NFN - Nationales Forschungsnetzwerk

Geometry + Simulation

http://www.gs.jku.at

Overlapping Multi-Patch Structures in Isogeometric Analysis

Somayeh Kargaran, Bert Jüttler, Stefan K. Kleiss, Angelos Mantzaflaris, Thomas Takacs

G+S Report No. 82

April 2019

Der Wissenschaftsfonds.

JOHANNES KEPLER UNIVERSITY LINZ

Overlapping Multi-Patch Structures in Isogeometric Analysis

S. Kargaran[a], B. Jüttler[b], S. K. Kleiss[a], A. Mantzaflaris[a,b], T. Takacs[b]

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

Abstract

In isogeometric analysis the domain of interest is usually represented by B-spline or NURBS patches, as they are present in standard CAD models. In order to avoid trimming, complicated domains can be represented as a union of simple overlapping subdomains, parameterized by single spline patches. Numerical simulation on such complicated domains is a serious challenge in IGA.

In this paper, we propose a non-iterative, robust and efficient method. The computational domain is represented as a collection of B-spline based geometries with overlaps. Consequently, the problem is divided into several sub-problems, which are coupled in an appropriate way. The resulting system can be solved directly in a single step. We compare the proposed method with iterative Schwarz domain decomposition approaches and explore the advantages of our method, especially when handling subdomains with small overlaps.

We will show that the problems can be solved on overlapping patches by a simple non-iterative method, without using trimming. Summing up, our method significantly simplifies the domain parameterization problem. The performance of the proposed method is demonstrated by several numerical experiments in two and three dimensions.

Keywords: isogeometric analysis, coupling, multi-patch, overlaps, Schwarz method

1. Introduction

Isogeometric analysis (IGA) is a computational approach, introduced by Hughes et al. [1]. IGA connects the technology of computer aided design (CAD) with numerical simulation via finite element analysis, e.g., [2, 3, 4]. In IGA, the same basis functions (typically tensor-product B-splines) are used for describing the geometry as well as for the numerical analysis. Hence, simulations can be performed directly on the geometry representation of CAD models.

Complex domains can be constructed as a union of several subdomains with interfaces or overlaps. Performing numerical simulations on such domains is not straightforward since we need to couple the patch-wise discretization in an appropriate way.

There exist several approaches for solving a PDE on multi-patch domains that consist of mutually disjoint patches. For matching or nested interfaces, the smoothness of the solution across patch boundaries can be enforced strongly, simply by identifying the degrees of freedom on the interfaces between the subdomains [5, 6, 7]. Depending on the order of the considered PDE, higher order of continuity may be needed [8, 9, 10].

The situation becomes more complicated if non-matching interfaces are present. In this situation, weak enforcement of the coupling constraints is needed. The mortar discretization method [11] can be employed to solve a PDE on the non-overlapping subdomains. In this case, the meshes on the subdomains are not matching on the interfaces and the continuity conditions are imposed using Lagrange multipliers. As an alternative to the mortar method, in [12, 13], the authors used a Galerkin discretization method for the non-overlapping subdomains. Instead of Lagrange multipliers, an interpolation scheme is used to enforce the continuity constraints on the non-matching interfaces. Also, the Nitsche method can be used to impose weak coupling conditions on the non-matching interfaces [14, 15, 16, 17, 18], similar to discontinuous Galerkin methods [19, 20, 21].

In geometric design, the construction of complex geometries makes frequent use of trimming operations. In the framework of IGA, this leads to computational domains represented as collections of trimmed tensor product B-spline patches [22, 23, 24]. These patches use only certain parts of the full tensor product domains, due to additional geometric boundaries specified by trimming curves or surfaces.

In order to perform numerical simulation on domains described by trimmed patches, one can convert them into untrimmed patches by suitably splitting the subdomains and generating new parameterizations [25]. For instance, triangular Bézier or Coons patches are used to convert a trimmed model to a watertight representation [26, 27]. However, the reparameterization is costly and may increase the number of the patches substantially. Also, it is not guaranteed to preserve the exactness of the geometry used for the simulation, which is considered to be one of the main assets of the isogeometric framework.

In order to avoid the conversion step, several techniques have been explored that deal directly with trimmed domain parameterizations in IGA, see [28] and the references cited therein. Applying IGA to trimmed domains creates three major challenges.

The first one is related to the stability of the basis functions near the trimmed boundary. Some of these basis functions, possessing supports that are cut by the geometric boundary, are not stable since their remaining support is very small. One way to resolve this issue is to add these functions to nearby stable basis functions. This concept of extended B-splines was introduced in [29]. The resulting modification of unstable basis functions is considered in [30, 31] to introduce the immersed and extended B-spline basis functions respectively. As an alternative approach, in [17], the authors use a finite cell method based on the fictitious domain approach to mitigate the influence of the cut basis functions around the trimming boundary on the analysis.

The second challenge is to devise efficient methods for numerical quadrature that are able to deal with functions defined on trimmed domains. Since numerical quadrature rules on boxes are easy to generate, it has been proposed to employ finer cells near the trimming boundary [17]. Alternatively, mesh deformation techniques can be introduced [32], in order to extend the quadrature rules from simple boxes to more general, deformed elements. The recent preprint [33] establishes special quadrature rules for trimmed geometries.

Finally, the coupling of adjacent trimmed domains needs to be dealt with efficiently. Most of the methods for coupling patch-wise discretizations are not applicable if trimmed patches are present. The only exception are Nitsche-based methods, such as [17], and dG-type approaches. The latter ones can be adapted to deal with overlapping subdomains directly [34].

Summing up, the presence of trimmed elements is a source of considerable difficulties. Several techniques have been developed to deal with them, which are based on different approaches, such as geometry preprocessing or Nitsche-type methods. In particular, we note that the proposed geometry processing (splitting the domain into the single patches, followed by a reparameterization) is not in line with the isogeometric paradigm of exact geometry representation. It may also lead to a large number of small subdomains.

In many cases, the use of trimming can be avoided by considering overlapping domains, see Fig. 1. Numerical simulation on domains of this type can be performed with the help of overlapping domain decomposition methods, see [35] and the references therein. Using the framework of isogeometric analysis, both *additive Schwarz* (ADS) and *multiplicative Schwarz* (MPS) domain decomposition methods have been explored [36], which are iterative approaches. In each iteration step, the PDE is solved separately on each subdomain. The sub-problems are coupled by imposing boundary conditions on the boundary of the overlap regions. Schwarz-type methods become costly for a large number of overlapping subdomains and require a high number of iterations if small overlaps are present.

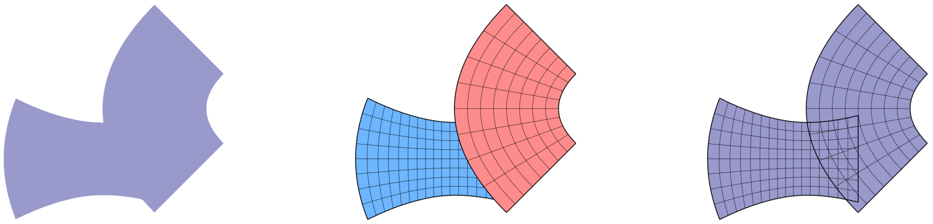


Figure 1: A non-four sided domain (left) can be represented either by non-overlapping, but trimmed, patches (center), or by untrimmed, but overlapping, patches (right).

Overlapping multi-patch structures are able to represent complex geometries, which are constructed as Boolean unions of patches. Such Boolean operations are common in CAD systems to design complex shapes. Consequently, trimmed interfaces can be replaced by overlaps in many situations. In order to perform numerical simulations on domains of this type, we propose in this paper the overlapping multi-patch (OMP) method, which is a non-iterative reformulation of the Schwarz method. More precisely, we show how to perform isogeometric simulations on overlapping multi-patch structures by numerically solving a single system. This system is derived by introducing suitable extension operators, which are used to couple the solutions on the individual subdomains.

For the Poisson equation, which serves as a model problem, we derive an equivalent reformulation in the continuous case, and we present numerical results which indicate that the isogeometric discretization preserves the coercivity for a particular choice of the extension operator.

We also use numerical experiments to compare the computational costs of the OMP method with the ADS and MPS methods. According to these results, the non-iterative reformulation provides a significant reduction of the overall costs and is far more robust with respect to the size of the overlaps.

The remainder of the paper is organized as follows. Section 2 introduces the overlapping multi-patch formulation, and shows that it is equivalent to the original one in the continuous case. The next section describes the isogeometric discretization via B-splines. We also set up the final system according to different extension operators. The existence and uniqueness of the discretized problem is studied via numerical experiments in Section 4, and Section 5 investigates the performance of the OMP method on different overlapping multi-patch domains. Finally we conclude the paper in Section 6.

2. The overlapping multi-patch formulation

In this section we introduce the geometry representation that we consider and derive the overlapping multi-patch formulation on a simplified model problem.

As a model problem we consider the Poisson problem on an open and bounded domain $\Omega \subset \mathbb{R}^d$ with a given right-hand side f, and zero Dirichlet boundary conditions. The approach we present in the following is not restricted to this specific problem. The method can be applied to different elliptic problems or other boundary conditions. Note that the main Theorem 1 is formulated for the Poisson problem but the proof can be extended to other variational formulations.

The physical domain Ω is formed by overlapping subdomains. We assume that the subdomains are open and that they only have pairwise overlaps, the intersection of three or more subdomains is always empty. Also, we do not allow interfaces between subdomains. It means

$$\Omega^i \cap \Omega^j \cap \Omega^k = \emptyset \text{ and } \overline{\Omega^i \cap \Omega^j} = \overline{\Omega^i} \cap \overline{\Omega^j}, \quad \forall i \neq j \neq k \neq i.$$

In order to keep the presentation simple, we introduce the formulation for two patches Ω^1 and Ω^2 only. In Section 5 we present some examples of domains having more than two patches with pairwise overlap.

2.1. Model problem

Assume that a physical domain $\Omega \subset \mathbb{R}^d$, is given. We consider the two bounded forms:

$$a(u,v) = \int_\Omega \nabla u \nabla v \,\mathrm{d}\boldsymbol{\xi} \quad \text{and} \quad \ell(v) = \int_\Omega f v \,\mathrm{d}\boldsymbol{\xi}, \tag{1}$$

then the weak form of the Poisson problem can be formulated as follows

Problem 1. *Find $u \in H_0^1(\Omega)$, such that*

$$a(u,v) = \ell(v) \qquad \text{for all } v \in H_0^1(\Omega). \tag{2}$$

For simplicity, we consider the weak form of the Poisson problem. Any other elliptic problem can be considered as well. Moreover, Problem 1 can be formulated for non-homogeneous Dirichlet boundary conditions.

In the following we define a formulation on overlapping multi-patch structures. This formulation will turn out to be equivalent to the initial Problem 1.

2.2. The overlapping multi-patch structure

We are interested in domains

$$\Omega = \Omega^1 \cup \Omega^2,$$

created by forming the union of two open subdomains with non-empty overlap

$$\Phi^{12} = \Omega^1 \cap \Omega^2 \neq \emptyset,$$

see Fig. 2. The boundary of the two subdomains is subdivided into the Dirichlet and coupling boundary

$$\Gamma_D^k = \partial\Omega^k \cap \partial\Omega \quad \text{and} \quad \Gamma_C^k = \partial\Omega^k \cap \Omega^\circ \quad (k = 1, 2),$$

respectively.

In order to exploit the special structure of the domain, we introduce two extension operators

$$M^k \colon H^1(\Omega^{k'}) \to H^1(\Omega^k),$$

which take a function from one patch and map it to the other. Here, for $k \in \{1,2\}$, k' is defined as the other index, i.e., $k' = 3 - k$. We will use this notation throughout. Note that the image of the projector has non-vanishing trace at the coupling boundary. The projector M^k is bounded and defined to take the value of a function on $\Omega^{k'}$ at the coupling boundary Γ_C^k and extend it into the patch Ω^k. More precisely, these operators are assumed to satisfy the conditions

$$M^k v = \begin{cases} v & \text{on } \Gamma_C^k \\ 0 & \text{on } \Gamma_D^k. \end{cases} \tag{3}$$

Clearly, there are several possibilities to choose these operators, which will be discussed in more detail later.

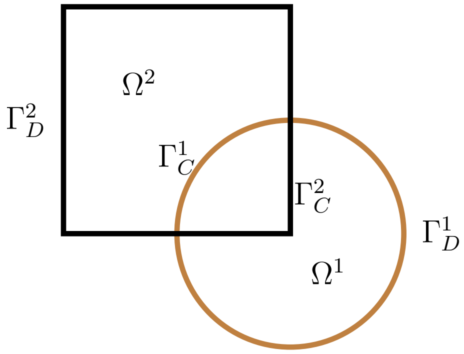


Figure 2: Two overlapping patches.

2.3. The overlapping multi-patch problem

To define the local problems, we consider the restrictions of the two forms $a(u,v)$ and $\ell(v)$ in (1) to the two subdomains,

$$a^k(u^k,v^k) = \int_{\Omega^k} \nabla u^k \nabla v^k \,\mathrm{d}\xi \quad \text{and} \quad \ell^k(v^k) = \int_{\Omega^k} f v^k \,\mathrm{d}\xi,$$

which are defined for $u^k, v^k \in H^1(\Omega^k)$. For $(u^1,u^2),(v^1,v^2) \in H^1(\Omega^1) \times H^1(\Omega^2)$, we define

$$A((u^1,u^2),(v^1,v^2)) = a^1(u^1,v^1) + a^2(u^2,v^2)$$

and

$$L(v^1,v^2) = \ell^1(v^1) + \ell^2(v^2).$$

Assumption 1. *The operators $M^1M^2\colon H^1(\Omega^1) \to H^1(\Omega^1)$ and $M^2M^1\colon H^1(\Omega^2) \to H^1(\Omega^2)$ are contractions.*

Since $M^kM^{k'}$ is a contraction, we can define $M^{kk'} : H^1(\Omega^k) \to H^1(\Omega^k)$ as

$$M^{kk'} = \sum_{\ell=0}^{\infty} (M^kM^{k'})^\ell = (I^k - M^kM^{k'})^{-1}, \tag{4}$$

where I^k is the identity operator mapping $H^1(\Omega^k)$ onto itself. Sufficient conditions for Assumption 1 are

$$M^1M^2 = M^2M^1 = 0. \tag{5}$$

Now we can formulate the localized weak form of the model problem:

Problem 2. *Find* $(u_0^1, u_0^2) \in H_0^1(\Omega^1) \times H_0^1(\Omega^2)$*, such that*

$$A((u^1, u^2), (v_0^1, v_0^2)) = L(v_0^1, v_0^2) \qquad \text{for all } (v_0^1, v_0^2) \in H_0^1(\Omega^1) \times H_0^1(\Omega^2), \tag{6}$$

where

$$u^k = M^{kk'}(u_0^k + M^k u_0^{k'}) \quad \text{for } k \in \{1, 2\}. \tag{7}$$

We can show that this form of the problem is equivalent to the original one:

Theorem 1. *Consider functions* $u \in H_0^1(\Omega)$ *and* $(u_0^1, u_0^2) \in H_0^1(\Omega^1) \times H_0^1(\Omega^2)$*.*

(i) If u *is a solution of Problem 1, then* $(u|_{\Omega^1} - M^1(u|_{\Omega^2}), u|_{\Omega^2} - M^2(u|_{\Omega^1}))$ *is a solution of Problem 2.*

(ii) If (u_0^1, u_0^2) *is a solution of Problem 2, then*

$$u = \begin{cases} M^{12}\,(u_0^1 + M^1 u_0^2) & \text{on } \Omega^1 \\ M^{21}\,(u_0^2 + M^2 u_0^1) & \text{on } \Omega^2 \setminus \Phi^{12} \end{cases} \tag{8}$$

is a solution of Problem 1.

(iii) The solution of Problem 2 is unique.

Proof. See the proof in the Appendix. □

Problem 2 can be rewritten as follows.

Problem 3. *Find* $(u_0^1, u_0^2) \in H_0^1(\Omega^1) \times H_0^1(\Omega^2)$ *and* $(u_M^1, u_M^2) \in H^1(\Omega^1) \times H^1(\Omega^2)$ *such that*

$$\begin{aligned} A((u_0^1 + u_M^1, u_0^2 + u_M^2), (v_0^1, v_0^2)) &= L(v_0^1, v_0^2), \quad \forall (v_0^1, v_0^2) \in H_0^1(\Omega^1) \times H_0^1(\Omega^2) && (9) \\ u_M^1 - M^1(u_0^2 + u_M^2) &= 0 \quad \text{on } \Omega^1 && (10) \\ u_M^2 - M^2(u_0^1 + u_M^1) &= 0 \quad \text{on } \Omega^2. && (11) \end{aligned}$$

In the formulation of Problem 3, we replace $M^k u^{k'}$ with u_M^k and obtain the additional constraints from $u^k = u_0^k + M^k u^{k'}$. This is the problem that we want to study and discretize in the following. For the variational equation (9) we will use a standard Galerkin discretization. The two equations (10) and (11), which implicitly define the extension operators, are discretized using finite differences as well as a collocation scheme at the coupling boundary.

3. Isogeometric discretization

In this section we briefly introduce B-spline basis functions and the local geometry mappings (see also [2]). To represent the discrete solutions we define the spaces on each subdomain by selecting the corresponding basis functions.

3.1. Geometry mapping

Let $\mathcal{S}^p_\Xi : (0,1) \to \mathbb{R}$ be a univariate B-spline space of degree p for a suitable open knot vector Ξ. Let $\hat{\beta}_{i,p}$, for $i = 1, \dots, n$, be the standard basis of the B-spline space $\mathcal{S}^p_\Xi$ (see, [37]).

For each patch Ω^k and each direction ξ_ν we have given a separate B-spline space $\mathcal{S}^p_{\Xi^k_\nu}$ and corresponding B-spline basis functions $\hat{\beta}^{k,\nu}_{i_\nu}$, with $1 \leq i_\nu \leq n^k_\nu$ for $k \in \{1,2\}$. We then define the tensor-product B-spline space $\mathcal{S}^k = \mathcal{S}^p_{\Xi^k_1} \otimes \ldots \otimes \mathcal{S}^p_{\Xi^k_d}$ with the basis functions

$$\hat{\beta}^k_i(\xi_1, \dots, \xi_d) = \hat{\beta}^{k,1}_{i_1}(\xi_1) \cdot \ldots \cdot \hat{\beta}^{k,d}_{i_d}(\xi_d),$$

where the index $i \in \mathcal{I}^k$ corresponds to the tuple $(i_1, \dots, i_d)$. So $\mathcal{I}^k$ is the index set

$$\{(i_1, \dots, i_d) \in \mathbb{Z}^d : 1 \leq i_\nu \leq n^k_\nu \text{ for all } \nu = 1, \dots, d\}. \tag{12}$$

We assume that the physical domains Ω^k are images of spline mappings $\mathbf{G}^k \in (\mathcal{S}^k)^d$, i.e., $\Omega^k = \mathbf{G}^k(\hat{\Omega}^k)$ for $k = 1, 2$, and $\hat{\Omega}^k = (0,1)^d$ are the corresponding parametric domains.

3.2. Isogeometric spaces

We can now define the isogeometric space on every patch as

$$V^k_h = \operatorname{span}\left\{\beta^k_i = \hat{\beta}^k_i \circ (\mathbf{G}^k)^{-1} \mid i \in \mathcal{I}^k\right\} \tag{13}$$

for $k \in \{1, 2\}$.

For every patch Ω^k we now define three subspaces of the full space V^k_h and corresponding index sets as follows:

- Interior functions:

$$V^k_{0h} = \operatorname{span}\left\{\beta^k_i \mid i \in \mathcal{I}^k_0\right\} = V^k_h \cap H^1_0(\Omega^k), \text{with } \mathcal{I}^k_0 = \left\{i \in \mathcal{I}^k \mid \beta^k_i = 0 \text{ on } \partial\Omega^k\right\},$$

- Coupling overlap functions:

$$V^k_{ch} = \operatorname{span}\left\{\beta^k_i \mid i \in \mathcal{I}^k_c\right\} \subset H^1(\Omega^k), \text{with } \mathcal{I}^k_c = \left\{i \in \mathcal{I}^k \mid \operatorname{supp}\beta^k_i \cap \Gamma^k_C \neq \emptyset\right\},$$

- Non-coupling overlap functions:

$$V^k_{\Phi h} = \operatorname{span}\left\{\beta^k_i \mid i \in \mathcal{I}^k_\Phi\right\} \subset V^k_{0h} \subset H^1_0(\Omega^k), \text{with } \mathcal{I}^k_\Phi = \left\{i \in \mathcal{I}^k \mid \operatorname{supp}\beta^k_i \subset \Phi^{12}\right\}.$$

Let moreover $V^k_{\Phi ch} = V^k_{ch} \oplus V^k_{\Phi h}$ and $\mathcal{I}^k_{\Phi c} = \mathcal{I}^k_c \cup \mathcal{I}^k_\Phi$. We define the descretized space

$$V_{0h} = V^1_{0h} \times V^2_{0h} \subset H^1_0(\Omega^1) \times H^1_0(\Omega^2),$$

which is used both as trial function space for (u^1_{0h}, u^2_{0h}) and as test function space.

3.3. The discretized problem

Using the spaces defined in the previous section, Problem 3 can be discretized as follows.

Problem 4. *Find $(u^1_{0h}, u^2_{0h}) \in V_{0h}$ and $(u^1_{Mh}, u^2_{Mh}) \in V^1_{\Phi ch} \times V^2_{\Phi ch}$ such that*

$$\begin{aligned} A((u^1_{0h} + u^1_{Mh}, u^2_{0h} + u^2_{Mh}), (v^1_{0h}, v^2_{0h})) &= L(v^1_{0h}, v^2_{0h}), \quad \forall (v^1_{0h}, v^2_{0h}) \in V_{0h} && (14)\\ u^1_{Mh} - M^1_h(u^2_{0h} + u^2_{Mh}) &= 0 \quad \text{on } \Omega^1 && (15)\\ u^2_{Mh} - M^2_h(u^1_{0h} + u^1_{Mh}) &= 0 \quad \text{on } \Omega^2, && (16)\end{aligned}$$

where M^1_h and M^2_h are suitable discretizations of the operators M^1 and M^2, respectively, which will be defined in Subsection 3.4.

Remark 1. Note that Problem 4 can be solved with non-zero Dirichlet boundary conditions as well.

Here the functions can be expressed as

$$u^k_{0h}(\mathbf{x}) = \sum_{i \in \mathcal{I}^k_0} c^k_i \beta^k_i(\mathbf{x}), \qquad u^k_{Mh}(\mathbf{x}) = \sum_{i \in \mathcal{I}^k_{\Phi c}} d^k_i \beta^k_i(\mathbf{x})$$

and

$$u^k_h(\mathbf{x}) = u^k_{0h}(\mathbf{x}) + u^k_{Mh}(\mathbf{x}),$$

for $\mathbf{x} = (x_1, \ldots, x_d) \in \Omega^k$. The variational equation (14) in Problem 4 is equivalent to

$$\begin{aligned} \sum_{i \in \mathcal{I}^1_0} c^1_i \underbrace{\int_{\Omega^1} \nabla\beta^1_i \, \nabla\beta^1_j \, \mathrm{d}\mathbf{x}}_{=:K^1_{ji}} + \sum_{i \in \mathcal{I}^1_{\Phi c}} d^1_i \underbrace{\int_{\Omega^1} \nabla\beta^1_i \, \nabla\beta^1_j \, \mathrm{d}\mathbf{x}}_{=K^1_{ji}} &= \underbrace{\int_{\Omega^1} f \, \beta^1_j \, \mathrm{d}\mathbf{x}}_{=:f^1_j} \quad \forall j \in \mathcal{I}^1_0 \\ \sum_{i \in \mathcal{I}^2_0} c^2_i \underbrace{\int_{\Omega^2} \nabla\beta^2_i \, \nabla\beta^2_j \, \mathrm{d}\mathbf{x}}_{=:K^2_{ji}} + \sum_{i \in \mathcal{I}^2_{\Phi c}} d^2_i \underbrace{\int_{\Omega^2} \nabla\beta^2_i \, \nabla\beta^2_j \, \mathrm{d}\mathbf{x}}_{=K^2_{ji}} &= \underbrace{\int_{\Omega^2} f \, \beta^2_j \, \mathrm{d}\mathbf{x},}_{=:f^2_j} \quad \forall j \in \mathcal{I}^2_0. \end{aligned}$$

Since $\mathcal{I}^k_{\Phi c} = \mathcal{I}^k_c \cup \mathcal{I}^k_\Phi$, we can write

$$\begin{aligned} \sum_{i \in \mathcal{I}^1_0} K^1_{ji} c^1_i + \sum_{i \in \mathcal{I}^1_c} K^1_{ji} d^1_i + \sum_{i \in \mathcal{I}^1_\Phi} K^1_{ji} d^1_i &= f^1_j \qquad \forall j \in \mathcal{I}^1_0 && (17)\\ \sum_{i \in \mathcal{I}^2_0} K^2_{ji} c^2_i + \sum_{i \in \mathcal{I}^2_c} K^2_{ji} d^2_i + \sum_{i \in \mathcal{I}^2_\Phi} K^2_{ji} d^2_i &= f^2_j, \qquad \forall j \in \mathcal{I}^2_0. && (18)\end{aligned}$$

We can write the matrices and vectors in block form as

$$\begin{pmatrix} K_0^1 & 0 & K_c^1 & 0 & K_\Phi^1 & 0 \\ 0 & K_0^2 & 0 & K_c^2 & 0 & K_\Phi^2 \end{pmatrix} \begin{pmatrix} \mathbf{c}_0^1 \\ \mathbf{c}_0^2 \\ \mathbf{d}_c^1 \\ \mathbf{d}_c^2 \\ \mathbf{d}_\Phi^1 \\ \mathbf{d}_\Phi^2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}^1 \\ \mathbf{f}^2 \end{pmatrix}, \tag{19}$$

where the block matrix is composed of sub-matrices $K_*^k = \left(K_{ji}^k\right)_{i\in\mathcal{I}_*^k, j\in\mathcal{I}_0^k}$, with $* \in \{0, c, \Phi\}$, and the vectors of coefficients are given as $\mathbf{c}_0^k = (c_i^k)_{i\in\mathcal{I}_0^k}$ and $\mathbf{d}_*^k = (d_i^k)_{i\in\mathcal{I}_*^k}$, with $* \in \{c, \Phi\}$.

3.4. Discrete extension operators

In the following we consider the discretization of equations (15)–(16). The continuous extension operators have only to satisfy the condition (3). For the discrete operators we assume $\mathrm{Im}(M_h^k) \subseteq V_{\Phi ch}^k$. We call M_h^k *minimal extension operator*, if $M_h^k \colon V_{0h}^{k'} \to V_{ch}^k$. There are different possible choices of minimal extension operators, which depend on the computation of the degrees of freedom related to the boundary of the overlap region.

As in the continuous case, we want the discrete extension operators to satisfy the following equations defined on the boundary of the overlap Φ^{12}

$$(M_h^k u_h^{k'})(\mathbf{x}) = u_{Mh}^k(\mathbf{x}) \approx u_h^{k'}(\mathbf{x}) \quad \text{on } \Gamma_C^k. \tag{20}$$

The equation is solved only approximately, by interpolating at a set of suitable points. Therefore, the related minimal extension operator is called *collocation-based extension operator* (CEO). This will be discussed in more detail in Section 3.4.1.

Recall that $M_h^k u_h^{k'} = u_{Mh}^k$, which we can write as a sum $u_{Mh}^k = u_{Mch}^k + u_{M\Phi h}^k$, where $u_{Mch}^k \in V_{ch}^k$ and $u_{M\Phi h}^k \in V_{\Phi h}^k$. Since $u_{M\Phi h}^k$ vanishes at Γ_C^k, only the non-vanishing term u_{Mch}^k is determined by equation (20). Hence, when considering CEO, the operator is already determined completely by the interpolation.

If $\mathrm{Im}(M_h^k) \supset V_{ch}^k$, then the function $u_{Mh}^k = M_h^k u_h^{k'}$ is obtained by solving (20) and additionally a discretization of the harmonic equation on the parameter domain, i.e.,

$$\Delta_{\boldsymbol{\xi}}[u_{Mh}^k \circ \mathbf{G}^k](\boldsymbol{\xi}) \approx 0 \quad \text{in } (\mathbf{G}^k)^{-1}(\Phi^{12}) \subset \hat{\Omega}^k. \tag{21}$$

We call this a *quasi-harmonic extension operator* (QEO). The details of this approach will be explained in Section 3.4.2.

3.4.1. Discretization by collocation

In the following we describe how to solve equation (20) by collocation. Let us recall that

$$u^k_{Mh}(\mathbf{x}) = \sum_{i \in \mathcal{I}^k_{\Phi c}} d^k_i \beta^k_i(\mathbf{x}),$$

so we obtain

$$u^k_{Mh}(\mathbf{x}) = \sum_{i \in \mathcal{I}^k_{\Phi c}} d^k_i \beta^k_i(\mathbf{x}) \approx \sum_{i \in \mathcal{I}^{k'}_0} c^{k'}_i \beta^{k'}_i(\mathbf{x}) + \sum_{i \in \mathcal{I}^{k'}_{\Phi c}} d^{k'}_i \beta^{k'}_i(\mathbf{x}) = u^{k'}_h(\mathbf{x}) \quad \text{on } \Gamma^k_C.$$

Since β^k_i vanish at Γ^k_C for $i \in \mathcal{I}^k_\Phi$, this reduces to

$$\sum_{i \in \mathcal{I}^k_c} d^k_i \beta^k_i(\mathbf{x}) \approx \sum_{i \in \mathcal{I}^{k'}_0} c^{k'}_i \beta^{k'}_i(\mathbf{x}) + \sum_{i \in \mathcal{I}^{k'}_c} d^{k'}_i \beta^{k'}_i(\mathbf{x}) \quad \text{on } \Gamma^k_C.$$

To determine the coefficients d^k_i for $i \in \mathcal{I}^k_c$ we define a set of suitable collocation points $(\mathbf{x}^k_\ell)_{\ell \in \mathcal{I}^k_c}$, with $\mathbf{x}^k_\ell \in \Gamma^1_C$, and obtain the set of equations

$$\sum_{i \in \mathcal{I}^k_c} d^k_i \beta^k_i(\mathbf{x}^k_\ell) = \sum_{i \in \mathcal{I}^{k'}_0} c^{k'}_i \beta^{k'}_i(\mathbf{x}^k_\ell) + \sum_{i \in \mathcal{I}^{k'}_c} d^{k'}_i \beta^{k'}_i(\mathbf{x}^k_\ell) \quad \text{for all } \ell \in \mathcal{I}^k_c. \tag{22}$$

As collocation points $\mathbf{x}^k_\ell$ one can use the images of the Greville points of the functions $\hat{\beta}^k_\ell$ under the geometry mapping $\mathbf{G}^k$.

Equation (22) can now be expressed in matrix form as follows,

$$C^k_c d^k_c = C^{k'}_0 c^{k'}_0 + C^{k'}_d d^{k'}_c, \tag{23}$$

where the matrices C^k_c, $C^{k'}_0$ and $C^{k'}_d$ are given as

$$\begin{aligned} (C^k_c)_{\ell i} &= \beta^k_i(\mathbf{x}^k_\ell), \quad && \ell \in \mathcal{I}^k_c, i \in \mathcal{I}^k_c \\ (C^{k'}_0)_{\ell i} &= \beta^{k'}_i(\mathbf{x}^k_\ell), \quad && \ell \in \mathcal{I}^k_c, i \in \mathcal{I}^{k'}_0 \\ (C^{k'}_d)_{\ell i} &= \beta^{k'}_i(\mathbf{x}^k_\ell), \quad && \ell \in \mathcal{I}^k_c, i \in \mathcal{I}^{k'}_c \end{aligned}$$

3.4.2. Discretization of the QEO

In the following we solve (21) with

$$u^k_{Mh}(\mathbf{x}) = \sum_{i \in \mathcal{I}^k_c} d^k_i \beta^k_i(\mathbf{x}) + \sum_{i \in \mathcal{I}^k_\Phi} d^k_i \beta^k_i(\mathbf{x}),$$

where we assume that the coefficients d_i^k, for $i \in \mathcal{I}_c^k$, in the first sum are completely determined by the equations (23).

To determine the coefficients d_i^k, for $i \in \mathcal{I}_\Phi^k$, we follow the approach presented in [38], by applying a finite difference discretization to equation (21) to get a system of linear equations

$$-d_i^k + \sum_{j \in \mathcal{N}(i)} \frac{d_j^k}{|\mathcal{N}(i)|} = 0 \quad \text{for } i \in \mathcal{I}_\Phi^k. \tag{24}$$

For $i \in \mathcal{I}_\Phi^k$ the set $\mathcal{N}(i) \subset \mathcal{I}^k$ is the set of neighboring indices of i. In 1D these are $i-1$ and $i+1$. In 2D (and similarly in higher dimensions) the neighboring indices of i corresponding to the pair (i_1, i_2) are given as $(i_1 - 1, i_2)$, $(i_1 + 1, i_2)$, $(i_1, i_2 - 1)$ and $(i_1, i_2 + 1)$, as depicted in Figure 3.

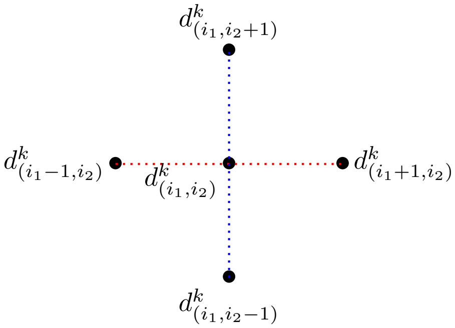


Figure 3: Visualization of the neighborhood $\mathcal{N}(i)$ for $i \sim (i_1, i_2)$.

We moreover assume $d_j^k = 0$ for $j \in \mathcal{N}(i) \setminus \mathcal{I}_{\Phi c}^k$, then equation (24) is equivalent to

$$-d_i^k + \sum_{j \in \mathcal{N}(i) \cap \mathcal{I}_{\Phi c}^k} \frac{d_j^k}{2^d} = 0 \quad \text{for } i \in \mathcal{I}_\Phi^k, \tag{25}$$

which can be written in matrix form as

$$D_c^k d_c^1 - D_\Phi^k d_\Phi^k = 0. \tag{26}$$

Remark 2. It should be noted that the sufficient conditions (5) are not always satisfied for the discrete operators M_h^k, i.e., $\|M_h^k M_h^{k'}\| \neq 0$. However, the operators are constructed in such a way that as $h \to 0$, the violation becomes less severe. Hence the contraction conditions (4) are satisfied for sufficiently small h.

3.5. Final system

After combining and properly rearranging the equations (19), (23) and (26), we obtain the following global system

$$\underbrace{\left(\begin{array}{cc|cccc} K_0^1 & 0 & K_c^1 & 0 & K_\Phi^1 & 0 \\ 0 & K_0^2 & 0 & K_c^2 & 0 & K_\Phi^2 \\ \hline 0 & C_0^2 & -C_c^1 & C_d^2 & 0 & 0 \\ C_0^1 & 0 & C_d^1 & -C_c^2 & 0 & 0 \\ 0 & 0 & D_c^1 & 0 & -D_\Phi^1 & 0 \\ 0 & 0 & 0 & D_c^2 & 0 & -D_\Phi^2 \end{array}\right)}_{=:K_g} \left(\begin{array}{c} \mathbf{c}_0^1 \\ \mathbf{c}_0^2 \\ \hline \mathbf{d}_c^1 \\ \mathbf{d}_c^2 \\ \mathbf{d}_\Phi^1 \\ \mathbf{d}_\Phi^2 \end{array}\right) = \left(\begin{array}{c} \mathbf{f}^1 \\ \mathbf{f}^2 \\ \hline 0 \\ 0 \\ 0 \\ 0 \end{array}\right). \tag{27}$$

The approximation of the exact solution u of Problem 1 restricted to a patch Ω^k is then given by

$$u|_{\Omega^k} \approx u_{0h}^k + u_{Mh}^k = \sum_{i \in \mathcal{I}_0^k \setminus \mathcal{I}_\Phi^k} c_i^k \beta_i^k(\mathbf{x}) + \sum_{i \in \mathcal{I}_\Phi^k} (c_i^k + d_i^k) \beta_i^k(\mathbf{x}) + \sum_{i \in \mathcal{I}_c^k} d_i^k \beta_i^k(\mathbf{x}). \tag{28}$$

Note that, when using the CEO, the coefficient vectors $\mathbf{d}_\Phi^1$, $\mathbf{d}_\Phi^2$ are omitted and the final system is given by the first (top-left) four-by-four blocks only:

$$\left(\begin{array}{cc|cc} K_0^1 & 0 & K_c^1 & 0 \\ 0 & K_0^2 & 0 & K_c^2 \\ \hline 0 & C_0^2 & -C_c^1 & C_d^2 \\ C_0^1 & 0 & C_d^1 & -C_c^2 \end{array}\right) \left(\begin{array}{c} \mathbf{c}_0^1 \\ \mathbf{c}_0^2 \\ \hline \mathbf{d}_c^1 \\ \mathbf{d}_c^2 \end{array}\right) = \left(\begin{array}{c} \mathbf{f}^1 \\ \mathbf{f}^2 \\ \hline 0 \\ 0 \end{array}\right). \tag{29}$$

In this case the approximate solution is given by

$$u|_{\Omega^k} \approx u_{0h}^k + u_{Mh}^k = \sum_{i \in \mathcal{I}_0^k \setminus \mathcal{I}_\Phi^k} c_i^k \beta_i^k(\mathbf{x}) + \sum_{i \in \mathcal{I}_\Phi^k} c_i^k \beta_i^k(\mathbf{x}) + \sum_{i \in \mathcal{I}_c^k} d_i^k \beta_i^k(\mathbf{x}). \tag{30}$$

Theorem 2. *If* $(\mathbf{c}_0^1, \mathbf{c}_0^2, \mathbf{d}_c^1, \mathbf{d}_c^2, \mathbf{d}_\Phi^1, \mathbf{d}_\Phi^2)^T$ *is a solution of* (27)*, then* $(\tilde{\mathbf{c}}_0^1, \tilde{\mathbf{c}}_0^2, \mathbf{d}_c^1, \mathbf{d}_c^2)^T$ *is a solution of* (29)*, where* $\tilde{\mathbf{c}}_0^k = (\tilde{\mathbf{c}}_i^k)_{i \in \mathcal{I}_0^k}$*, with*

$$\tilde{\mathbf{c}}_i^k = \begin{cases} c_i^k & \textit{for } i \in \mathcal{I}_0^k \setminus \mathcal{I}_\Phi^k \\ c_i^k + d_i^k & \textit{for } i \in \mathcal{I}_\Phi^k. \end{cases} \tag{31}$$

Proof. It is easy to see that the first two equations in (29) are satisfied by definition. The last two equations are satisfied, since $C_0^k \tilde{c}_0^k = C_0^k c_0^k$. This is due to the fact that the rows in the matrix C_0^k are zero for all row indices $i \in \mathcal{I}_\Phi^k$ where the vectors $\tilde{\mathbf{c}}_0^k$ and $\mathbf{c}_0^k$ differ, i.e., $(C_0^k)_{\ell i} = \beta_i^k(\mathbf{x}_\ell^{k'}) = 0$, for $\mathbf{x}_\ell^{k'} \in \Gamma_C^{k'}$ and $i \in \mathcal{I}_\Phi^k$. □

It means, if Problem 4 has a unique solution with respect to the QEO, it also has a solution with respect to the CEO. However, the theorem does not guarantee the uniqueness of the solution with respect to the CEO.

4. Existence and uniqueness for the discretized problem

The continuous Problem 2 always has a unique solution, thanks to Theorem 1, where the equivalence of solutions of Problems 1 and 2 was shown. However, Theorem 1 does not apply to Problem 4. Instead we will show by numerical experiments the existence and uniqueness of the solution. For this aim, we need to analyze the regularity of the system matrix K_g. However, we do not analyze the full matrix directly but make a Schur complement reformulation.

We write the final matrix as presented in (27) as a block matrix

$$K_g = \begin{pmatrix} A & B \\ C & -D \end{pmatrix},$$

where

$$A = \begin{pmatrix} K_0^1 & 0 \\ 0 & K_0^2 \end{pmatrix}, \qquad B = \begin{pmatrix} K_c^1 & 0 & K_{\mathcal{E}}^1 & 0 \\ 0 & K_c^2 & 0 & K_{\mathcal{E}}^2 \end{pmatrix},$$
$$C = \begin{pmatrix} 0 & C_0^2 \\ C_0^1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} C_c^1 & -C_d^2 & 0 & 0 \\ -C_d^1 & C_c^2 & 0 & 0 \\ -D_c^1 & 0 & D_\Phi^1 & 0 \\ 0 & -D_c^2 & 0 & D_\Phi^2 \end{pmatrix}.$$

It can be checked easily that the block matrix D is invertible. This is due to the interpolation together with the finite difference scheme having a unique solution. Then the Schur complement matrix with respect to the block D of the matrix K_g is obtained by

$$\tilde{K_g} := A - BD^{-1}C. \tag{32}$$

What remains to be shown is whether or not the matrix $\tilde{K_g}$ is invertible. To study this, we will show numerically that the induced bilinear form $\alpha(u,v) = \mathbf{v}^T\tilde{K_g}\mathbf{u}$ is bounded and coercive, i.e., $\alpha(u,v) \leq \overline{\mu}\|u\|_{H^1}\|v\|_{H^1}$ and $\alpha(u,u) \geq \underline{\mu}\|u\|_{H^1}^2$. Here, u, v are the trial/test functions corresponding to the vectors $\mathbf{u}$, $\mathbf{v}$.

Let $K_g^s = \frac{1}{2}(\tilde{K_g} + \tilde{K_g}^T)$ and N be the matrix related to the H^1-norm, i.e., $\mathbf{u}^T N\mathbf{u} = \|u\|_{H^1}^2$. Computing the minimal eigenvalue of the matrix $K_g^s N^{-1}$

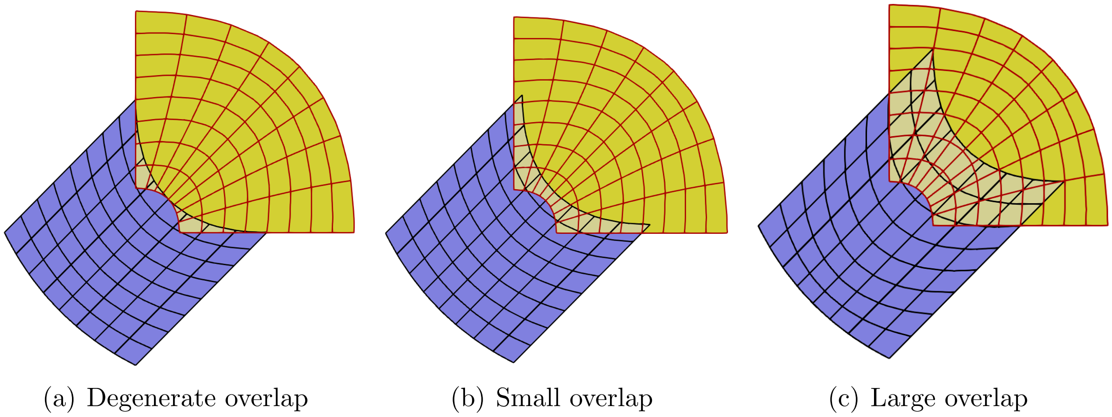
$$\lambda_{\min} = \inf_{\mathbf{u}} \frac{\mathbf{u}^T K_g^s \mathbf{u}}{\mathbf{u}^T N \mathbf{u}} \tag{33}$$

we get that

$$\lambda_{\min}\|u\|_{H^1}^2 = \lambda_{\min}\mathbf{u}^T N\mathbf{u} \leq \mathbf{u}^T K_g^s\mathbf{u} = \mathbf{u}^T\tilde{K_g}\mathbf{u} = \alpha(u,u). \tag{34}$$

Hence, having a positive minimal eigenvalue $\lambda_{\min}$ guarantees that the bilinear form α is coercive, therefore the matrix $\tilde{K_g}$ is invertible and so is the original system matrix K_g.

In the following, we compute $\lambda_{\min}$ for some test examples, where we observe that it is quite robust with respect to the degree p and the mesh size h for a moderately sized overlap. For small overlaps, however, the minimal eigenvalue may tend to zero, and solvability of the matrix K_g is not guaranteed. It is worth to mention here, that, even though we do not have a rigorous proof, we did not encounter cases where the linear system was not solvable.


(a) Degenerate overlap

(b) Small overlap

(c) Large overlap

Figure 4: Two-patch setting for different size of overlaps.

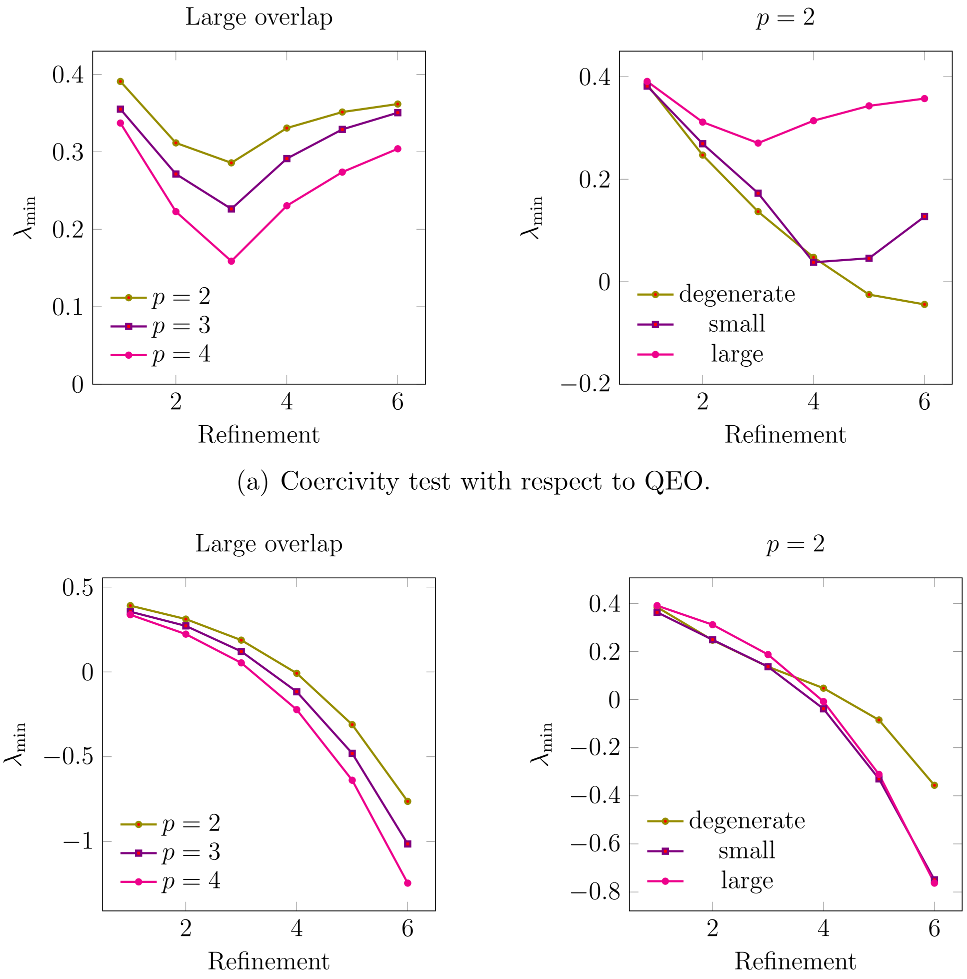
We consider the domains illustrated in Figure 4. While the domain itself cannot be parameterized by a single regular spline patch, it can be represented as the union of two patches with overlap.

In Figures 5(a) (left) we plot the minimum eigenvalues, obtained by equation (33) with respect to QEO, for different degrees of basis functions. In the first three refinement steps, the support of the basis functions are bigger. Therefore we do not have enough basis functions in the overlap regions, consequently the minimal eigenvalues increase, up to the forth refinement step.

When increasing the degree of the basis functions, the value of the minimal eigenvalue decreases. We suspect that this is due to the increase of the support size of the basis functions. However, in all the cases we considered, the eigenvalues are still positive and increase again after several refinement steps.

In Figures 5(a) (right) we compare the results for different size of the overlaps. For degenerate overlap, we get negative eigenvalues after some refinement steps. For moderately sized overlap regions the minimal eigenvalue increases as the mesh size goes to zero. As we can see in Figure 5(a) (right), larger overlaps always guarantee the positive eigenvalues and so the solvability of the system. Summing up, we need enough basis functions in the overlap to obtain coercivity this can be guaranteed by refinement.

As we can see in Figure 5(b), in case of using CEO, the minimal eigenvalues obtained by equation



(a) Coercivity test with respect to QEO.

(b) Coercivity test with respect to CEO.

Figure 5: Coercivity test with different degree of basis functions (left) and different size of overlaps (right).

(33) are negative after some refinement steps either for different degrees of the basis functions or different size of overlaps. However, since Problem 4 has a unique solution with respect to QEO, according to Theorem 2, it also has a solution with respect to CEO. While the theorem does not guarantee uniqueness, we always found that the CEO based formulation is uniquely solvable.

5. Numerical experiments

We study the performance of the OMP method for some numerical examples. Since, according to Theorem 2, any solution using the QEO directly yields a solution using the CEO, we only use CEO for our examples. We compare the computational time of the OMP method with respect to CEO and QEO in Section 5.1. In Section 5.2 we solve the Poisson problem with non-homogeneous boundary condition and given right-hand side on different domains, which are a union of two or more subdomains. In all cases, we observe an optimal convergence rate. In Section 5.3 we compare the OMP method with two versions of the Schwarz methods. We implemented this work using the C++ library **G+Smo** [39].

5.1. Time comparison between CEO and QEO

We compare the performance of the OMP method for CEO and QEO with respect to the computational time. For QEO the support of the additional basis functions need to be entirely in the overlap region. Hence, to obtain a difference between CEO and QEO, a domain with sufficiently large overlap is required. Otherwise, the final systems related to CEO and QEO are of similar size and the computational time regarding the extension operators is not considerably different. We solve the Poisson problem on the domain depicted in Figure 4(c) with respect to CEO and QEO having the exact solution (35). Table 1 and Figure 6 indicate results for degree two and different levels of refinement. Moreover, in Table 2 we compare the two extension operators for varying degree on a fixed refinement level. The observed results show that the OMP method with respect to CEO is almost two times faster than QEO.

They also demonstrate experimentally that the computational time of the both methods have the same order of magnitude. Consequently, the observations regarding CEO, which are represented in the following subsections are expected to carry over to the QEO as well.

5.2. Convergence tests

In Example 5.2.1 we consider a half-circle shaped domain, which is the union of two overlapping patches. The two subdomains are sharing a Dirichlet boundary part. A coffee cup shaped domain is considered in Example 5.2.2. The domain is a union of four pairwise overlapping subdomains. In Example 5.2.3 the performance of the OMP method is tested on a $3D$ domain, which is constructed as the union of two overlapping cubes.

In all three examples we solve the Poisson problem, where the right-hand side f and the Dirichlet boundary conditions are defined by a prescribed exact solution. The numerical solution is computed by solving the global system (29) for varying degree and mesh size. We always compute the L^2 and H^1 errors. In all examples the total error is given as the sum of local errors on each subdomain.

	CEO			QEO			time ratio
ref	time	#it	size	time	#it	size	QEO/CEO
1	9.9×10^{-4}	31	648	2.2×10^{-3}	37	746	2.22
2	7.5×10^{-3}	63	2312	1.3×10^{-2}	74	2834	1.73
3	6×10^{-2}	128	8712	1×10^{-1}	162	11136	1.66
4	5.5×10^{-1}	234	33800	1	316	44186	1.81
5	4.9	499	133128	8.8	649	176006	1.79
6	44	949	528392	86	1340	702720	1.95

Table 1: Time comparison with respect to refinement levels for CEO and QEO on domain 4(c) and p = 2. Here and in the following tables #it stands for solver iteration.

	CEO			QEO			time ratio
degree	time	#it	size	time	#it	size	QEO/CEO
2	44	949	528392	86	1340	702720	1.95
3	68.8	831	530450	131	1159	704734	1.90
4	104	756	532512	201	1220	706756	1.93

Table 2: Time comparison with respect to degree for CEO and QEO on domain 4(c)

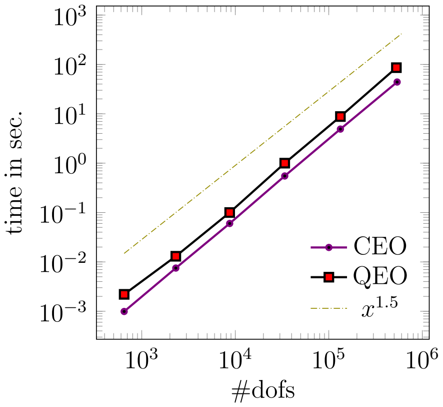


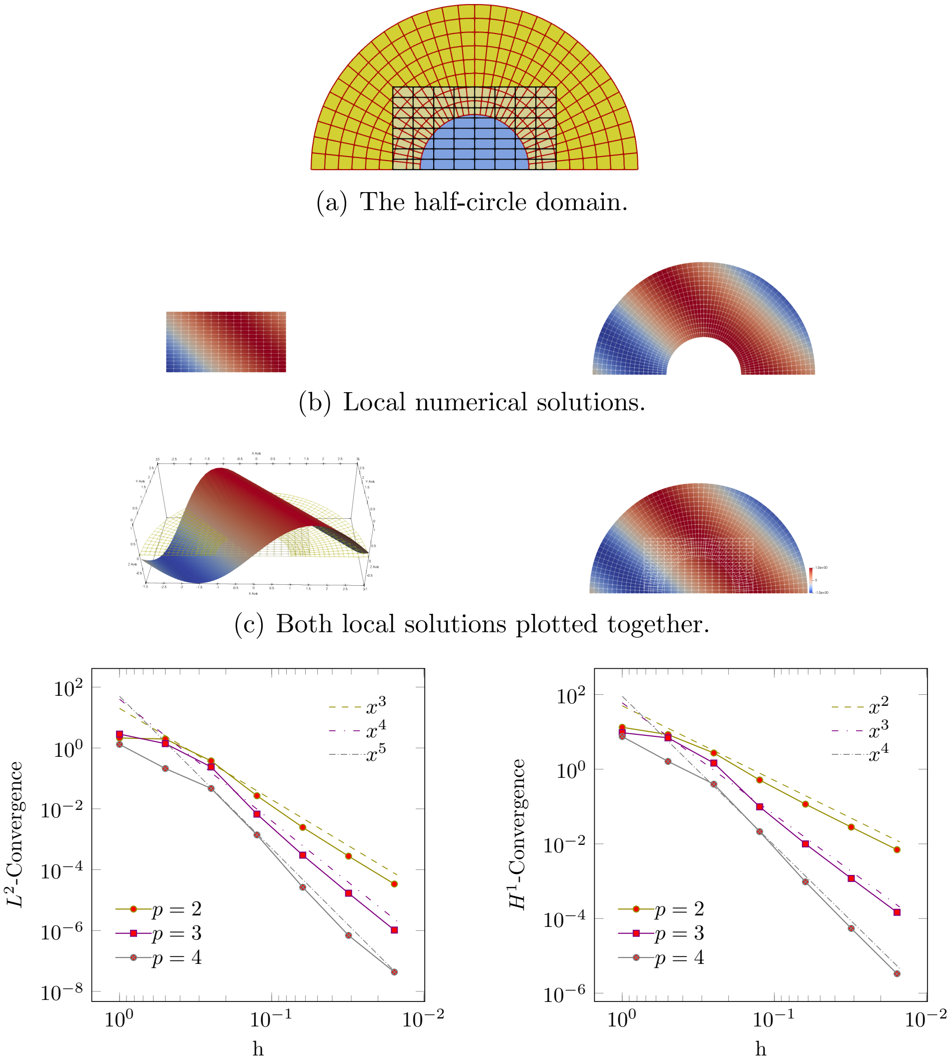
Figure 6: Time comparison with respect to refinement levels for CEO and QEO on domain 4(c).

5.2.1. Poisson problem on a half-circle

The domain illustrated in Figure 7(a), approximates a half-circle with radius 3 around the origin (the domain is parameterized with quadratic B-splines and thus *not* an exact half-circle). While the domain itself cannot be parameterized by a single regular spline patch, it can be represented as the union of an annulus and a rectangular domain. We consider the exact solution

$$u = \sin(x + y). \tag{35}$$

The numerical solution with a total of 1512 dofs and using quadratic B-splines, is plotted in Figures 7(b) and 7(c). The L^2 and H^1 errors are plotted in Figure 7(d). The observed order of convergence is optimal.



(a) The half-circle domain.

(b) Local numerical solutions.

(c) Both local solutions plotted together.

(d) L^2 (left) and H^1 (right) errors for the half-circle.

Figure 7: Numerical results related to solving the Poisson problem on the half-circle.

5.2.2. Poisson problem on a coffee cup shaped domain

In this example, we consider the domain depicted in Figure 8(a). The domain is composed of four subdomains parameterized with quadratic B-splines. We consider the following exact solution

$$u = \sin(x)\sin(y).$$

In Figure 8(c) we plot the numerical solution with a total of 1296 dofs for $p = 2$. The convergence of L^2 and H^1 errors are shown in Figure 8(d), which verifies the optimal convergence rate for this example.

5.2.3. Poisson problem on overlapping cubes

We consider the $3D$ domain illustrated in Figure 9(a), which is given as the union of two cubes. The exact solution is

$$u = \sin(x + y + z).$$

The numerical solution for $p = 2$ with a total of 2000 dofs is plotted in Figure 9(b).

We again observe the optimal order of convergence for the L^2 and H^1 errors in Figure 9(c).

5.2.4. Linear elasticity problem on a plate with circular hole

The domain, which is shown in Figure 10(a), is a quarter of a plate with a circular hole. It is represented as a Boolean union of two subdomains, both parameterized by quadratic NURBS. The radius of the hole and the edge length of the quarter are set to one and four, respectively.

On this domain, we consider the same problem as in [40, Section 5.2], where the domain was simply a square. We recall the exact solution

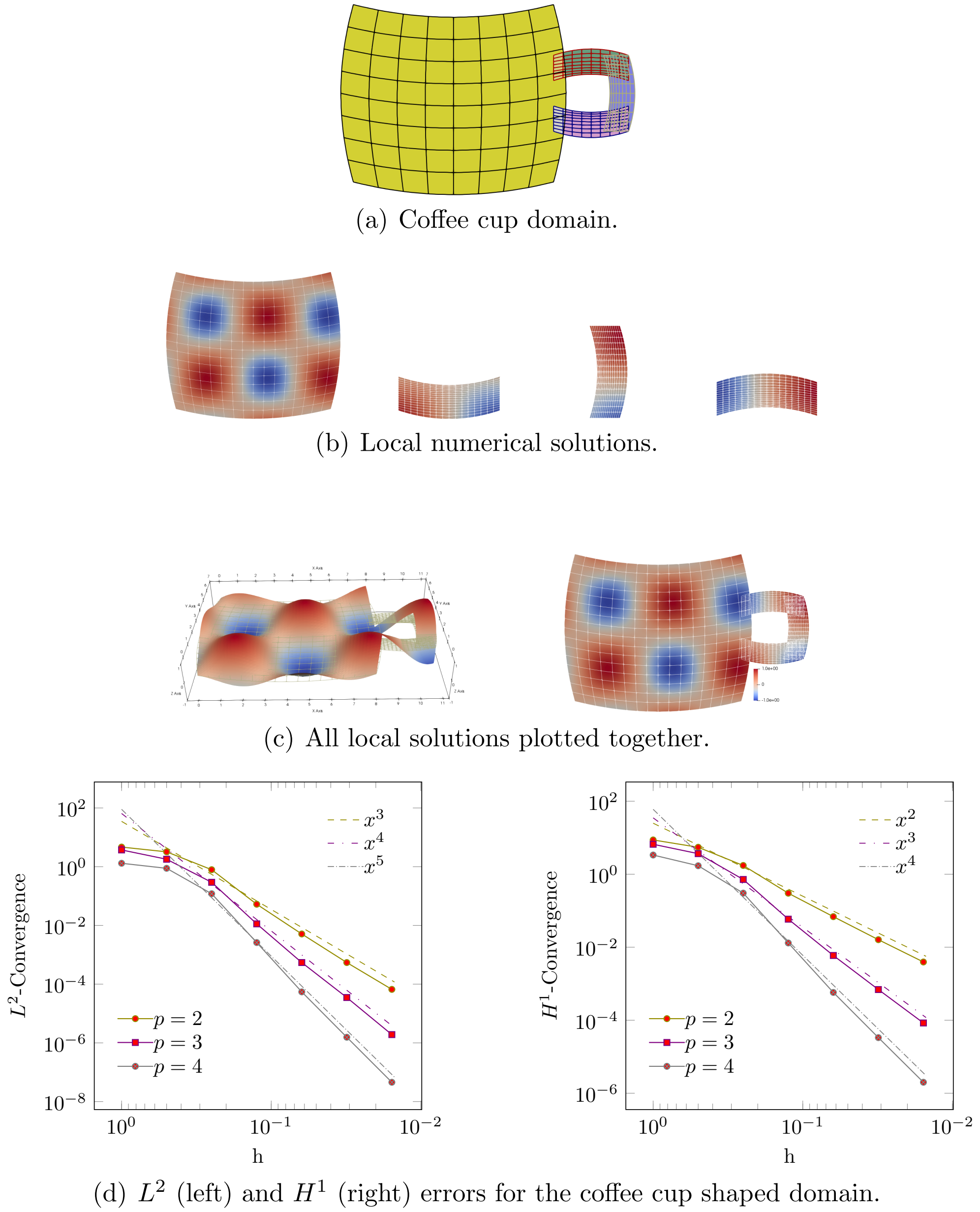
$$\begin{aligned} u_1 &= -\frac{\cos^2(x)\cos(y)\sin(y)}{2} \\ u_2 &= \frac{\cos^2(y)\cos(x)\sin(x)}{2}, \end{aligned}$$

and right-hand side

$$\begin{aligned} f_1 &= \mu\cos(y)\sin(y)(1 - 4\cos^2(x)) - 2xy\cos(x^2y) \\ f_2 &= -\mu\cos(x)\sin(x)(1 - 4\cos^2(y)) - x^2\cos(x^2y). \end{aligned}$$

The Lamé parameters are determined by Young's modulus $E = 10^5$ and Poisson's ratio $\nu = 0.3$.

Figures 10(b) and 10(c) visualize the numerical solution with a total of 2664 dofs (corresponding to $h = 0.03125$) and using quadratic NURBS. The L^2 and H^1 errors for various values of the discretization parameter h are plotted in Figure 10(d). The observed order of convergence is optimal.



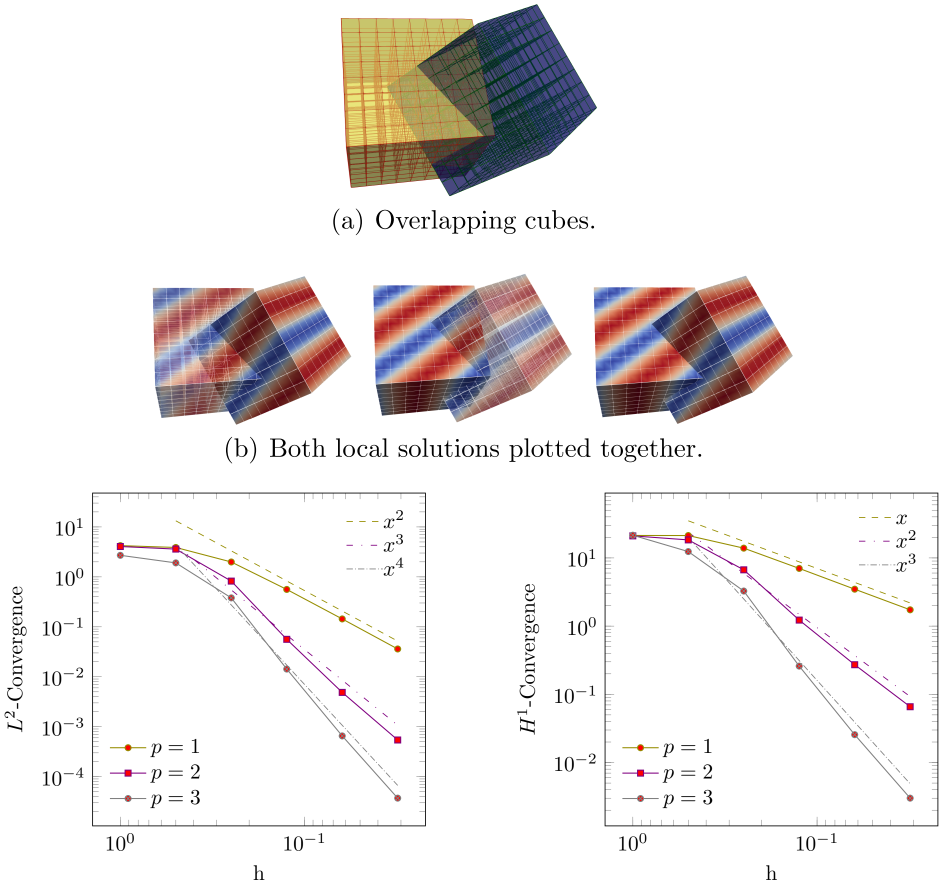
(a) Coffee cup domain.

(b) Local numerical solutions.

(c) All local solutions plotted together.

(d) L^2 (left) and H^1 (right) errors for the coffee cup shaped domain.

Figure 8: Numerical results related to solving the Poisson problem on the coffee cup shaped domain.



(a) Overlapping cubes.

(b) Both local solutions plotted together.

(c) L^2 (left) and H^1 (right) errors for the overlapping cubes.

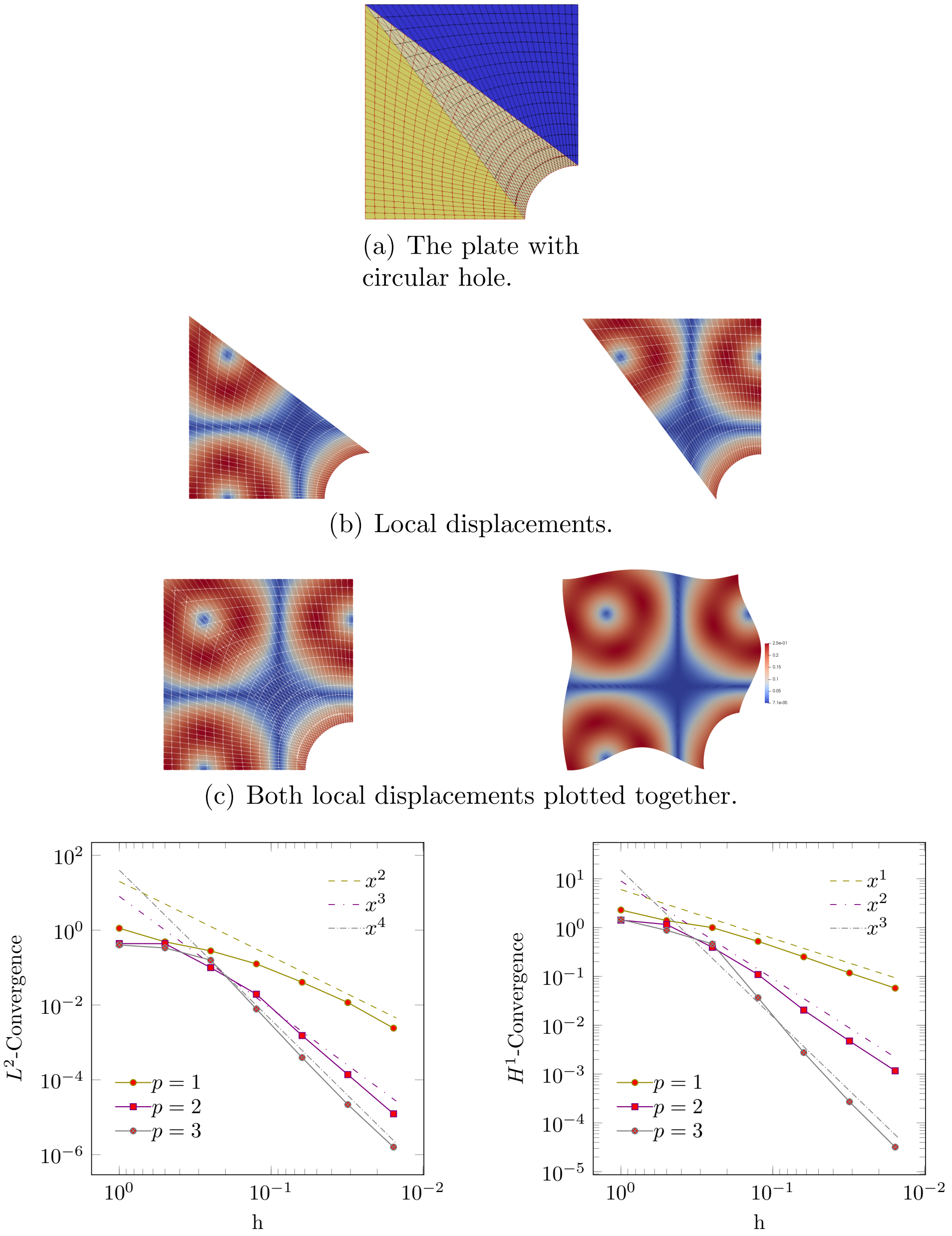
Figure 9: Numerical results related to solving the Poisson problem on the overlapping cubes.

5.3. Comparison of the OMP method with Schwarz methods

We compare the OMP method with two different kinds of Schwarz algorithms, additive Schwarz (ADS) and multiplicative Schwarz (MPS).

The ADS and MPS domain decomposition methods, can be used to handle the numerical solution on overlapping subdomains. Moreover, from a linear algebra point of view, the ADS method is a variation of the block Jacobi algorithm and the MPS method corresponds to a symmetric Gauss-Seidel algorithm. In Figure 11 we visualize the dependence of every iterate with respect to the previous solutions. For more information see for instance [41, 42].

In Section 5.3.1 we compare the presented Schwarz algorithms and the proposed OMP method with respect to the computational cost.



(a) The plate with circular hole.

(b) Local displacements.

(c) Both local displacements plotted together.

(d) L^2 (left) and H^1 (right) errors for the plate with circular hole.

Figure 10: Numerical results related to solving the linear elasticity problem on the plate with circular hole.

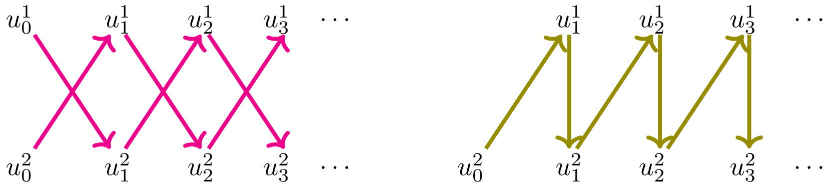


Figure 11: ADS (pink) and MPS (olive) algorithms

In all examples, the Schwarz iteration is terminated if the Euclidean norm of the difference of two successive solutions is below the threshold 10^{-8}. The maximum number of iterations for the MPS and ADS method is set to 4000.

5.3.1. Comparison with respect to the computational cost

We solve the Poisson problem with non-zero Dirichlet boundary condition on a domain, which is composed of two subdomains and with different size of overlaps, as illustrated in Figure 4. Note that the domain is the same for the three examples and only one subdomain changes. The exact solution

$$u = \sin(\pi x) \sin(\pi y)$$

is used for all three approaches.

The BiCG solver is employed for solving the systems related to the proposed methods. The tolerance of the solver is set to 10^{-10}.

The experimental results illustrated in the Tables 3, 4, 5 and Figure 12 for $p = 2$. The observed results show that the MPS method is almost two times faster than the ADS method and the OMP method is considerably faster than both Schwarz approaches. Also, the Schwarz approaches are not efficient for degenerate and small overlaps, since we lose convergence after some refinement steps. Also, setting a higher number of maximum iterations of the Schwarz method (> 4000), increases the computation time.

Note that in the presented tables, the dofs are not divided by two for the OMP method.

As another experiment, we fix the number of dofs, and we elevate the degree of the basis functions. The results are indicated in Tables 6, 7 and 8 for different size of overlaps. The results show again that the OMP method is significantly faster than the ADS and MPS methods. Since the ADS and MPS methods fail to converge after some refinement steps, due to the degenerate overlap and $p = 4$, it is not possible to compute the time consumption in this case. The ADS method seems to behave worse than the MPS method with that respect. However, the performance of the OMP method, even for degenerate overlaps, is not compromised as we obtain the optimal convergence rate in all cases, and the computational cost is computed.

The MPS and ADS systems are symmetric. One can solve those systems using a CG solver.

Overlap size = Degenerate									
$p = 2$		OMP Method		MPS Method			ADS Method		
ref	#dofs/2	time	#it	calls	time	#it	calls	time	#it
1	9	8.3×10^{-6}	5	2	1.7×10^{-5}	18	2	1.3×10^{-5}	18
2	16	2×10^{-6}	9	48	2.9×10^{-4}	490	113	6.7×10^{-4}	1140
3	36	5.5×10^{-6}	15	83	1.6×10^{-3}	1428	160	3.2×10^{-3}	2738
4	100	2.2×10^{-4}	26	139	1.3×10^{-2}	3501	268	2.5×10^{-2}	6726
5	324	1.4×10^{-3}	48	291	1.5×10^{-1}	11103	560	2.9×10^{-1}	21330
6	1156	9.2×10^{-3}	83	660	2.8	51435	1268	5.4	98742
7	4356	8.2×10^{-2}	159	1490	50.2	229183	2799	93.8	430450
8	16900	7.1×10^{-1}	309	-	>10min	-	-	>10min	-

Table 3: Comparison between the OMP, MPS, and ADS methods.

Overlap size = Small									
$p = 2$		OMP Method		MPS Method			ADS Method		
ref	#dofs/2	time	#it	calls	time	#it	calls	time	#it
1	9	1.2×10^{-5}	13	57	1.9×10^{-4}	348	110	3.3×10^{-4}	666
2	16	1.3×10^{-5}	11	30	1.8×10^{-4}	310	62	4.04×10^{-4}	630
3	36	5.6×10^{-5}	15	32	6.7×10^{-4}	593	62	1.3×10^{-3}	1132
4	100	1.9×10^{-4}	21	33	003.3×10^{-3}	850	64	6.1×10^{-3}	1625
5	324	1.1×10^{-3}	35	34	1.9×10^{-2}	1341	66	3.5×10^{-2}	2569
6	1156	7.7×10^{-3}	65	35	1.5×10^{-1}	2771	68	2.9×10^{-1}	5314
7	4356	6.1×10^{-2}	128	52	1.7	8090	121	4.07	18630
8	16900	6.1×10^{-1}	243	-	> 10min	-	-	> 10min	-

Table 4: Comparison between the OMP, MPS, and ADS methods.

Overlap size = Large									
$p = 2$		OMP Method		MPS Method			ADS Method		
ref	#dofs/2	time	#it	calls	time	#it	calls	time	#it
1	9	1.1×10^{-5}	12	12	4.19×10^{-5}	77	24	8.67×10^{-5}	148
2	16	2.6×10^{-5}	11	8	5.36×10^{-5}	88	17	1.03×10^{-4}	177
3	36	3.3×10^{-5}	12	9	2.03×10^{-4}	170	19	3.98×10^{-4}	339
4	100	1.6×10^{-4}	18	9	10.07×10^{-4}	260	19	1.98×10^{-3}	518
5	324	9.9×10^{-4}	30	10	6.15×10^{-3}	450	19	1.11×10^{-2}	819
6	1156	6.4×10^{-3}	61	10	4.90×10^{-2}	879	19	9.01×10^{-2}	1606
7	4356	5.7×10^{-2}	121	26	9.95×10^{-1}	4485	51	1.92	8578
8	16900	0.6	234	3000	864.244	983107	-	> 15min	-

Table 5: Comparison between the OMP, MPS, and ADS methods.

This would be two times faster. However, according to the observed results, the OMP method is still significantly faster.

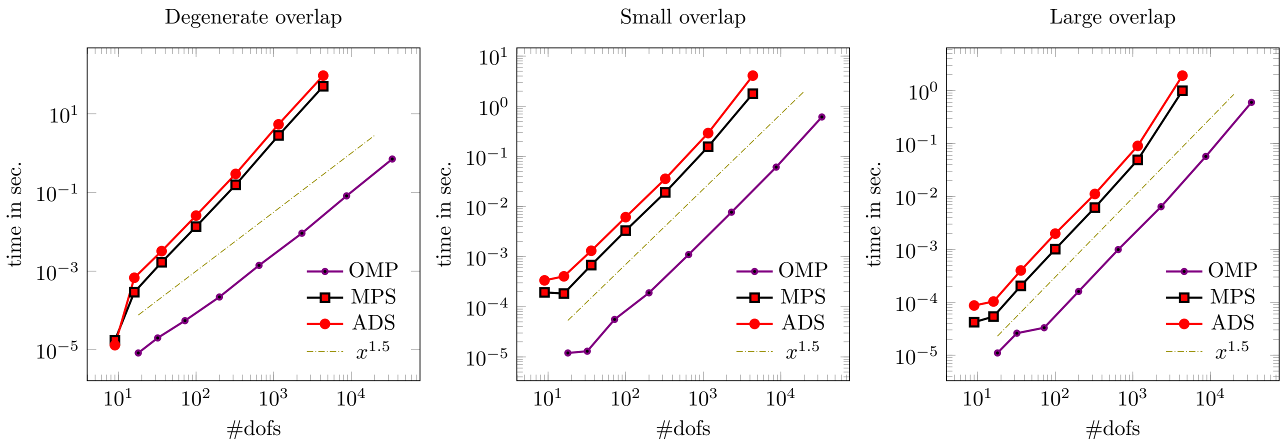


Figure 12: Comparison between the OMP, MPS and ADS methods with respect to the computational time.

refinement level = 7									
Overlap size = Degenerate									
		OMP Method		MPS Method			ADS Method		
degree	#dofs/2	time	#it	calls	time	#it	calls	time	#it
2	4356	8.2×10^{-2}	159	1490	50.2	229183	2799	93.8	430450
3	4489	1.2×10^{-1}	131	395	22.4188	52090	898	50.6409	118263
4	4624	9.5×10^{-1}	482	-	-	-	-	-	-

Table 6: Comparison between the OMP, MPS and ADS methods for different degrees of the basis functions.

refinement level = 7									
Overlap size = Small									
		OMP Method		MPS Method			ADS Method		
degree	#dofs/2	time	#it	calls	time	#it	calls	time	#it
2	4356	6.1×10^{-2}	128	52	1.7	8090	121	4.07	18630
3	4489	1.1×10^{-1}	109	58	3.45672	7836	111	6.41868	14882
4	4624	1.8×10^{-1}	104	38	3.25108	4766	74	6.28625	9157

Table 7: Comparison between the OMP, MPS and ADS methods for different degrees of the basis functions.

6. Conclusion

We proposed the OMP method, which is an isogeometric method for numerical simulation on overlapping multi-patch structures, where the individual patches are represented by tensor-product

refinement level = 7									
Overlap size = large									
		OMP Method		MPS Method			ADS Method		
degree	#dofs/2	time	#it	calls	time	#it	calls	time	#it
2	4356	5.7×10^{-2}	121	26	9.95×10^{-1}	4485	51	1.92	8578
3	4489	9.5×10^{-2}	104	25	1.62	3684	44	2.80	6397
4	4624	0.15	94	16	1.53	2194	22	2.07	2963

Table 8: Comparison between the OMP, MPS and ADS methods for different degrees of the basis functions.

spline parameterizations. We observe that neither trimming nor reparameterization are required for domains constructed by Boolean unions. Also, the OMP method avoids the iterative approach of the ADS or MPS methods.

The coupling of the individual solutions on the subdomains relies on two different extension operators, CEO and QEO. We explored that their convergence rates behave analogously. Therefore we used the CEO for all the numerical examples. It should be noted that we cannot guarantee the uniqueness of the solution with respect to the CEO (not even numerically), while numerical experiments indicate that coercivity of the discrete formulation is valid for QEO.

We compared the OMP method with the two Schwarz methods, with respect to computational time. While ADS and MPS are not efficient for small overlaps, the OMP method provides the optimal rate of convergence, even in these situations. Moreover, we can employ OMP to more than two patches with pairwise overlaps, and in any dimension.

In our ongoing work, we plan to extend the OMP method to multi-patch domains containing overlaps of three or more patches.

Furthermore, we are interested in studying higher order PDEs, where a higher order of smoothness needs to be imposed on the boundary of the overlaps. Another possible extension is the case of PDEs on surfaces. We believe that the use of overlapping multi-patch structures might be useful for applications in geometric modeling also. As an additional difficulty, the individual patches may not coincide exactly in the overlap, but only approximately.

The OMP method consists of a single monolithic linear system that needs to be solved. This is different from the additive Schwarz methods, which are inherently iterative and parallelizable. Nevertheless, existing and well developed packages for parallelizing large sparse linear systems (e.g. Trilinos or Hypre) can be employed to obtain a parallel version of our method. The structure of the global matrix can be exploited by splitting into sub-problems. Devising an adapted parallel version of our method is also a potential topic for future research.

7. Acknowledgments

This work was supported by the ERC AdG project CHANGE n. 694515, by the Austrian Science Fund (FWF) through project NFN S11708 as well as by the FWF together with the government of Upper Austria through the project P 30926-NBL. This support is gratefully acknowledged.

References

[1] T. J. R. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer Methods in Applied Mechanics and Engineering 194 (39-41) (2005) 4135–4195.

[2] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley and Sons, Ltd, 2009.

[3] A. Buffa, G. Sangalli (Eds.), IsoGeometric Analysis: A New Paradigm in the Numerical Approximation of PDEs, Lecture Notes in Mathematics, Springer International Publishing Switzerland, 2016.

[4] L. Beirão Da Veiga, A. Buffa, G. Sangalli, R. Vázquez, Mathematical analysis of variational isogeometric methods, Acta Numerica 23 (2014) 157–287.

[5] M. Scott, D. Thomas, E. Evans, Isogeometric spline forests, Computer Methods in Applied Mechanics and Engineering 269 (2014) 222–264.

[6] F. Buchegger, B. Jüttler, Planar multi-patch domain parameterization via patch adjacency graphs, Computer-Aided Design 82 (2017) 2–12.

[7] S. K. Kleiss, C. Pechstein, B. Jüttler, S. Tomar, IETI–isogeometric tearing and interconnecting, Computer methods in applied mechanics and engineering 247 (2012) 201–215.

[8] F. Buchegger, B. Jüttler, Adaptively refined multi-patch B-splines with enhanced smoothness, Applied Mathematics and Computation 272 (2016) 159–172.

[9] A. Collin, G. Sangalli, T. Takacs, Analysis-suitable G^1 multi-patch parametrizations for C^1 isogeometric spaces, Computer Aided Geometric Design 47 (2016) 93–113.

[10] M. Kapl, G. Sangalli, T. Takacs, Construction of analysis-suitable G^1 planar multi-patch parameterizations, Computer-Aided Design 97 (2018) 41–55.

[11] E. Brivadis, A. Buffa, B. Wohlmuth, L. Wunderlich, Isogeometric mortar methods, Computer Methods in Applied Mechanics and Engineering 284 (2015) 292–319.

[12] S. Deparis, D. Forti, P. Gervasio, A. Quarteroni, INTERNODES: an accurate interpolation-based method for coupling the galerkin solutions of PDEs on subdomains featuring non-conforming interfaces, Computers and Fluids 141 (2016) 22–41.

[13] P. Gervasioa, A. Quarteroni, Analysis of the INTERNODES method for non-conforming discretizations of elliptic equations, Computer Methods in Applied Mechanics and Engineering 334 (2018) 138–166.

[14] A. Apostolatos, R. Schmidt, R. Wüchner, K.-U. Bletzinger, A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis, International Journal for Numerical Methods in Engineering 97 (7) (2014) 473–504.

[15] V. P. Nguyen, P. Kerfriden, M. Brino, S. P. Bordas, E. Bonisoli, Nitsches method for two and three dimensional NURBS patch coupling, Computational Mechanics 53 (6) (2014) 1163–1182.

[16] Y. Guo, M. Ruess, Nitsches method for a coupling of isogeometric thin shells and blended shell structures, Computer Methods in Applied Mechanics and Engineering 284 (2015) 881–905.

[17] M. Ruess, D. Schillinger, A. I. Özcan, E. Rank, Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries, Computer Methods in Applied Mechanics and Engineering 269 (2014) 46–71.

[18] R. Bouclier, J.-C. Passieux, A Nitsche-based non-intrusive coupling strategy for global/local isogeometric structural analysis, Computer Methods in Applied Mechanics and Engineering.

[19] F. Brunero, L. F. Pavarino, C. Pechstein, Discontinuous Galerkin methods for isogeometric analysis, Universit degli Studi di Milano.

[20] U. Langer, A. Mantzaflaris, S. E. Moore, I. Toulopoulos, Multipatch discontinuous Galerkin isogeometric analysis, in: Isogeometric Analysis and Applications 2014, Springer, 2015, pp. 1–32.

[21] C. Hofer, U. Langer, Dual-primal isogeometric tearing and interconnecting solvers for multipatch dG-IgA equations, Computer Methods in Applied Mechanics and Engineering 316 (2017) 2–21.

[22] F. Massarwi, G. Elber, A B-spline based framework for volumetric object modeling, Computer Aided Design 78 (2016) 36–47.

[23] H.-J. Kim, Y.-D. Seo, S.-K. Youn, Isogeometric analysis for trimmed CAD surfaces, Computer Methods in Applied Mechanics and Engineering 198 (37-40) (2009) 2982–2995.

[24] R. Schmidt, R. Wüchner, K.-U. Bletzinger, Isogeometric analysis of trimmed NURBS geometries, Computer Methods in Applied Mechanics and Engineering 241 (2012) 93–111.

[25] F. Massarwi, B. van Sosin, G. Elber, Untrimming: Precise conversion of trimmed-surfaces to tensor-product surfaces, Computers & Graphics 70 (2018) 80–91.

[26] M. Randrianarivony, Geometric processing of CAD data and meshes as input of integral equation solvers, Ph.D. thesis, Computer Science Faculty Technische Universität Chemnitz, 2006.

[27] S. Xia, X. Wang, X. Qian, Continuity and convergence in rational triangular Bézier spline based isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 297 (2015) 292–324.

[28] B. Marussig, T. J. R. Hughes, A review of trimming in isogeometric analysis: Challenges, data exchange and simulation aspects, Archives of Computational Methods in Engineering 25 (2018) 1059–1127.

[29] K. Höllig, Finite Element Methods with B-Splines, Vol. 26, SIAM, 2003.

[30] R. Sanches, P. Bornemann, F.Cirak, Immersed B-spline (i-spline) finite element method for geometrically complex domains, Computer Methods in Applied Mechanics and Engineering 200 (2011) 1432–1445.

[31] B. Marussig, R. Hiemstra, T. J. R. Hughes, Improved conditioning of isogeometric analysis matrices for trimmed geometries, Computer Methods in Applied Mechanics and Engineering 334 (2018) 79–110.

[32] L. Kudela, N. Zander, S. Kollmannsberger, E. Rank, Smart octrees: Accurately integrating discontinuous functions in 3D, Computer Methods in Applied Mechanics and Engineering 306 (2016) 406–426.

[33] F. Scholz, A. Mantzflaris, B. Jüttler, First order error correction for trimmed quadrature in isogeometric analysis, RICAM-Report.

[34] H. Zhang, R. Mo, N. Wan, An iga discontinuous galerkin method on the union of overlapped patches, Computer Methods in Applied Mechanics and Engineering 326 (2017) 445–418.

[35] X. Cai, Overlapping Domain Decomposition Methods, Vol. 33, Springer and Berlin and Heidelberg, 2003.

[36] M. Bercovier, I. Soloveichik, Overlapping non matching meshes domain decomposition method in isogeometric analysis, arXiv 1502.03756v1.

[37] H. Prautzsch, W. Boehm, M. Paluszny, Bézier and B-Spline Techniques, Springer, 2002.

[38] J. Monterde, H. Ugail, On harmonic and biharmonic Bézier surfaces, Computer Aided Geometric Design 21 (2004) 697–715.

[39] A. Mantzaflaris, S. Kargaran, others (see website), G+smo (geometry plus simulation modules) v0.8.1, http://github.com/gismo (2018).

[40] F. Auricchio, L. Beirão da Veiga, A. Buffa, C. Lovadina, A. Reali, G. Sangalli, A fully "locking-free" isogeometric approach for plane linear elasticity problems: A stream function formulation, Computer Methods in Applied Mechanics and Engineering 197 (2007) 160–172.

[41] L. Beirão Da Veiga, D. Cho, L. F. Pavarino, S. Scacchi, Overlapping Schwarz methods for isogeometric analysis, SIAM Journal on Numerical Analysis 50 (2012) 1394–1416.

[42] V. Dolean, P. Jolivet, F. Nataf, An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation, SIAM, 2015.

Appendix

Proof of Theorem 1

First we prove (i). The assumption (3) about the extension operators implies that the two functions $u_0^k = u|_{\Omega^k} - M^k(u|_{\Omega^{k'}})$, $k = 1, 2$, belong to $H_0^1(\Omega^k)$. In order to verify that (u_0^1, u_0^2) satisfies the localized weak form of the Poisson problem, we introduce the trivial extension operators $T^k : H_0^1(\Omega^k) \to H_0^1(\Omega)$,

$$T^k v^k = \begin{cases} v^k \text{ on } \Omega^k \\ 0 \text{ on } \Omega \setminus \Omega^k \end{cases}$$

and note that $a^k(u, v^k) = a(u, T^k v^k)$ and $\ell^k(v^k) = \ell(T^k v^k)$. Therefore we evaluate the left-hand side of Eq. (6) for $u_0^k = u|_{\Omega^k} - M^k u|_{\Omega^{k'}}$:

$$\begin{gathered} a^1\left(M^{12}\left(u|_{\Omega^1} - M^1 u|_{\Omega^2} + M^1(u|_{\Omega^2} - M^2 u|_{\Omega^1})\right), v^1\right) + \\ a^2\left(M^{21}\left(u|_{\Omega^2} - M^2 u|_{\Omega^1} + M^2(u|_{\Omega^1} - M^1 u|_{\Omega^2})\right), v^2\right) = \\ a^1\left(M^{12}(u|_{\Omega^1} - M^1 M^2(u|_{\Omega^1})), v^1\right) + a^2\left(M^{21}(u|_{\Omega^2} - M^2 M^1(u|_{\Omega^2})), v^2\right) = \\ a^1\left(M^{12}(I^1 - M^1 M^2)(u|_{\Omega^1}), v^1\right) + a^2\left(M^{21}(I^2 - M^2 M^1)(u|_{\Omega^2}), v^2\right) = \\ a^1(u|_{\Omega^1}, v^1) + a^2(u|_{\Omega^2}, v^2), \end{gathered}$$

because $M^{kk'}\left(I^k - M^k M^{k'}\right)^{-1} = I$. We use the trivial extension operators and the assumption that u solves the weak form (2) of the problem to rewrite the result as

$$a^1(u|_{\Omega^1}, v^1) + a^2(u|_{\Omega^2}, v^2) = a(u|_{\Omega^1}, T^1 v^1) + a(u|_{\Omega^2}, T^2 v^2) = \ell(T^1 v^1) + \ell(T^2 v^2) = \ell^1(v^1) + \ell^2(v^2),$$

thereby completing the proof of (i).

Second we prove (ii). We consider the restrictions of the two forms

$$a^{12}(u^{12}, v^{12}) = \int_{\Phi^{12}} \nabla u^{12} \nabla v^{12}\, \mathrm{d}\xi \quad \text{and} \quad \ell^{12}(v^{12}) = \int_{\Phi^{12}} f v^{12}\, \mathrm{d}\xi,$$

to the overlap region Φ^{12} and the trivial extension operator $T^{12} : H_0^1(\Phi^{12}) \to H_0^1(\Omega)$,

$$T^{12} v^{12} = \begin{cases} v^{12} \text{ on } \Phi^{12} \\ 0 \text{ on } \Omega \setminus \Phi^{12}. \end{cases}$$

We first show that $M^{12}\left(u_0^1 + M^1 u_0^2\right)$ and $M^{21}\left(u_0^2 + M^2 u_0^1\right)$ coincide on the boundary of Φ^{12}. We know, $M^{kk'} = \sum_{\ell=0}^{\infty}(M^k M^{k'})^\ell$, then we can write

$$\begin{array}{l} M^{12}\left(u_0^1 + M^1 u_0^2\right) = \sum_{\ell=0}^{\infty}(M^1 M^2)^\ell \left(u_0^1 + M^1 u_0^2\right) \\ M^{21}\left(u_0^2 + M^2 u_0^1\right) = \sum_{\ell=0}^{\infty}(M^2 M^1)^\ell \left(u_0^2 + M^2 u_0^1\right). \end{array} \tag{36}$$

According to the assumption (3) we get on Γ_C^1

$$\sum_{\ell=0}^{\infty}(M^1 M^2)^\ell \left(u_0^1 + M^1 u_0^2\right) = \sum_{\ell=0}^{\infty}(M^1 M^2)^\ell u_0^1 + \sum_{\ell=0}^{\infty}(M^1 M^2)^\ell \underbrace{M^1 u_0^2}_{=u_0^2}.$$

The first sum satisfies

$$\sum_{\ell=0}^{\infty}(M^1 M^2)^\ell u_0^1 = \underbrace{u_0^1}_{=0} + \sum_{\ell=1}^{\infty}(M^1 M^2)^\ell u_0^1 = \sum_{\ell=0}^{\infty}(M^1 M^2)^\ell \underbrace{M^1 M^2 u_0^1}_{=M^2 u_0^1},$$

from which we obtain

$$M^{12}\left(u_0^1 + M^1 u_0^2\right) = \sum_{\ell=0}^{\infty}(M^1 M^2)^\ell (u_0^2 + M^2 u_0^1) = M^{21}\left(u_0^2 + M^2 u_0^1\right), \text{ on } \Gamma_C^1.$$

Analogously we get

$$M^{21}\left(u_0^2 + M^2 u_0^1\right) = \sum_{\ell=0}^{\infty}(M^2 M^1)^\ell (u_0^1 + M^1 u_0^2) = M^{12}\left(u_0^1 + M^1 u_0^2\right), \text{ on } \Gamma_C^2.$$

Therefore, the solutions are equal on the boundary of Φ^{12}.

For any $v^{12} \in H_0^1(\Phi^{12})$, we get

$$a^{12}(M^{12}\left(u_0^1 + M^1 u_0^2\right), v^{12}) = a^1(M^{12}\left(u_0^1 + M^1 u_0^2\right), T^{12}v^{12}) = \ell^1(T^{12}v^{12}) = \ell^{12}(v^{12})$$

and

$$a^{12}(M^{21}\left(u_0^2 + M^2 u_0^1\right), v^{12}) = a^2(M^{21}\left(u_0^2 + M^2 u_0^1\right), T^{12}v^{12}) = \ell^2(T^{12}v^{12}) = \ell^{12}(v^{12}),$$

since (u_0^1, u_0^2) solves Problem 2. Hence, we obtain

$$a^{12}(M^{12}\left(u_0^1 + M^1 u_0^2\right) - M^{21}\left(u_0^2 + M^2 u_0^1\right), v^{12}) = 0,$$

for all $v^{12} \in H_0^1(\Phi^{12})$. Since $M^{21}\left(u_0^2 + M^2 u_0^1\right)$ and $M^{21}\left(u_0^2 + M^2 u_0^1\right)$ coincide at the boundary of Φ^{12}, we can conclude from the coercivity of $a(\cdot, \cdot)$ that

$$M^{12}\left(u_0^1 + M^1 u_0^2\right) = M^{21}\left(u_0^2 + M^2 u_0^1\right) \text{ on } \Phi^{12}. \tag{37}$$

Any $v \in H_0^1(\Omega)$ can be represented as $v = T^1 v^1 + T^2 v^2$, where $v^k \in H_0^1(\Omega^k)$. We thus obtain

$$a(u, v) = a(u, T^1 v^1 + T^2 v^2) = a^1(u|_{\Omega^1}, v^1) + a^2(u|_{\Omega^2}, v^2)$$

and

$$\ell(v) = \ell(T^1 v^1 + T^2 v^2) = \ell^1(v^1) + \ell^2(v^2).$$

We have

$$u|_{\Omega^k} = M^{kk'}\left(u_0^k + M^k u_0^{k'}\right)$$

by definition (8) for $k = 1$ and due to (37) for $k = 2$. Since (u_0^1, u_0^2) solves Problem 2, we obtain

$$\begin{aligned} a(u, v) &= a^1(u|_{\Omega^1}, v^1) + a^2(u|_{\Omega^2}, v^2) \\ &= a^1(M^{12}\left(u_0^1 + M^1 u_0^2\right), v^1) + a^2(M^{21}\left(u_0^2 + M^2 u_0^1\right), v^2) \\ &= \ell^1(v^1) + \ell^2(v^2) = \ell(v). \end{aligned}$$

This completes the proof of (ii).

Third we prove (iii). We consider two solutions (u_0^1, u_0^2) and $(\tilde{u}_0^1, \tilde{u}_0^2)$ of Problem 2, which correspond to the same solution u of Problem 1. According to (8), it is obvious that $u_0^1 = \tilde{u}_0^1$ on $\Omega^1 \setminus \Phi^{12}$ and $u_0^2 = \tilde{u}_0^2$ on $\Omega^2 \setminus \Phi^{12}$.

As we showed in (37) the solutions are equal on Φ^{12}, so we can write

$$u = \underbrace{M^{12}\left(u_0^1 + M^1 u_0^2\right)}_{A_1} = \underbrace{M^{21}\left(u_0^2 + M^2 u_0^1\right)}_{A_2} = \underbrace{M^{12}\left(\tilde{u}_0^1 + M^1 \tilde{u}_0^2\right)}_{A_3} = \underbrace{M^{21}\left(\tilde{u}_0^2 + M^2 \tilde{u}_0^1\right)}_{A_4} \text{ on } \Phi^{12}.$$

Since $A_1 = A_3$ and $A_2 = A_4$ we can write

$$\begin{aligned} M^{12}\left(u_0^1 + M^1 u_0^2\right) &= M^{12}\left(\tilde{u}_0^1 + M^1 \tilde{u}_0^2\right), \text{ on } \Phi^{12} \\ M^{21}\left(u_0^2 + M^2 u_0^1\right) &= M^{21}\left(\tilde{u}_0^2 + M^2 \tilde{u}_0^1\right), \text{ on } \Phi^{12}. \end{aligned}$$

We apply $(I - M^1 M^2)$ and $(I - M^2 M^1)$ to the first and second equations respectively and obtain

$$\begin{aligned} \left(u_0^1 - \tilde{u}_0^1\right) + M^1\left(u_0^2 - \tilde{u}_0^2\right) &= 0, \text{ on } \Phi^{12} \\ \left(u_0^2 - \tilde{u}_0^2\right) + M^2\left(u_0^1 - \tilde{u}_0^1\right) &= 0, \text{ on } \Phi^{12}. \end{aligned}$$

Now we apply $-M^2$ to the first equation and add both sides of the two equations and obtain

$$\left(u_0^2 - \tilde{u}_0^2\right) - M^2 M^1\left(u_0^2 - \tilde{u}_0^2\right) = 0, \text{ on } \Phi^{12}$$

or equivalently

$$(I - M^2 M^1)\left(u_0^2 - \tilde{u}_0^2\right) = 0.$$

After applying M^{21} to both sides of the last equation we obtain

$$u_0^2 = \tilde{u}_0^2 \text{ on } \Phi^{12}.$$

Similarly, we conclude $u_0^1 = \tilde{u}_0^1$ on Φ^{12}, which completes the proof.