde{\mathbf{K}}_{fd}\widetilde{\mathbf{g}}_d \end{pmatrix}, & \overline{\mathbf{f}}_R &= \widetilde{\mathbf{f}}_R - \widetilde{\mathbf{K}}_{Rd}\widetilde{\mathbf{g}}_d, \\
\overline{\mathbf{B}}_P &= \begin{pmatrix} 0 & \widetilde{\mathbf{B}}_f \end{pmatrix}, & \overline{\mathbf{B}}_R &= \widetilde{\mathbf{B}}_R.
\end{aligned}$$

Again, the coupling conditions of the form (18) and (19) can be neglected at all the Dirichlet primal DOF, but it has no effect on the algorithm if they remain.

Remark 4.1. Similar to the construction (30)–(33), we can also directly incorporate the Dirichlet boundary conditions at the remaining non-primal Dirichlet DOF. If this is done, the corresponding multipliers can again be left out. This approach would be closer to the original FETI-DP method as proposed in [17].

4.3.3. Dual problem

In our model problems the matrix $\overline{\mathbf{K}}$ in (30) is invertible, and the first line of (29) yields

$$\widetilde{\mathbf{u}} = \overline{\mathbf{K}}^{-1}(\overline{\mathbf{f}} - \overline{\mathbf{B}}^T\boldsymbol{\lambda}).$$

Inserting this identitiy into the second line of (29), we obtain the dual problem

$$\mathbf{F}\boldsymbol{\lambda} \;=\; \mathbf{d} \tag{34}$$

with $\mathbf{F} = \overline{\mathbf{B}}\,\overline{\mathbf{K}}^{-1}\overline{\mathbf{B}}^T$ and $\mathbf{d} = \overline{\mathbf{B}}\,\overline{\mathbf{K}}^{-1}\overline{\mathbf{f}} - \overline{\mathbf{b}}$. To realize the application of $\overline{\mathbf{K}}^{-1}$, we use the block factorization

$$\begin{aligned}
\overline{\mathbf{K}}^{-1} = &\begin{pmatrix} \mathbf{I} & 0 \\ -\overline{\mathbf{K}}_{RR}^{-1}\overline{\mathbf{K}}_{RP} & \mathbf{I} \end{pmatrix} \\
&\begin{pmatrix} \overline{\mathbf{S}}_{PP}^{-1} & 0 \\ 0 & \overline{\mathbf{K}}_{RR}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -\overline{\mathbf{K}}_{PR}\overline{\mathbf{K}}_{RR}^{-1} \\ 0 & \mathbf{I} \end{pmatrix}
\end{aligned} \tag{35}$$

where

$$\overline{\mathbf{S}}_{PP} = \overline{\mathbf{K}}_{PP} - \overline{\mathbf{K}}_{PR}\overline{\mathbf{K}}_{RR}^{-1}\overline{\mathbf{K}}_{RP}.$$

Recall that $\mathbf{K}_{RR}$ is block diagonal. Hence, applying $\mathbf{K}_{RR}^{-1}$ corresponds to solving local problems independently on each subdomain, e.g. by sparse LU factorization [11]. Note that $\mathbf{K}_{RR}^{-1}$ appears three times in (35), but it has to be applied only twice, because two of the applications of $\mathbf{K}_{RR}^{-1}$ are on the same vector.

The matrix $\overline{\mathbf{S}}_{PP}$ can be assembled from local contributions

$$\overline{\mathbf{S}}_{PP}^{(i)} = \overline{\mathbf{K}}_{PP}^{(i)} - \overline{\mathbf{K}}_{PR}^{(i)}(\overline{\mathbf{K}}_{RR}^{(i)})^{-1}\overline{\mathbf{K}}_{RP}^{(i)}.$$

One can show that $\overline{\mathbf{S}}_{PP}$ is sparse and that it can be factorized using standard sparse LU factorization [11]. The size of $\overline{\mathbf{S}}_{PP}$ is determined by the number of primal DOF. Since we only have primal DOF at the subdomain vertices, their number is bounded by $4N$. Typically, the number of subdomains, and therefore the size of $\overline{\mathbf{S}}_{PP}$, is much smaller than the size of $\mathbf{K}_{RR}^{(i)}$, but even in the case of many subdomains $\overline{\mathbf{S}}_{PP}$ is sparse.

We can solve the symmetric and positive definite system (34) for $\boldsymbol{\lambda}$ by a CG algorithm. Once we have obtained $\boldsymbol{\lambda}$, we can calculate

$$\widetilde{\mathbf{u}}_P = \overline{\mathbf{S}}_{PP}^{-1}\Big(\overline{\mathbf{f}}_P - \overline{\mathbf{B}}_P^T\boldsymbol{\lambda} - \overline{\mathbf{K}}_{PR}\overline{\mathbf{K}}_{RR}^{-1}(\overline{\mathbf{f}}_R - \overline{\mathbf{B}}_R^T\boldsymbol{\lambda})\Big). \tag{36}$$

The remaining local solutions are then given by

$$\widetilde{\mathbf{u}}_R = \overline{\mathbf{K}}_{RR}^{-1}\Big(\overline{\mathbf{f}}_R - \overline{\mathbf{B}}_R^T\boldsymbol{\lambda} - \overline{\mathbf{K}}_{RP}\widetilde{\mathbf{u}}_P\Big). \tag{37}$$

In [18], the unpreconditioned interface problem (34) is discussed for the classical FETI method, and it is shown that the condition number is of order

$$\kappa(\mathbf{F}) = \mathcal{O}(H/h), \tag{38}$$

where H and h denote the characteristic subdomain size and the finite element mesh size, respectively. The numerical tests presented in Section 6 indicate that the IETI method behaves similarly. In the next section, following [17], we define a preconditioner for the interface problem which will be used for the numerical examples in Section 6.

4.4. Preconditioner

Our construction follows the scaled Dirichlet preconditioner that was introduced in [18, 26, 42] and extended to the dual-primal formulation in [17, 24]. We indicate *interior* DOF with the subscript I, and the DOF associated with the *boundary* $\partial\Omega^{(i)}$ of a subdomain with the subscript B. Assume that the DOF are now numbered such that the interior DOF are listed first, then the local stiffness matrix $\mathbf{K}^{(i)}$ takes the form

$$\mathbf{K}^{(i)} = \begin{pmatrix} \mathbf{K}_{II}^{(i)} & \mathbf{K}_{IB}^{(i)} \\ \mathbf{K}_{BI}^{(i)} & \mathbf{K}_{BB}^{(i)} \end{pmatrix}. \tag{39}$$

The dual-primal Dirichlet preconditioner is defined by

$$\mathbf{M}^{-1} = \sum_{i=1}^{N} \mathbf{D}^{(i)}\,\mathbf{B}^{(i)} \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{S}_{BB}^{(i)} \end{pmatrix} \mathbf{B}^{(i)\,T}\,\mathbf{D}^{(i)}, \tag{40}$$

where

$$\mathbf{S}_{BB}^{(i)} = \mathbf{K}_{BB}^{(i)} - \mathbf{K}_{BI}^{(i)}(\mathbf{K}_{II}^{(i)})^{-1}\mathbf{K}_{IB}^{(i)}.$$

Since $\mathbf{K}_{II}^{(i)}$ is the local stiffness matrix of $\Omega^{(i)}$ with all boundary DOF fixed, it can be factorized as easily and cheaply as $\mathbf{K}_{RR}^{(i)}$. The matrix $\mathbf{B}^{(i)}$ in (40) is the restriction of $\mathbf{B}$ to the interface conditions associated with $\Omega^{(i)}$. The matrix $\mathbf{D}^{(i)}$ is a scaled diagonal matrix of size $J \times J$, where J is the number of Lagrange multipliers. Its entries are

$$(\mathbf{D}^{(i)})_{kk} = 1/\operatorname{mult}(k),$$

where $\operatorname{mult}(k)$ is the number of subdomains which have interfaces associated with the Lagrange multiplier λ_k. In particular, $\operatorname{mult}(k)$ takes the following values:

- $\operatorname{mult}(k) = 1$, if λ_k corresponds to a Dirichlet boundary condition.
- $\operatorname{mult}(k) = 2$, if λ_k corresponds to a coupling condition that does not involve a subdomain vertex.
- $\operatorname{mult}(k) \geq 2$, if λ_k corresponds to a coupling condition that involves a subdomain vertex.

These scalings can, e.g., be found in [24, 26], where the authors show that certain jumps in the diffusion coefficient (problem (I)) or the Lamé parameters (problem (II)) can be treated robustly.

In [24], it was shown that the condition number of the preconditioned FETI-DP interface problem behaves like

$$\kappa(\mathbf{M}^{-1}\mathbf{F}) = \mathcal{O}((1 + \log(H/h)^2),$$

where H and h are as defined at the end of Section 4.3. The numerical results presented in Section 6 show that a similar behaviour can be observed in the IETI method.

4.5. Isogeometric tearing and interconnecting algorithm

To summarize, the overall IETI-DP algorithm is as follows.

1. For each $i = 1, \ldots, N$, locally on each subdomain $\Omega^{(i)}$ (in parallel)
   a) Assemble the local stiffness matrix $\mathbf{K}^{(i)}$ and load vector $\mathbf{f}^{(i)}$ using a fully local numbering of the DOF.
   b) Partition $\mathbf{K}^{(i)}$ and $\mathbf{f}^{(i)}$ as in (26), factorize $\mathbf{K}_{RR}^{(i)}$, calculate $\overline{\mathbf{S}}_{PP}^{(i)}$.
   c) Partition $\mathbf{K}^{(i)}$ as in (39), factorize $\mathbf{K}_{II}^{(i)}$.
2. Assemble and factorize $\overline{\mathbf{S}}_{PP}$, calculate $\mathbf{d}$.
3. Solve $\mathbf{F}\boldsymbol{\lambda} = \mathbf{d}$ by PCG with preconditioner $\mathbf{M}^{-1}$ as in (40).
4. Calculate $\widetilde{\mathbf{u}}_P$ as in (36),

$$\widetilde{\mathbf{u}}_P = \overline{\mathbf{S}}_{PP}^{-1}\Big(\overline{\mathbf{f}}_P - \overline{\mathbf{B}}_P^T\boldsymbol{\lambda} - \overline{\mathbf{K}}_{PR}\overline{\mathbf{K}}_{RR}^{-1}(\overline{\mathbf{f}}_R - \overline{\mathbf{B}}_R^T\boldsymbol{\lambda})\Big).$$

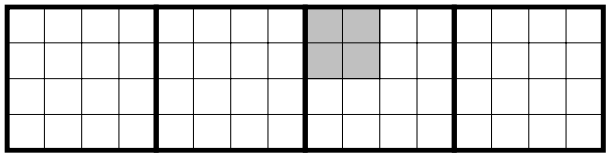
5. For each $i = 1, \ldots, N$, obtain $\widetilde{\mathbf{u}}_R^{(i)}$ as in (37) (in parallel),

$$\widetilde{\mathbf{u}}_R^{(i)} = {\mathbf{K}_{RR}^{(i)}}^{-1}\Big(\mathbf{f}_R^{(i)} - (\overline{\mathbf{B}}_R^{(i)})^T\boldsymbol{\lambda} - \overline{\mathbf{K}}_{RP}^{(i)}\widetilde{\mathbf{u}}_P\Big).$$

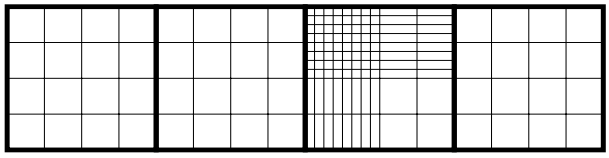
5. Refinement options

The tensor-product structure of NURBS basis functions is inconvenient for local refinement. The insertion of a knot affects the whole domain and may introduce superfluous DOF. Recent work on methods for local refinement in IGA includes analysis suitable T-splines (see, e.g. [4, 32, 33, 44, 45, 46]) and PHT-splines (see [12, 38]), as well as the use of finite element-based strategies in the IGA framework (see [28]).

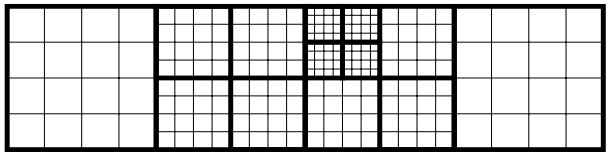
However, the IETI-approach introduces some possibilites for restricting the refinement to one or a few subdomains (of many), even when working with tensor-product NURBS basis functions and straight-forward knot insertion. We will sketch two such methods: h-refinement on one subdomain, and refinement by substructuring (see Fig. 3). Note that, in both cases, we assume that the initial setting is fully matching.



(a) Shaded area marked for refinement.



(b) h-refinement of (a) on one subdomain (see Section 5.1).



(c) Refinement of (a) by two steps of 1-level substructuring (see Sections 5.2 and 5.3).

Figure 3: Two options for refining the shaded area in (a).

5.1. h-Refinement on one subdomain

Although a knot insertion affects the whole *parameter* domain due to the tensor-product structure of the NURBS basis functions, we can limit the refinement to a single subdomain, as depicted in Fig. 3(b).

We assume that on an interface, which is not fully matching, the knot vector on one subdomain (the *fine side*) is a refinement of the knot vector on the other subdomain (the *coarse side*), as in the example in Fig. 3(b). In Fig. 4, such a case is illustrated schematically: The knot vector $s^{(i)}$ is obtained from $s^{(j)}$ by one step of uniform h-refinement. In constrast to the fully matching case, the numbers of DOF of $V_h^{(i)}$ and $V_h^{(j)}$ on the interface $\Gamma^{(i,j)}$ are

not equal, and condition (ii) for the fully matching case is not fulfilled. In reference to hanging nodes in finite element methods, we call this a *setting with hanging knots* (note that we still assume that the geometry is conforming).

As a consequence, we cannot couple the DOF as simply as in (16). In particular, the matrix $\mathbf{A}^{(i,j)}$ in (15) has to be modified accordingly.

The number of interface conditions on such an interface is determined by the fine side, which is chosen as the master subdomain. Hence, for $(i,j) \in \mathcal{C}_\Gamma$, we set

$$\begin{aligned} &(i,j) \in \mathcal{C}, \quad \text{if } \#\mathcal{B}(i,j) \geq \#\mathcal{B}(j,i), \\ &(j,i) \in \mathcal{C}, \quad \text{otherwise.} \end{aligned}$$

For example, in Fig. 4, the master subdomain is $\Omega^{(i)}$, i.e. $(i,j) \in \mathcal{C}$.

Without loss of generality, we assume that $s^{(i)}$ is a refinement of $s^{(j)}$ and that the weights on the finer side are obtained by the knot insertion algorithm [41]. Hence, on the interface $\Gamma^{(i,j)}$ the coarse basis function $\widehat{R}^{(j)}_{\boldsymbol{\ell}}|_{\Gamma^{(i,j)}}$, $\boldsymbol{\ell} \in \mathcal{B}(j,i)$ can be represented *exactly* as a linear combination of fine basis functions $\widehat{R}^{(i)}_{\boldsymbol{k}}|_{\Gamma^{(i,j)}}$, $\boldsymbol{k} \in \mathcal{B}(i,j)$. Therefore, for each $\boldsymbol{\ell} \in \mathcal{B}(j,i)$, there exist coefficients $z_{\boldsymbol{\ell},\boldsymbol{k}}$, such that

$$\widehat{R}^{(j)}_{\boldsymbol{\ell}}|_{\Gamma^{(i,j)}} = \sum_{\boldsymbol{k} \in \mathcal{B}(i,j)} z_{\boldsymbol{\ell},\boldsymbol{k}}\ \widehat{R}^{(i)}_{\boldsymbol{k}}|_{\Gamma^{(i,j)}}. \tag{41}$$

The coefficients $z_{\boldsymbol{\ell},\boldsymbol{k}}$ can be obtained from well-known formulae for the refinement of B-Spline basis functions [41].

We require C^0-continuity of u_h across the interface $\Gamma^{(i,j)}$, i.e. we require

$$\begin{aligned} \sum_{\boldsymbol{k} \in \mathcal{B}(i,j)} u^{(i)}_{\boldsymbol{k}} \widehat{R}^{(i)}_{\boldsymbol{k}}|_{\Gamma^{(i,j)}} &= \sum_{\boldsymbol{\ell} \in \mathcal{B}(j,i)} u^{(j)}_{\boldsymbol{\ell}} \widehat{R}^{(j)}_{\boldsymbol{\ell}}|_{\Gamma^{(i,j)}} \\ &= \sum_{\substack{\boldsymbol{\ell} \in \mathcal{B}(j,i) \\ \boldsymbol{k} \in \mathcal{B}(i,j)}} u^{(j)}_{\boldsymbol{\ell}} z_{\boldsymbol{\ell},\boldsymbol{k}}\ \widehat{R}^{(i)}_{\boldsymbol{k}}|_{\Gamma^{(i,j)}}. \end{aligned}$$

By comparing the coefficients, we obtain

$$u^{(i)}_{\boldsymbol{k}} - \sum_{\boldsymbol{\ell} \in \mathcal{B}(j,i)} z_{\boldsymbol{\ell},\boldsymbol{k}}\ u^{(j)}_{\boldsymbol{\ell}} = 0,$$

i.e. we obtain a continuity constraint in the same form as in (16) and (18), where the coefficients of the coupling matrix $\mathbf{A}^{(i,j)}$, see (15), are given by

$$a^{(i,j)}_{\boldsymbol{k},\boldsymbol{\ell}} = z_{\boldsymbol{\ell},\boldsymbol{k}}.$$

Remark 5.1. In [22], the issue of coupling mortar discretizations in the FETI-DP context was addressed. The procedure for formulating the jump operator given therein would result in the same coefficients $a^{(i,j)}_{\boldsymbol{k},\boldsymbol{\ell}}$ if it is applied to the setting considered here.

With the modified coupling matrices, one can then perform the same steps as in section 4.3.2 and the sections thereafter. Here we would like to point out that there are Lagrange multipliers which connect primal Dirichlet DOF with a boundary condition as well as multipliers connecting one and the same primal DOF. Both types of multipliers are superfluous and can be left out. However, in the presence of an interface with hanging knots, there are also multipliers that connect primal and remaining DOF. These multipliers cannot be left out and the corresponding entries in the vector $\overline{\mathbf{b}}$ do not vanish in general.

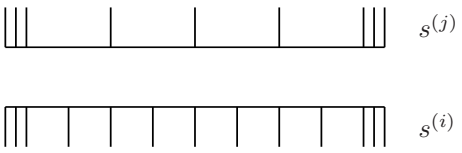


Figure 4: Interface with hanging knots: $p^{(i)} = p^{(j)} = 2$, $s^{(i)} = \{0,0,0,1/8,2/8,\ldots,7/8,1,1,1\}$ is a refinement of $s^{(j)} = \{0,0,0,1/4,1/2,3/4,1,1,1\}$.

5.2. Local refinement by substructuring

As an alternative, the number of DOF can be increased locally by subdividing one subdomain into smaller subdomains as illustrated in Fig. 3(c).

If one wanted to subdivide the NURBS geometry mapping, e.g. in order to be able to edit the geometry locally, one would split the mapping and construct new knot vectors, weights and control nets for the new geometry mappings. This is not necessary in the IETI context, since we are not interested in actually splitting the geometry representation. Instead, we split the parameter domain into smaller subdomains which are mapped to the physical domain by the original (unchanged) coarse geometry mapping.

A very simple method for substructuring the subdomain $\Omega^{(i)}$ is the following: We split the parameter domain $Q = [0,1]^2$ into four subdomains

$$\begin{aligned} Q^{(i,1)} &= (0,1/2) \times (0,1/2), \\ Q^{(i,2)} &= (0,1/2) \times (1/2,1), \\ Q^{(i,3)} &= (1/2,1) \times (0,1/2), \\ Q^{(i,4)} &= (1/2,1) \times (1/2,1). \end{aligned}$$

We refer to this substructuring method as *cross insertion*. The basis functions of the original parameter domain Q are pushed forward to the smaller subdomain $Q^{(i,k)}$ by a linear mapping $G^{(i,k)} : Q \to Q^{(i,k)}$ and then transformed to the physical domain by the original mapping $G^{(i)}$ (see Fig. 5 for an illustration). The basis functions on $\Omega^{(i,k)}$ have the form

$$R^{(i)}_{\boldsymbol{k}} \circ \left({G^{(i,k)}}^{-1} \circ {G^{(i)}}^{-1} \right).$$

The domain decompostition obtained by substructuring is again a setting with hanging knots. The matrix $\mathbf{A}^{(i,j)}$ in the interface condition (16) is adapted as described in Section 5.1.

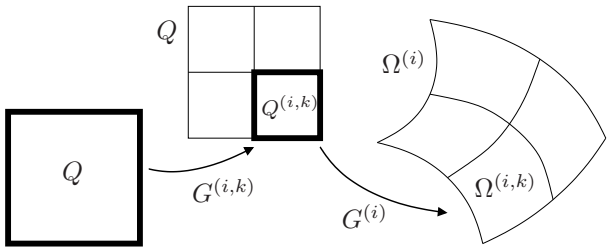
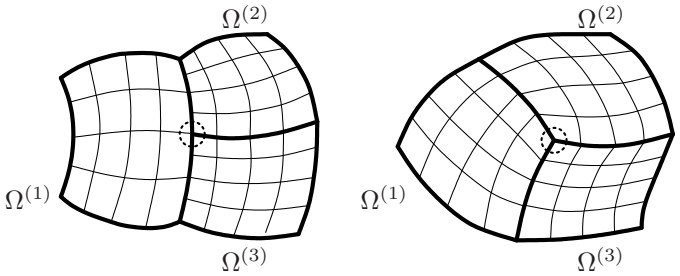


Figure 5: Embedding smaller subdomains $Q^{(i,k)}$ into the original parameter domain $Q^{(i)}$.

5.3. Substructuring and primal DOF

When we refine by substructuring, we introduce situations where the vertex of one subdomain coincides with the edge of another subdomain. Such cases are illustrated in Fig. 3(c) and Fig. 6(a). We call such a subdomain vertex a *hanging subdomain vertex* (or short *hanging vertex*). Note that not every T-shaped subdomain vertex is a hanging vertex, as illustrated in the example in Fig. 6(b).



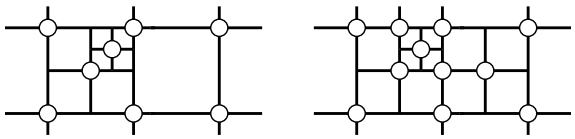
(a) Hanging subdomain vertex marked by dashed circle.

(b) T-shaped, but not hanging subdomain vertex.

Figure 6: Examples for hanging and not hanging subdomain vertices.

The choice of primal DOF in substructured subdomains, where we have hanging vertices, is not as straightforward as in the fully matching case. In the example of a hanging vertex in Fig. 6(a), there is exactly one DOF on $\Omega^{(2)}$ that is associated with the hanging vertex marked by the dashed circle (cf. the discussion at the beginning of Section 4.3). While the same applies to $\Omega^{(3)}$, this is not true on $\Omega^{(1)}$, where we have several NURBS basis functions which are nonzero at the marked hanging vertex. Instead of incorporating a special treatment of hanging vertices, we choose to omit primal DOF at hanging vertices and discuss under which conditions this is possible.

For the scalar elliptic problem (I), the kernel of the stiffness matrix of a floating subdomain is spanned by the constant function, i.e. the kernel has dimension one. In this case, it is sufficient to have at least one primal DOF on each subdomain. This is easily guaranteed, if we start from a fully matching setting, apply substructuring by cross-insertion as described in Section 5.2, and select primal DOF at all subdomain vertices which are not hanging. The example in Fig. 7(a) shows the positions of primal DOF after two cross insertions.



(a) Cross-insertion results in at least one primal DOF on each subdomain.

(b) 1-level refinement results in at least two primal DOF on each subdomain.

Figure 7: Subdomains refined by substructuring. Positions of primal DOF marked by ◯.

For the two-dimensional linearized elasticity problem (II), where the kernel is spanned by the three rigid body modes, we need at least two primal DOF per subdomain. As illustrated in Fig. 7(a), this is not guaranteed if we apply substructuring by cross-insertion without additional considerations.

For linearized elasticity problems, we introduce *refinement levels* and we assign refinement level 0 to every subdomain in the initial setting. When a subdomain is split into four smaller subdomains by cross insertion, the levels of the new, smaller subdomains are increased by 1 (see Fig. 8 for an illustration). We call the refinement a *1-level substructuring*, if the refinement levels of any two subdomains with an edge as their interface differ by at most 1. If we start from a fully matching setting, apply 1-level substructuring by cross-insertion, and choose all non-hanging vertices as primal DOF, then it is guaranteed that there are at least two primal DOF on each subdomain. The example in Fig. 7(b) illustrates the positions of primal DOF after two such 1-level substructuring steps. Note that, depending on the location of the refined area, 1-level substructuring can effect neighbouring subdomains. This disadvantage is accepted as a trade-off for avoiding an involved treatment of hanging vertices.

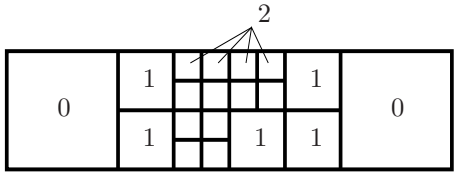


Figure 8: Refinement levels of subdomains (initial stetting as in Fig. 3(a)).

Note that the discretization is only C^0-continuous along subdomain interfaces. By substructuring a subdomain, new interfaces are introduced, and thereby the discretization is changed.

5.4. Preconditioning in the presence of hanging knots

As mentioned in Section 5.1, when we have hanging knots, the coupling matrix $\mathbf{B}^{(i)}$ is not a signed Boolean

matrix any more. The Dirichlet preconditioner that was defined in (40) can still be applied in settings with hanging knots. However, as already mentioned in [22] in the context of mortar discretizations, while the asymptotic behaviour of the condition number remains the same, the condition number itself increases. This can also be observed in the numerical tests with the IETI-DP method (see Section 6 for the results).

We now adapt the preconditioner for settings with hanging knots by replacing the scaling matrix $\mathbf{D}^{(i)}$ in (40) by a modified diagonal matrix $\mathbf{D}_A^{(i)}$. Its entries are defined as follows:

- $\left(\mathbf{D}_A^{(i)}\right)_{kk} = 1/\operatorname{mult}(k)$, if λ_k corresponds to a fully matching interface. Here, $\operatorname{mult}(k)$ is as defined in Section 4.4.
- $\left(\mathbf{D}_A^{(i)}\right)_{kk} = 1$, if λ_k corresponds to an interface with hanging knots and $\Omega^{(i)}$ is the master subdomain.
- $\left(\mathbf{D}_A^{(i)}\right)_{kk} = 0$, otherwise.

The preconditioner $\mathbf{M}_A^{-1}$ for the case with hanging knots is defined analogous to (40) by

$$\mathbf{M}_A^{-1} = \sum_{i=1}^{N} \mathbf{D}_A^{(i)}\, \mathbf{B}^{(i)} \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{S}_{BB}^{(i)} \end{pmatrix} \mathbf{B}^{(i)^T}\, \mathbf{D}_A^{(i)}.$$

As it will be reported in Section 6, this preconditioner leads to lower condition numbers as compared to $\mathbf{M}^{-1}$ from Section 4.4 in settings with hanging knots. Note that we have $\mathbf{M} = \mathbf{M}_A$ in fully matching settings.

6. Numerical examples

In this section, we present three numerical test examples for the IETI-DP method. We test the method and the refinement options presented in Section 5, and we study the performance of the proposed preconditioners. The condition numbers in the following tables were computed numerically using the Lanczos method.

6.1. Bracket under load

Our first example is a linearized elasticity problem (type (II) in Section 2.1). We consider the two geometries displayed in Fig. 9(a) and Fig. 10(a), where the first one, which is taken from an illustration in [10], has a rounded reentrant corner, and the second one has a sharp reentrant corner. We refer to these geometries as case (A) and case (B), respectively.

In Fig. 9(a) and 10(a), we show the subdomain decomposition and indicate the boundary conditions. We fix the two lower holes by applying homogenous Dirichlet boundary conditions, while a constant downward pointing traction t_N with magnitude 1000N is applied at the walls of the rightmost hole. On the remaining boundaries, we apply homogenous Neumann boundary conditions. The material parameters are set to $E = 3 \cdot 10^7$kPa and $\nu = 0.3$. Note that the circular holes contained in the domains are represented exactly by NURBS geometry mappings of degree 2.

In Fig. 9(b) and 10(b), the calculated stress component σ_{11} is depicted for cases (A) and (B), respectively. Note that the scales are different, and that the scale in Fig. 10(b) has been cutoff below for a better visibility of the stress distribution. The results illustrate that the IETI-DP method can be applied to non-trivial geometries including holes and consisting of numerous subdomains.

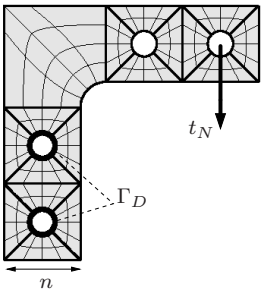
The condition numbers for the fully matching settings are given in the tables in Fig. 9(c) and 10(c). The column labeled n shows the number of knot spans in the direction indicated by the small arrows at the bottom in Fig. 9(a) and 10(a). The column $\#\boldsymbol{\lambda}$ shows the number of Lagrange multipliers, i.e. the size of the interface problem (34). The columns labeled $\kappa(\mathbf{F})$ and $\kappa(\mathbf{M}^{-1}\mathbf{F})$ display the condition numbers of the interface problems without preconditioner, and with preconditioner $\mathbf{M}^{-1}$ as defined in Section 4.4, respectively. The results show the expected, moderate growth in the preconditioned case.

In Fig. 10(b), the peak stress near the reentrant corner in case (B) is clearly visible. To obtain a better resolution of the peak stress, we introduce case (C), which is indicated in Fig. 11(a). Here, the subdomains near the corner have a finer discretization than the subdomains which are far from the corner, and we have interfaces with hanging knots. The number of knot spans on the finest and coarsest subdomain discretizations are denoted by n_{fine} and n_{coarse}, respectively. These numbers are measured in the directions indicated by the small arrows in Fig. 11(a). The ratio $n_{\text{fine}}/n_{\text{coarse}} = 4$ is the same for all chosen meshes. The condition numbers for this setting with hanging knots are presented in Fig. 11(b). Clearly, the preconditioner $\mathbf{M}_A^{-1}$ defined in Section 5.4 performs better than $\mathbf{M}^{-1}$ from Section 4.4. The stress component σ_{11} is not plotted for case (C), since it is the same as in case (B), Fig. 10(b). The energy norm of the numerical solutions in case (B) and case (C) is compared in Fig. 12. It shows that for given DOF a faster convergence can be achieved by local h-refinement.

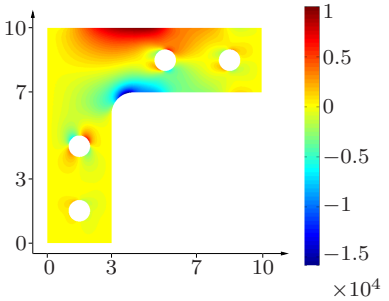
6.2. Bending of a cantilever

We now consider a linearized elasticity problem (type (II) in Section 2.1) on a cantilever of length L and thickness D. It is fixed at $x = 0$ and subject to a parabolic traction at $x = L$ with resultant P as illustrated in Fig. 13(a). We choose the parameters as follows: $L = 48$m, $D = 12$m, $E = 3 \cdot 10^7$kPa, $\nu = 0.3$, and $P = 1000$N.

An analytical solution for the displacement field $u =$



(a) Setting in case (A).

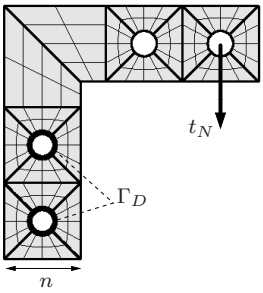


(b) Stress component σ_{11}.

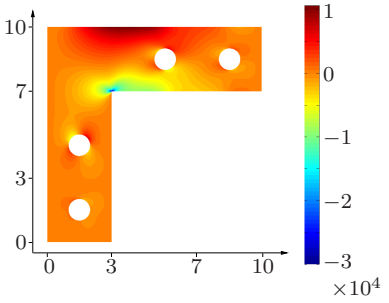
n	$\#\boldsymbol{\lambda}$	$\kappa(\mathbf{F})$	$\kappa(\mathbf{M}^{-1}\mathbf{F})$
8	464	173.95	39.60
16	784	470.09	67.98
32	1424	1230.11	105.87
64	2704	3074.62	154.10

(c) Condition numbers.

Figure 9: Case (A), bracket with rounded reentrant corner.



(a) Setting in case (B).

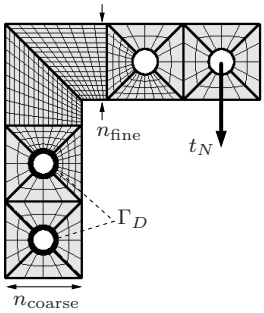


(b) Stress component σ_{11}.

n	$\#\boldsymbol{\lambda}$	$\kappa(\mathbf{F})$	$\kappa(\mathbf{M}^{-1}\mathbf{F})$
8	484	174.17	51.72
16	820	509.39	85.71
32	1492	1388.58	130.24
64	2836	3543.44	185.42

(c) Condition numbers.

Figure 10: Case (B), bracket with sharp reentrant corner.



(a) Setting in case (C).

n_{coarse}	n_{fine}	$\#\boldsymbol{\lambda}$	$\kappa(\mathbf{F})$	$\kappa(\mathbf{M}^{-1}\mathbf{F})$	$\kappa(\mathbf{M}_A^{-1}\mathbf{F})$
4	16	428	469.89	344.72	86.40
8	32	708	1324.01	542.19	130.75
16	64	1268	3419.50	820.16	185.40
32	128	2388	8448.24	1170.29	258.89

(b) Condition numbers.

Figure 11: Case (C), bracket with sharp reentrant corner and local h-refinement near the corner. The ratio $n_{\text{fine}}/n_{\text{coarse}} = 4$ is the same for all chosen meshes.The stress component σ_{11} as as in Fig. 10(b).

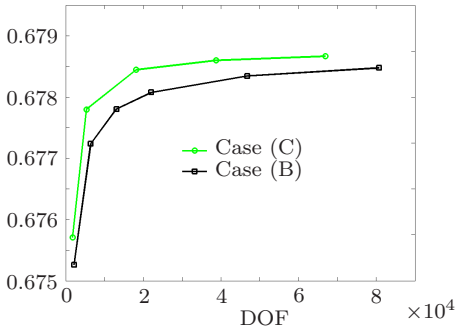


Figure 12: Comparison of the energy norms of the discrete solutions $\|u_h\|_E$ in cases (B) and (C).

$(u_1, u_2)^T$ can be found, e.g., in [28, 38, 48]:

$$\begin{aligned} u_1 =& \frac{Py}{6E_0 I}\left((6L-3x)x + (2+\nu_0)\Big(y^2 - \frac{D^2}{4}\Big)\right), \\ u_2 =& -\frac{P}{6E_0 I}\left(3\nu_0 y^2(L-x) + (4+5\nu_0)x\frac{D^2}{4}\right. \\ & \left. + (3L-x)x^2\right), \end{aligned}$$

where $I = D^3/12$ is the moment of inertia of the cross section of the cantilever. When we consider the plane stress problem, we set $E_0 = E$ and $\nu_0 = \nu$. For the plane strain problem, we set $E_0 = E/(1-\nu^2)$ and $\nu_0 = \nu/(1-\nu)$. Then, in both cases, the resulting exact stress components are as follows:

$$\begin{aligned} \sigma_{11} &= \frac{P(L-x)y}{I}, \\ \sigma_{12} &= -\frac{P}{2I}\left(\frac{D^2}{4} - y^2\right), \\ \sigma_{22} &= 0. \end{aligned}$$

We apply exact Dirichlet boundary conditions at the boundary $x = 0$, and the exact traction t_N as Neumann boundary condition at $x = L$. The remaining boundaries at $y = \pm D/2$ are free of traction.

Note that the exact displacement field $u = (u_1, u_2)^T$ is a cubic polynomial. By using basis functions of degree 3 and due to the simple geometry mappings, the exact solution u is in V_h. We verify that the IETI-DP method is capable of retrieving the exact solution, despite the fact that we enforce only C^0-continuity across interfaces. Also, in order to demonstrate the performace of the IETI-DP method in settings with hanging knots, we refine randomly chosen subdomains (note that the exact solution is smooth and does not have peaks or singularities, therefore local refinement is not induced by the problem). In Fig. 13(c)–(g), the thick lines indicate the subdomain decomposition, while the thin lines schematically indicate how fine the discretizations of the respective subdomains are, and whether the setting is fully matching or not. The numerical tests confirm that, for all considered discretizations, the calculated numerical solution is exact (up to the accuracy to which we solve (34) for $\boldsymbol{\lambda}$), even though only C^0-continuity is imposed in the presented IETI-DP method.

The tables presented in Fig. 13(c)–(g) show that the condition numbers behave as expected. The columns labeled n_{coarse} and n_{fine} indicate the number of knot spans in one direction on the coarse and fine discretizations, respectively. The other columns are labeled as in the previous example. In the settings with hanging knots presented in Fig. 13(e)–(g), it can be observed that the condition number of $\mathbf{M}^{-1}\mathbf{F}$ grows slower than in the unpreconditioned case. The absolute value of $\kappa(\mathbf{M}^{-1}\mathbf{F})$, however, is larger than $\kappa(\mathbf{F})$ on coarse discretizations (cf. the discussion in the beginning of Section 5.4). The preconditioner $\mathbf{M}_A^{-1}$ performs much better in all cases with hanging knots.

6.3. Poisson problem on Yeti's footprint

In our third example, we solve the Poisson problem $-\Delta u = f$ (type (I) in Section 2.1 with $\alpha = 1$) on the physical domain Ω resembling the footprint of a Yeti. The domain is shown in Fig. 14(a) and consists of 21 subdomains. We set Dirichlet boundary conditions at the big toe, and Neumann boundary conditions everywhere else. The boundary conditions and the right hand side f are determined by the exact solution

$$u(x,y) = \begin{cases} (R - r(x,y))^4 + y/10, & \text{if } r(x,y) < R, \\ y/10, & \text{else,} \end{cases}$$

where $r(x,y) = |(x,y) - (x_0,y_0)|$. This solution u is constructed in such a way that it has a peak at (x_0, y_0). We set $R = 1$ and $(x_0, y_0) = (2.6, 2.7)$ (see Fig. 14(a)).

Since the discussion of a posteriori error estimation is not in the scope of this paper, and because we know the exact solution u, we apply adaptive refinement based on the exact error. For each subdomain $\Omega^{(i)}$, we calculate the local error in the energy norm $\eta^{(i)} = \|u - u_h\|_{E,\Omega^{(i)}}$. Then we mark all subdomains for refinement, for which

$$\eta^{(i)} \geq 0.1 \max\{\eta^{(j)},\ j = 1, \ldots, N\}$$

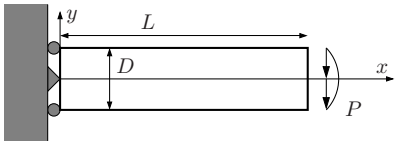
holds. In each such refinement step, we apply uniform h-refinement on the marked subdomains. The global mesh after 5 such refinement steps is shown in Fig. 14(b). In Fig. 14(c), we compare the error obtained by global uniform refinement and by the described adaptive refinement. As expected, for a given number of DOF a more accurate solution can be achieved by adaptive, local h-refinement, as compared to global refinement in a fully matching setting.

In Fig. 14(d), the condition numbers for the fully matching setting are presented. The column labeled "DOF" indicates the global number of DOF. In Fig. 14(e), the condition numbers in the adaptive refinement are shown.

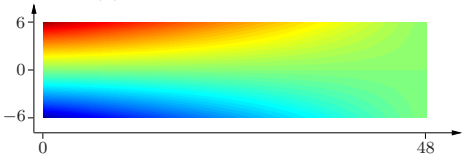
7. Conclusion

We have proposed the IETI method which combines the ideas and advantages of IGA and the FETI method. We preserve the exact geometry representation from the coarsest discretization level, thereby eliminating the need for data transformation, and also eliminating consequent approximation errors in the geometry. At the same time, we apply the techniques from FETI methods to couple NURBS patches and to solve the presented model problems on complicated computational domains which consist of many NURBS patches and may contain holes.

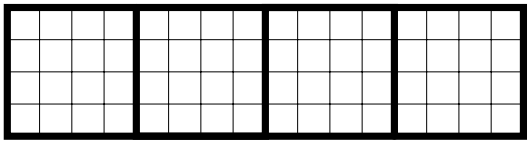
We have discussed the coupling of interfaces with hanging knots, thereby introducing options for local refinement. These can be applied using only NURBS basis functions with tensor-product structure, without the need for involved local NURBS-refinement techniques.



(a) Problem setting.

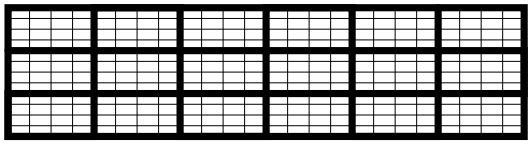


(b) Stess component σ_{11}.



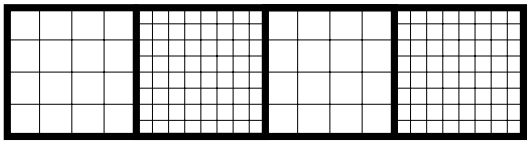
n	$\#\boldsymbol{\lambda}$	cond. numbers $\kappa(\mathbf{F})$	$\kappa(\mathbf{M}^{-1}\mathbf{F})$
8	88	33.80	11.81
16	152	77.02	14.99
32	280	170.81	18.38
64	536	378.38	23.49

(c) Fully matching setting with 4 subdomains.



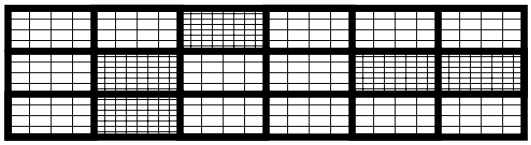
n	$\#\boldsymbol{\lambda}$	$\kappa(\mathbf{F})$	$\kappa(\mathbf{M}^{-1}\mathbf{F})$
8	700	53.87	13.50
16	1100	116.68	16.79
32	2140	264.72	20.38
64	4060	595.88	24.21

(d) Fully matching setting with 18 subdomains.



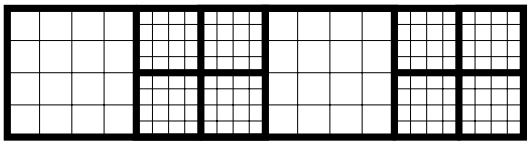
n_{coarse}	n_{fine}	$\#\boldsymbol{\lambda}$	$\kappa(\mathbf{F})$	$\kappa(\mathbf{M}^{-1}\mathbf{F})$	$\kappa(\mathbf{M}_A^{-1}\mathbf{F})$
8	16	136	66.03	109.10	28.43
16	32	248	157.41	146.07	35.53
32	64	472	358.85	184.14	43.02
64	128	920	795.88	224.55	51.11

(e) Setting with 4 subdomains, 2 of which are h-refined.



n_{coarse}	n_{fine}	$\#\boldsymbol{\lambda}$	$\kappa(\mathbf{F})$	$\kappa(\mathbf{M}^{-1}\mathbf{F})$	$\kappa(\mathbf{M}_A^{-1}\mathbf{F})$
4	8	580	46.38	74.78	31.02
8	16	940	102.90	109.54	38.59
16	32	1660	231.74	142.50	47.01
32	64	3100	530.70	175.41	55.94

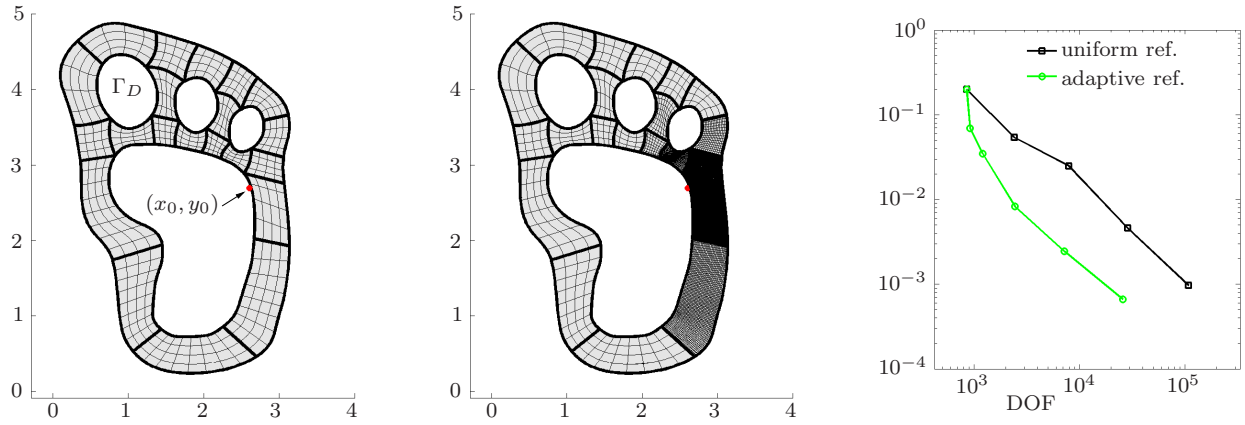
(f) Setting with 18 subdomains, 5 of which are h-refined.



n	$\#\boldsymbol{\lambda}$	$\kappa(\mathbf{F})$	$\kappa(\mathbf{M}^{-1}\mathbf{F})$	$\kappa(\mathbf{M}_A^{-1}\mathbf{F})$
8	338	184.11	351.08	98.71
16	578	387.44	433.00	141.50
32	1058	813.77	532.37	194.07
64	2018	1733.71	647.72	257.28

(g) Setting with 10 subdomains.

Figure 13: Bending of a cantilever, problem setting and discussed cases.



(a) Problem setting on Yeti's footprint.

(b) Discretization after 5 refinement steps.

(c) Exact errors in the energy norm $\|u - u_h\|_E$ vs. DOF.

n	DOF	$\#\boldsymbol{\lambda}$	$\kappa(\mathbf{F})$	$\kappa(\mathbf{M}^{-1}\mathbf{F})$
4	852	180	16.26	5.52
8	2420	292	35.25	7.55
16	7956	516	81.37	9.90
32	28628	964	188.51	12.49
64	108372	1860	431.61	15.36

(d) Condition numbers, fully matching setting with global refinement.

ref. steps	DOF	$\#\boldsymbol{\lambda}$	$\kappa(\mathbf{F})$	$\kappa(\mathbf{M}^{-1}\mathbf{F})$	$\kappa(\mathbf{M}_A^{-1}\mathbf{F})$
0	852	180	16.26	5.52	5.52
1	916	188	22.92	39.14	13.99
2	1204	212	51.94	68.62	19.63
3	2444	268	119.36	111.01	25.94
4	7196	384	272.22	174.72	33.13
5	25756	628	612.87	317.69	41.14

(e) Condition numbers, adaptive refinement.

Figure 14: Yeti's footprint with adaptive refinement.

Numerical examples demonstrated the performance of the IETI-DP method and of the proposed preconditioners, both in fully matching settings, and in settings with hanging knots.

However, in this paper, we have only discussed geometrically conforming subdomain interfaces. Furthermore, we have assumed that these interfaces are either fully matching, or that the discretization on one side of the interface is a refinement of the other. The treatment of more general interfaces, including interfaces that are not necessarily conforming (i.e. that may have small gaps or overlaps) is open for future work. Further interesting issues for future research on the IETI method include its extension to three-dimensional problems, as well as the incorporation of sophisticated subdomain solvers, such as wavelet solvers.

Acknowledgements

The authors gratefully acknowledge the support from the Austrian National Science Foundation FWF through the project P21516-N18, the European Union through the 7th Framework Programme, project 218536 "EXCITING", and the Austrian Academy of Sciences (ÖAW).

[1] M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, and A.-V. Vuong. Swept Volume Parameterization for Isogeometric Analysis. In *Proc. IMA Int. Conf. Mathematics of Surfaces XIII*, pages 19–44. Springer, 2009.

[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] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Comput. Methods Appl. Mech. Engrg.*, 199(5-8):229–263, 2010.

[4] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Comput. Methods Appl. Mech. Engrg.*, 199:229–263, 2010.

[5] L. Beirão da Veiga, A. Buffa, J. Rivas, and G. Sangalli. Some estimates for h-p-k-refinement in isogeometric analysis. *Numerische Mathematik*, 118:271–305, 2005.

[6] L. Beirão da Veiga, A. Buffa, D. Cho, and G. Sangalli. IsoGeometric analysis using T-splines on two-patch geometries. *Comput. Methods Appl. Mech. Engrg.*, 200(21-22):1787–1803, 2011.

[7] A. Buffa, G. Sangalli, and R. Vázquez. Isogeometric analysis in electromagnetics: B-splines approximation. *Comput. Methods Appl. Mech. Engrg.*, 199(17-20):1143–1152, 2010.

[8] E. Cohen, T. Martin, R.M. Kirby, T. Lyche, and R.F. Riesenfeld. Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 199(5-8):334 – 356, 2010.

[9] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Comput. Methods Appl. Mech. Engrg.*, 196:4160–4183, 2007.

[10] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Comput. Methods Appl. Mech. Engrg.*, 195:5257–5296, 2006.

[11] T.A. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2006.

[12] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng. Polynomial splines over hierarchical T-meshes. *Graphical Models*, 70:76–86, 2008.

[13] M.R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric

analysis by local h-refinement with T-splines. *Comput. Methods Appl. Mech. Engrg.*, 199(5-8):264–275, 2010.

[14] Z. Dostál, D. Horák, and R. Kučera. Total FETI – An easier implementable variant of the FETI method for numerical solution of elliptic PDE. *Commun. Numer. Methods Eng.*, 12:1155–1162, 2006.

[15] T. Elguedj, Y. Bazilevs, V.M. Calo, and T.J.R. Hughes. $\bar{B}$ and $\bar{F}$ projection methods for nearly incompressible linear and nonlinear elasticity and plasticity using higher-order NURBS elements. *Comput. Methods Appl. Mech. Eng.*, 197(33-40):2732–2762, 2008.

[16] C. Farhat. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Meth. Engng*, 32:1205–1227, 1991.

[17] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI-method – part I: A faster alternative to the two-level FETI method. *Int. J. Numer. Meth. Engng.*, 50:1523–1544, 2001.

[18] C. Farhat, J. Mandel, and F.X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Comput. Methods Appl. Mech. Engrg.*, 115:265–385, 1994.

[19] D. Großmann and B. Jüttler. Volumetric Geometry Reconstruction of Turbine Blades for Aircraft Engines. In *Curve and Surface Design: Avignon 2010*. Springer, 2011. to appear.

[20] T.J.R. Hughes, J. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Engrg.*, 194(39-41):4135–4195, 2005.

[21] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 199(5-8):301–313, 2010.

[22] H.H. Kim and C.-O. Lee. A preconditioner for the FETI-DP formulation with mortar methods in two dimensions. *SIAM Journal on Numerical Analysis*, 42(5):2159–2175, 2005.

[23] A. Klawonn, L. Pavarino, and O. Rheinbach. Spectral element FETI-DP and BDDC preconditioners with multi-element subdomains. *Comput. Meth. Appl. Mech. Engrg.*, 198:511–523, 2008.

[24] A. Klawonn, O. Widlund, and M. Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM J. Numer. Anal.*, 40:159–179, 2002.

[25] A. Klawonn and O. B. Widlund. A domain decomposition method with Lagrange multipliers and inexact solvers for linear elasticity. *SIAM J. Sci. Comput.*, 22(4):1199–1219, 2000.

[26] A. Klawonn and O. B. Widlund. FETI and Neumann-Neumann iterative substructuring methods: Connections and new results. *Comm. Pure Appl. Math.*, 54(1):57–90, 2001.

[27] A. Klawonn and O. B. Widlund. Dual-primal FETI methods for linear elasticity. *Comm. Pure Appl. Math.*, 59(11):1523–1572, 2006.

[28] S.K. Kleiss, B. Jüttler, and W. Zulehner. Enhancing isogeometric analysis by a finite element-based local refinement strategy. *Comput. Methods Appl. Mech. Engrg.*, 213-216:168–182, 2012.

[29] U. Langer and C. Pechstein. Coupled finite and boundary element tearing and interconnecting solvers for nonlinear potential problems. *Z. Angew. Math. Mech.*, 86(12):915–931, 2006.

[30] U. Langer and O. Steinbach. Boundary element tearing and interconnecting methods. *Computing*, 71(3).

[31] X. Li, J. Deng, and F. Chen. Polynomial splines over general T-meshes. *The Visual Computer*, 26:277–286, 2010.

[32] X. Li and M.A. Scott. On the nesting behavior of T-splines. Technical Report ICES REPORT 11-13.

[33] X. Li, J. Zheng, T.W. Sederberg, T.J.R. Hughes, and M.A. Scott. On linear independence of T-splines. (ICES REPORT 10-14).

[34] J. Mandel and R. Tezaur. Convergence of a substructuring method with Lagrange multipliers. *Numer. Math.*, 73:473–487, 1996.

[35] J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. *Numerische Mathematik*, 88:543–558, 2001.

[36] T. Martin and E. Cohen. Volumetric parameterization of complex objects by respecting multiple materials. *Computers & Graphics*, 34(3):187–197, 2010.

[37] T. Martin, E. Cohen, and R.M. Kirby. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Comput. Aided Geom. Design*, 26(6):648–664, 2009.

[38] N. Nguyen-Thanh, H. Nguyen-Xuan, S.P.A. Bordas, and T. Rabczuk. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Comput. Methods Appl. Mech. Engrg.*, 200(21-22):1892–1908, 2011.

[39] P.N. Nielsen, A.R. Gersborg, J. Gravesen, and N.L. Pedersen. Discretizations in isogeometric analysis of Navier-Stokes flow. *Comput. Methods Appl. Mech. Engrg.*, 200(45-46):3242–3253, 2011.

[40] G. Of and O. Steinbach. The all-floating boundary element tearing and interconnectiong method. *J. Num. Math.*, 17(4), 2009.

[41] L. Piegl and W. Tiller. *The NURBS book*. Springer-Verlag, London, UK, 1995.

[42] D. Rixen and C. Farhat. Preconditioning the FETI method for problems with intra- and inter-subdomain coefficient jumps. In P. E. Bjørstad, M. Espedal, and D. E. Keyes, editors, *Proceedings of 9th International Conference on Domain Decomposition*, pages 472–479, 1998.

[43] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003.

[44] M.A. Scott, T.W. Sederberg X. Li, and T.J.R. Hughes. Local refinement of analysis-suitable T-splines. Technical Report ICES REPORT 11-06.

[45] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, and N.S. North. T-spline simplification and local refinement. *ACM Transactions on Graphics*, 23(3), 2004.

[46] T.W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Transactions on Graphics*, 22(3):161–172, 2003.

[47] T. Takacs and B. Jüttler. Existence of stiffness matrix integrals for singularly parameterized domains in isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 200:3568–3582, 2011.

[48] S.P. Timoshenko and J.N. Goodier. *Theory of Elasticity, 3rd edition*. McGraw-Hill Book Company, New York, 1970.

[49] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*. Springer-Verlag, Berlin, 2005.

[50] U. Trottenberg, C.W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.

[51] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 200(49-52):3554 – 3567, 2011.

[52] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Parameterization of computational domain in isogeometric analysis: Methods and comparison. *Comput. Methods Appl. Mech. Engrg.*, 200(23-24):2021 – 2031, 2011.