IGA using Offset-based Overlapping Domain Parameterizations

S. Kargaran[a,c], B. Jüttler[b], T. Takacs[b]

[a] *Doctoral Program Computational Mathematics, Johannes Kepler University Linz, Altenberger Straße 69, A-4040 Linz, Austria*

[b] *Institute of Applied Geometry, Johannes Kepler University Linz, Altenberger Straße 69, A-4040 Linz, Austria*

[c] *Software Competence Center Hagenberg, GmbH (SCCH), Hagenberg, Austria*

Abstract

Isogeometric analysis (IGA) is a numerical method, proposed in [1], that connects computer-aided design (CAD) with finite element analysis (FEA). In CAD the computational domain is usually represented by B-spline or NURBS patches. Given a B-spline or NURBS parameterization of the domain, an isogeometric discretization is defined on the domain using the same B-spline or NURBS basis as for the domain parameterization. Ideally, such an isogeometric discretization allows an exact representation of the underlying CAD model.

CAD models usually represent only the boundary of the object. For planar domains, the CAD model is given as a collection of curves representing the boundary. Finding a suitable parameterization of the interior is one of the major issues for IGA, similar to the mesh generation process in the FEA setting. The objective of this isogeometric parameterization problem is to obtain a set of patches, which exactly represent the boundary of the domain and which are parameterized regularly and without self-intersections. This can be achieved by segmenting the domain into patches which are matching along interfaces, or by covering the domain with overlapping patches. In this paper we follow the second approach.

To construct from a given boundary curve a planar parameterization suitable for IGA, we propose an offset-based domain parameterization algorithm. Given a boundary curve, we obtain an inner curve by generalized offsetting. The inner curve, together with the boundary curve, naturally defines a ring-shaped patch with an associated parameterization. By definition, the ring-shaped patch has a hole, which can be covered by a multi-cell domain. Consequently, the domain is represented as a union of two overlapping subdomains which are regularly parameterized. On such a configuration, one can employ the overlapping multi-patch (OMP) method, as introduced in [2], to solve PDEs on the given domain. The performance of the proposed method is reported in several numerical examples, considering different shape properties of the given boundary curve.

Keywords: Isogeometric analysis; Domain parameterization; Overlapping multi-patch; Generalized offsetting; multi-cell domain

1. Introduction

Isogeometric Analysis (IGA) is a computational approach, proposed by Hughes et al. [1], connecting computer-aided design (CAD) and finite element analysis (FEA). See also [3, 4, 5] for an overview and summary of the framework. In the IGA framework, we use the same basis functions for describing the geometry and for the numerical analysis. Therefore, isoparametric/isogeometric test and trial functions are employed to perform the simulations directly on the geometry representation of the CAD models.

One of the advantages of IGA – when compared to the finite element method (FEM) – is the ability to exactly represent computational domains from CAD using B-spline or NURBS basis functions. In FEM, one first has to obtain a discrete mesh from a given CAD model. This mesh generation process is, in general, expensive. In IGA, the geometry of the computational domain is often given directly from the CAD model. A spline geometry is determined by the degree of the basis functions, knot vectors and control points. However, in a CAD model, the computational domain is usually given by a boundary representation, that is, by a (collection of) boundary curves in 2D or surfaces in 3D. Hence, obtaining a spline representation of the interior of a complex domain from a given CAD description of its boundary is a big challenge in IGA.

The domain parameterization from a given, complicated boundary curve requires, in general, a segmentation of the domain as a first step. More precisely, if a complex boundary curve is given, one needs to divide the interior into several segments, such that each segment can be parameterized using simple patches. Hence, after the segmentation step, appropriate parameterization methods can be performed on each single segment to obtain a parameterization of the entire computational domain.

In the planar case, the representation of the computational domain is given by a collection of boundary curves. There exist several methods to segment and parameterize the interior of the computational domain, e.g., [6] based on patch adjacency graphs, [7] using a template segmentation, or [8, 9] based on a skeleton-based parameterization method. In [10] and [11] the authors proposed algorithms to decompose trimmed surfaces into regular patches. In [12], an algorithm is introduced to decompose a complex planar domain into square-like patches.

The goal is always to cover the interior, starting from the given boundary curves, using appropriate spline patches. The efficiency of the parameterization method depends on the requirements for the resulting parameterization of the domain. To obtain multi-patch volume segmentations for IGA, one may apply the isogeometric segmentation pipeline [13, 14, 15, 16, 17, 18] or [19, 20, 21].

The accuracy of a numerical simulation method performed on the computational domain depends on the quality of the patch parameterization. Therefore, we need to apply a parameterization method, in which the resulting patches are regular, smooth and without any self-intersections. Possible approaches include [22], based on a low-rank parameterization, [23], based on an adaptive template mapping technique, a PDE-based parameterization method proposed in [24], [25], based

on a Teichmüller map, or [26, 27], based on harmonic functions.

It is important to note, that one usually does not allow overlaps between patches, unless one applies a method that can handle overlapping subdomains, such as the one developed in [2].

In this paper, we assume that the interior of the given boundary curve is parameterized by two overlapping patches, a ring-shaped patch with a hole and another patch covering the hole. Such a segmentation process is easy and can be performed for a considerable range of given boundary curves in 2D. To this end, we propose an algorithm to construct a ring-shape patch parameterization. Moreover, allowing overlapping patches, a multi-cell domain covering the remaining hole can also be defined relatively easily. In the following, we consider related approaches for segmentation as well as for parameterizations of computational domains.

The parameterization problem is related to shape optimization. See, e.g., [28, 29], where the authors propose a general framework for isogeometric parameterization and shape optimization by describing several linear and non-linear parameterization methods, for an isogeometric shape optimization model problem, in which the shape can be varied freely.

It is not always clear if a parameterization method yields a parameterization without self-intersections. In [30], the authors propose a method where a constrained optimization problem is solved by minimizing the quadratic energy function corresponding to the first and second derivatives of the planar B-spline parameterization. This method ensures that the resulting parameterization does not have any self-intersection.

For some applications, it is not enough to obtain a multi-patch parameterization, since additional conditions need to be satisfied. This is, e.g., the case if global C^1 smoothness is needed (when solving fourth order PDEs). Such a parameterization problem is considered in [31]. Since the present paper is based on the OMP method in [2] for second order PDEs, a discretization that allows C^0 coupling is sufficient.

In the present paper, we propose a new parameterization method for IGA, the so-called offset-based overlapping domain parameterization (OODP) method. In this method, we employ a regularly parameterized periodic spline curve. Using the periodic curve mentioned above, we generate an inner curve by generalized offsetting. In the next step, we solve an optimization problem based on its first and second derivatives to make the inner curve as smooth as possible. Summing up, we obtain a ring-shaped patch as the resulting parameterization, which by definition has a hole. In the end, we construct an appropriate multi-cell domain covering the hole. Hence, the final computational domain is represented as a union of two overlapping subdomains. We use the overlapping multi-patch (OMP) method proposed in [2] to solve PDEs on the constructed domain.

The OODP method is similar to [32], where the authors proposed an isogeometric scaled boundary parameterization method, which is an extension of the scaled boundary finite element method. There the resulting patch parameterizations are bivariate or trivariate B-spline functions which are suitable for standard Galerkin-based IGA. The difference is that the scaled boundary method results in a singular point in the interior that one needs to handle properly.

Moreover, the OODP method can be seen as an alternative to the method proposed in [33], based on an immersed boundary curve. There one considers a boundary curve immersed in a regular grid. The physical domain is then given by those (cut) grid cells in the interior of the given curve. The grid cells that cover the physical domain can thus be interpreted as a multi-cell domain with cut cells. Since the boundary curve cuts some of the cells, the basis functions need to be cut as well along the prescribed boundary curve. To obtain a stable discretization, some basis functions that have support near the boundary need to be modified.

In our method, instead of cutting the basis functions that have support at the boundary, we drop all functions that are close to the boundary. To be able to approximate any function on the physical domain, the region close to the boundary is covered by the ring-shaped patch. Since we can handle such overlapping patches directly, we do not need to apply any modification of the basis functions.

The structure of this paper is given as follows. In Section 2 we give an overview of the input and output of the OODP method. The OODP algorithm is described and its performance is studied experimentally in Section 3. In Section 4 we extend the method to boundary curves with corners. Finally, we employ the overlapping multi-patch (OMP) method, which is proposed in [2] and briefly summarized in Section 5, to solve second order PDEs on the resulting domain parameterizations. We provide numerical experiments in Section 6.

2. The offset-based overlapping domain parameterization method

Given a simply-connected planar domain represented by its boundary curve, we propose an algorithm to generate a parameterization of the domain consisting of two overlapping patches. First we construct a ring-shaped patch from the boundary curve by generalized offsetting. The part of the domain which is not parameterized by this ring-shaped patch is then covered with a multi-cell domain. We call this approach *offset-based overlapping domain parameterization* (OODP). We first discuss the structure of the input and output in Section 2.1. Then, in Section 2.2 we give a step-by-step overview of the OODP strategy. In Section 2.3 we summarize how the overlapping multi-patch formulation developed in [2] is applied to the resulting two-patch parameterization.

2.1. Structure of input and output

As an input we consider a regularly parameterized, 1-periodic, smooth, simple spline curve $\mathbf{C}_B(t) : \mathbb{R} \to \mathbb{R}^2$, with counter-clockwise orientation, representing the boundary of the domain Ω. We assume $\mathbf{C}_B \in (\mathcal{S}_h^p)^2$, where $\mathcal{S}_h^p$ is a spline space of degree p with 1-periodic knot vector Ξ_h over $\mathbb{R}$. From this boundary curve we construct a ring-shaped patch as is shown in Figure 1 (left). By definition, the ring-shaped patch has a hole in the middle, which we cover with a multi-cell domain. A multi-cell domain consists of a finite set of cells that are defined as follows

$$C_{ij} = [hi, h(i+1)] \times [hj, h(j+1)], \quad (i,j) \in \mathcal{I}_M, \tag{1}$$

where, $\mathcal{I}_M$ is a finite index set. The multi-cell domain $\Omega^{\mathcal{C}}$ is then given as the interior of the union of all C_{ij} for $(i,j) \in \mathcal{I}_M$.

Therefore, the output of our algorithm consists of a ring-shaped patch $\Omega^{\mathcal{R}}$ with parameterization $\mathbf{F}$, covering a neighborhood of the boundary curve, as well as a multi-cell domain $\Omega^{\mathcal{C}}$ covering the hole $\Omega \setminus \Omega^{\mathcal{R}}$. See Figure 1.

2.2. The OODP algorithm

In the following we present the algorithm to construct an OODP for a smooth curve as given in Section 2.1. We will consider curves with corners in more detail in Section 4. Therefore, we construct an open, ring-shaped patch $\Omega^{\mathcal{R}}$ having the parameterization $\mathbf{F} :]0,1[\times [0,1[$, with

$$\mathbf{F}(s,t) = \mathbf{C}_B(t) \cdot (1-s) + \mathbf{C}_I(t) \cdot s, \tag{2}$$

satisfying the periodicity condition $\mathbf{F}(s,0) = \mathbf{F}(s,1)$, where the value at $t = 1$ has to be considered in the limit. By construction, the curves $\mathbf{C}_B$ and $\mathbf{C}_I$ are in the same 1-periodic spline space $\mathbf{C}_B, \mathbf{C}_I \in (\mathcal{S}_h^p)^2$. We want the parameterization to be regular, i.e.,

$$\det \nabla \mathbf{F}(s,t) \geq c > 0, \quad \text{for all } (s,t) \in]0,1[\times [0,1[\,.$$

The algorithm consists of the following steps.

Step 1: Construct a generalized inner offset curve

Given a boundary curve $\mathbf{C}_B$ we define a generalized inner offset curve $\mathbf{C}_O$ as follows

$$\mathbf{C}_O(t) = \mathbf{C}_B(t) + \mu(t) \cdot \mathbf{q}(t), \quad t \in [0,1[\,, \tag{3}$$

where $\mu(t) > 0$ and $\mathbf{q}$ is a predefined quasi-normal vector to the curve $\mathbf{C}_B$, that is, a vector which is continuous in t, 1-periodic, non-tangential and pointing inwards (a generalization of a normal vector). The function μ is assumed to be a 1-periodic spline of degree p, which is determined by a regularized, quadratic minimization problem, which will be described later.

Note that the curve $\mathbf{C}_O$ is a generalization of a classical offset curve. For instance, if $\mathbf{q}$ is the unit normal to $\mathbf{C}_B$ and μ is chosen to be a constant, then $\mathbf{C}_O$ is a classical offset curve.

Step 2: Fit a B-spline parameterization to the ring-shaped patch

Given $\mathbf{C}_B$ and $\mathbf{C}_O$, as constructed in Step 1, we define

$$\widetilde{\mathbf{F}}(s,t) = (1-s)\mathbf{C}_B(t) + s\,\mathbf{C}_O(t) = \mathbf{C}_B(t) + s\,\mu(t)\mathbf{q}(t), \quad (s,t) \in]0,1[\times [0,1[\,. \tag{4}$$

Since $\widetilde{\mathbf{F}}$ is, in general, not a spline parameterization, we solve the following fitting problem

$$\min_{\mathbf{F} \in (\mathbf{S}_1^1 \otimes \mathbf{S}_h^p)^2} \|\mathbf{F} - \widetilde{\mathbf{F}}\|^2_{L^2(]0,1[^2)}, \quad \text{s.t. } \mathbf{F}(0,t) = \widetilde{\mathbf{F}}(0,t) \text{ and } \mathbf{F}(s,0) = \mathbf{F}(s,1), \tag{5}$$

where $\mathbf{S}_1^1$ is the space of polynomials of degree 1 over $]0,1[$ and $\mathbf{S}_h^p$ is the spline space of degree p and mesh size h containing the input curve.

If the curve $\mathbf{C}_O$ obtained in Step 1 is suitable, then the parameterization $\widetilde{\mathbf{F}}$ is regular, cf. Theorem 2. Assuming moreover that the fitting is stable and does not yield a too large distortion, the B-spline parameterization $\mathbf{F}$ is also regular. Thereby, we set the ring-shaped patch to be $\Omega^{\mathcal{R}} = \mathbf{F}(]0,1[\times [0,1[)$.

A short computation confirms that the minimization problem simplifies to

$$\begin{aligned} &\min_{\substack{\mathbf{C}_I \in (\mathbf{S}_h^p)^2 \\ \mathbf{C}_I(0) = \mathbf{C}_I(1)}} \quad \|\mathbf{C}_I(t) - \mathbf{C}_O(t)\|^2_{L^2(]0,1[)}. \end{aligned} \tag{6}$$

Step 3: Cover the hole by a multi-cell domain

From the previous step we obtain a ring-shaped patch $\Omega^{\mathcal{R}} \subset \overline{\Omega}$. We cover the hole $\Omega \setminus \Omega^{\mathcal{R}}$ inside the ring, using a multi-cell domain $\Omega^{\mathcal{C}}$, such that $\Omega \setminus \Omega^{\mathcal{R}} \subset \Omega^{\mathcal{C}} \subset \Omega$. An example of a ring-shaped patch is illustrated in Figure 1 (left). We cover the hole by a multi-cell domain, see Figure 1 (center), such that the patches are overlapping. The resulting parameterization is shown in Figure 1 (right).

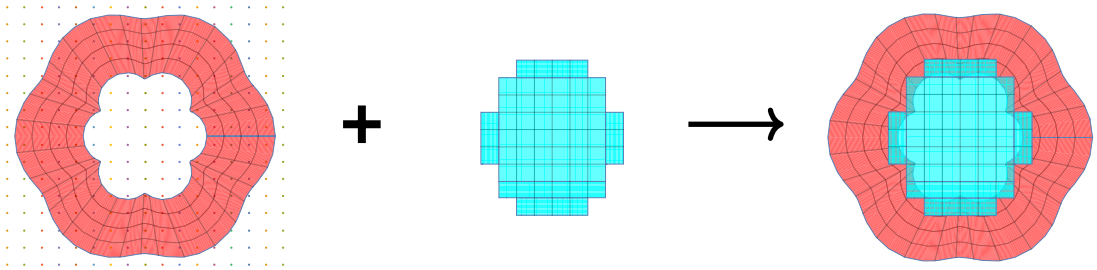


Figure 1: A ring-shaped domain $\Omega^{\mathcal{R}}$ (red) and a multi-cell domain $\Omega^{\mathcal{C}}$ (blue) covering the hole $\Omega \setminus \Omega^{\mathcal{R}}$.

Remark 1. In the numerical experiments, we always cover the hole with a rectangle or a domain which is a union of two overlapping rectangles with matching meshes, to simplify the implementation. In the examples we present here, this is done by hand. However, the task may be automatized relatively easily using the following approach.

To explain the algorithm for creating a multi-cell domain, we consider Figure 1 (left), which depicts a ring-shaped patch $\Omega^{\mathcal{R}}$ and a lattice with distance h in both directions. This lattice creates a set of cells C_{ij} as in (1). All cells that satisfy $C_{ij} \cap (\Omega \setminus \Omega^{\mathcal{R}}) \neq \emptyset$ are selected for the multi-cell domain $\overline{\Omega^{\mathcal{C}}} = \bigcup_{i,j \in \mathcal{I}_M} C_{ij}$. If the ring-shaped patch does not have any artifacts, the obtained

multi-cell domain $\Omega^{\mathcal{C}}$ and the ring-shaped domain are overlapping and, moreover, the cells are fully connected without any gaps and there are no isolated cells. If the lattice distance h is chosen small enough, then $\Omega^{\mathcal{C}} \subset \Omega$. Thereby, we end up with a domain parameterization, which is a union of two overlapping subdomains (see, Figure 1 (right)).

2.3. The overlapping multi-patch formulation

From the algorithm described above, we obtain open, overlapping patches $\Omega^{\mathcal{R}}$ and $\Omega^{\mathcal{C}}$ covering the open domain $\Omega = \Omega^{\mathcal{R}} \cup \Omega^{\mathcal{C}}$. We solve a PDE on the domain Ω using the overlapping multi-patch (OMP) method as proposed in [2].

Assume that $\mathcal{A}(.)$ and $\mathcal{L}(.)$ are appropriate multilinear forms derived from a PDE. The continuous problem looks as follows: find $(u_0^{\mathcal{R}}, u_0^{\mathcal{C}}) \in H_0^1(\Omega^{\mathcal{R}}) \times H_0^1(\Omega^{\mathcal{C}})$ and $(u_M^{\mathcal{R}}, u_M^{\mathcal{C}}) \in H^1(\Omega^{\mathcal{R}}) \times H^1(\Omega^{\mathcal{C}})$ such that

$$\begin{aligned} \mathcal{A}((u_0^{\mathcal{R}} + u_M^{\mathcal{R}}, u_0^{\mathcal{C}} + u_M^{\mathcal{C}}), (v_0^{\mathcal{R}}, v_0^{\mathcal{C}})) &= \mathcal{L}(v_0^{\mathcal{R}}, v_0^{\mathcal{C}}), \quad \forall (v_0^{\mathcal{R}}, v_0^{\mathcal{C}}) \in H_0^1(\Omega^{\mathcal{R}}) \times H_0^1(\Omega^{\mathcal{C}}) \quad (7) \\ C(u_0^{\mathcal{R}}, u_0^{\mathcal{C}}, u_M^{\mathcal{R}}, u_M^{\mathcal{C}}) &= \mathbf{0}. \quad (8) \end{aligned}$$

Here, (7) corresponds to a variational formulation of a PDE and (8) is a coupling condition on the coupling boundaries $\partial\Omega^{\mathcal{R}} \cap \Omega$ and $\partial\Omega^{\mathcal{C}} \cap \Omega$. The formulation is derived in more detail in Section 5.

For discretizing the variational equation (7), we use a standard Galerkin approach. The coupling condition (8) is discretized using a collocation scheme at the coupling boundaries. Since the patch $\Omega^{\mathcal{R}}$ has a 1-periodic parameterization, we use isogeometric basis functions based on standard periodic B-splines to discretize the functions $u_0^{\mathcal{R}}$, $v_0^{\mathcal{R}}$ and $u_M^{\mathcal{R}}$ on $\Omega^{\mathcal{R}}$. The functions $u_0^{\mathcal{C}}$, $v_0^{\mathcal{C}}$ and $u_M^{\mathcal{C}}$ can be discretized using standard B-splines.

3. Constructing a generalized inner offset curve

In this section we provide details on Step 1 of the OODP algorithm, constructing a generalized inner offset curve. In Section 3.1 we present an algorithm to construct the curve, as given in (3), from a given boundary curve and quasi-normal vector along the boundary. In Section 3.2 we study the performance of the algorithm on several examples. Moreover, we consider the dependence of the algorithm on its parameters. In Section 3.3 we discuss the construction of suitable quasi-normal vectors.

3.1. The generalized offsetting algorithm

In the following we propose an algorithm to find, for any given, 1-periodic boundary curve $\mathbf{C}_B$ and quasi-normal vector $\mathbf{q}$, a generalized inner offset curve $\mathbf{C}_O$, such that the resulting ring-shaped patch $\tilde{\mathbf{F}}$ is regular and smooth. In Remarks 4 to 6 we discuss the influence of the parameters which are used in the algorithm. If the parameters are selected properly, the patch parameterization $\tilde{\mathbf{F}}$ is regular.

The generalized offsetting algorithm needs as input a

- 1-periodic boundary curve $\mathbf{C}_B$,
- 1-periodic quasi-normal vector $\mathbf{q}$ to $\mathbf{C}_B$,
- offsetting parameters $0 < c < 1$, $0 < d$, and
- regularization parameters $\alpha \geq 0$ and $\beta \geq 0$.

The steps of the algorithm are give by:

1. Compute
$$\mu_{\max}(t) = \begin{cases} \frac{[\mathbf{C}_B'(t), \mathbf{q}(t)]}{[\mathbf{q}(t), \mathbf{q}'(t)]} & \text{if } 0 < [\mathbf{q}(t), \mathbf{q}'(t)] \\ \infty & \text{otherwise,} \end{cases} \tag{9}$$
for all $t \in [0, 1[$. Here we denote by $[\mathbf{v}, \mathbf{w}]$ the determinant of the 2×2-matrix $(\mathbf{v}, \mathbf{w})$.
2. Set $\mu_{\text{target}}(t) = \min\{c \cdot \mu_{\max}(t), d\}$.
3. Find $\mu \in \mathcal{S}_h^p$ minimizing the quadratic energy functional
$$\|\mu - \mu_{\text{target}}\|^2_{L^2([0,1])} + \alpha \|\mathbf{C}_O^{(1)}\|^2_{L^2([0,1])} + \beta \|\mathbf{C}_O^{(2)}\|^2_{L^2([0,1])} \to \min \tag{10}$$
where $\mathbf{C}_O^{(i)}$ denotes the i-th derivative of $\mathbf{C}_O$.
4. If the resulting generalized offset curve $\mathbf{C}_O$ is simple and orientated counter-clockwise, the algorithm terminates. Otherwise, shrink one of the parameters d, h, α or β and repeat steps 2 to 4.

The following theorem states that $\mu_{\max}$ gives an upper bound on the offsetting distance μ.

Theorem 2. *Assume that the patch parameterization $\tilde{\mathbf{F}}$ is given as in (4) where $\mathbf{C}_B$ is a 1-periodic, counter-clockwise oriented, simple curve and $\mathbf{q}$ is a corresponding quasi-normal vector. Moreover, assume that μ is a 1-periodic and continuous function. If for all t we have $0 < \mu(t) < \mu_{\max}(t)$, where $\mu_{\max}$ is defined as in (9), then the parameterization $\tilde{\mathbf{F}}$ is regular.*

Proof. See Appendix A. □

Lemma 3. *Let $\alpha = \beta = 0$. Then there exists an $h > 0$ and a $d > 0$, both sufficiently small, such that the algorithm terminates.*

Proof. This is a direct consequence of Theorem 2. □

For non-zero regularization parameters, the algorithm may not terminate. In practice, we progressively shrink d by setting $d' = s \cdot d$, with $s \in]0, 1[$ fixed. If the algorithm does not yield a simple, counter-clockwise curve, we decrease the value of the regularization parameters (usually by a factor of 10). The goal is to find a quasi-normal vector and regularization parameters such that the resulting ring-shaped patch covers a large part of Ω.

In the following we discuss the choice of parameters in the generalized offsetting algorithm.

Remark 4. The distance between the outer and the inner curve depends on the choice of the constants c and d. When the constants are smaller, the curve $\mathbf{C}_O$ is closer to the boundary $\mathbf{C}_B$. Here, the goal is to make the ring-shaped patch as wide as possible. On the other hand, if c is close to one, the patch parameterization is almost singular. Therefore, in practice, we set $c = \frac{1}{2}$. If d is too large, the offset curve may not be inside the boundary curve. This issue is resolved by Step 4 of the algorithm.

Remark 5. The shape of the resulting ring-shaped patch depends strongly on the regularization parameters α and β and the choice of the constants c and d. If c and d are small enough and $\alpha = \beta = 0$, then the parameterization is regular. However, nonzero values of α and β often give better results.

Remark 6. In Step 3 of the algorithm we assume $\mu \in \mathcal{S}_h^p$. The choice of the underlying spline space has an effect on the shape and smoothness of the resulting ring-shaped patch. Hence, selecting a higher degree and/or more refined spline space may improve the results.

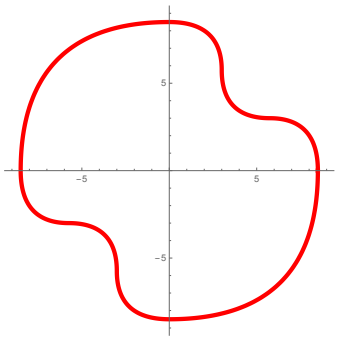
In the following we consider several examples to show how we can create suitable parameterizations using the generalized offsetting algorithm.

3.2. A study of the dependence of the generalized offsetting algorithm on its parameters

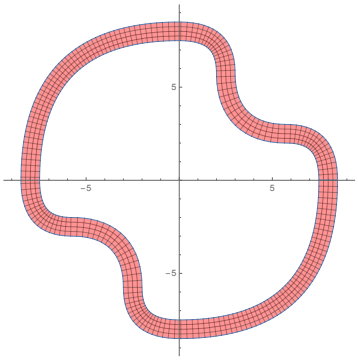
We use the generalized offsetting algorithm to obtain a suitable ring-shape parameterization from several given boundary curves. In Example 1 the function $\mathbf{q}$ is chosen to be the normal vector. This does not always result in a satisfactory parameterization, as can be seen in Example 2. Hence, a suitable quasi-normal vector to the outer curve needs to be defined. In Example 3 we consider a star-shaped domain and we discus the importance of the choice of $\mathbf{q}$ to obtain a regular ring-shape patch of sufficient width. In Section 3.3, we discuss how to improve the choice of $\mathbf{q}$ (locally) for a given boundary curve. This is considered in Example 4.

Remark 7. In the following examples, we assume that the function μ is a periodic B-spline of degree three. The size h of the spline space for μ is set to 0.02 for all the examples.

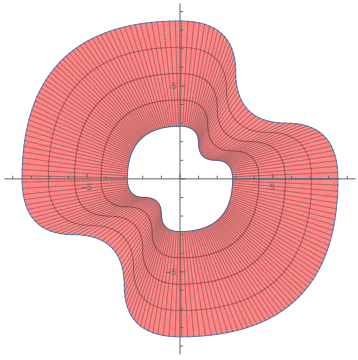
Example 1. We consider the peanut-shaped curve $\mathbf{C}_B^1$, which is shown in Figure 2a. In Figure 2b we show the resulting patch parameterization where the function $\mathbf{q}$ is the normal vector $\mathbf{n}_B$ to the curve $\mathbf{C}_B^1$ and α and β are set to zero. In Figure 2c the resulting parameterization is depicted for when we set $\mathbf{q} = -\mathbf{C}_B^1$. This is a valid choice, since the domain is star shaped with respect to the origin. The patch parameterizations in Figures 2d and 2e are obtained for $\alpha = 10^{-7}$ and $\beta = 10^{-7}$, respectively. As one can see, having $\beta > 0$ yields a smoother parameterization for $\mathbf{q} = \mathbf{n}_B$. In this case, the choice $\mathbf{q} = -\mathbf{C}_B^1$ as in Figure 2c yields the largest patch, which can be most easily covered by a multi-cell domain.



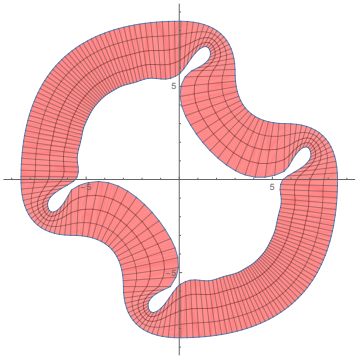
(a) Peanut-shaped curve $\mathbf{C}_B^1$.



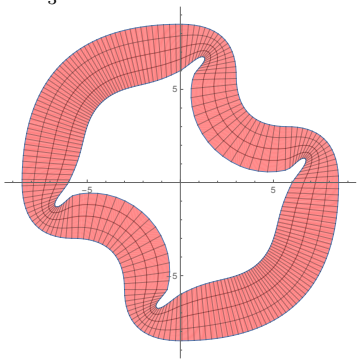
(b) $d = 1$, $c = 1$, $\alpha = \beta = 0$ and $\mathbf{q} = \mathbf{n}_B$.



(c) $d = \frac{2}{3}$, $c = 1$, $\alpha = \beta = 0$ and $\mathbf{q} = -\mathbf{C}_B^1$.



(d) $d = 3$, $c = 1$, $\alpha = 10^{-7}$ and $\mathbf{q} = \mathbf{n}_B$.



(e) $d = 3$, $c = 1$, $\beta = 10^{-7}$ and $\mathbf{q} = \mathbf{n}_B$.

Figure 2: Different parameterizations of a domain inside a peanut-shaped curve $\mathbf{C}_B^1$. Note that in this figure (and in the following) the mesh is given for visualization purposes and does not correspond to the underlying Bézier mesh. Here $\mathbf{n}_B$ stands for the normal vector to the outer curve.

Example 2. In this example we consider a curve $\mathbf{C}_B^2$ shaped like planet B-612, as shown in Figure 3a. The curve is named in reference to [34]. In Figure 3b we visualize the resulting patch parameterization for $\mathbf{q} = \mathbf{n}_B$, $\mu_{\text{target}} = \min\{\frac{1}{2}, \frac{1}{2}\mu_{\max}\}$ and $\alpha = \beta = 0$. One can see that the parameterization has self-intersections. Therefore, we set $\mu_{\text{target}} = \min\{\frac{1}{5}, \frac{1}{2}\mu_{\max}\}$ with $\alpha = \beta = 0$ in Figure 3c and $\beta = 10^{-6}$ in Figure 3d. The resulting parameterization in Figure 3d does not have any self-intersections, but one part of the patch is very slim. In this case, it is not possible to cover the hole with a multi-cell domain, without having a very small mesh size. Hence, we need to define a different function $\mathbf{q}$, such that the resulting parameterization becomes wider. For this aim, instead of $\mathbf{q}$ being the normal vector, we define $\mathbf{q} = -\mathbf{C}_B^2$. This is a valid quasi-normal vector, since the curve is star-shaped with respect to a ball at the origin. The result is illustrated in Figure 3e. In this case, covering the hole is easily possible.

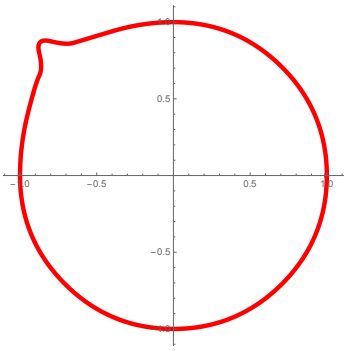
Example 3. We consider the star-shaped boundary curve illustrated in Figure 4a. Here, we compare the parameterizations resulting from different choices of $\mathbf{q}$ and different choices for the parameters α and β. The results using $\mathbf{q}$ as the normal vector are illustrated in Figures 4b and 4d. The results using a more general quasi-normal vector (the normal vector of a circle ($\mathbf{n}_C$)) are depicted in Figures 4c and 4e. As remarked earlier, the quasi-normal vector yields better results. Moreover, having $\beta > 0$ yields a smoother patch parameterization.

3.3. A discussion on constructions for the quasi-normal vector $\mathbf{q}$

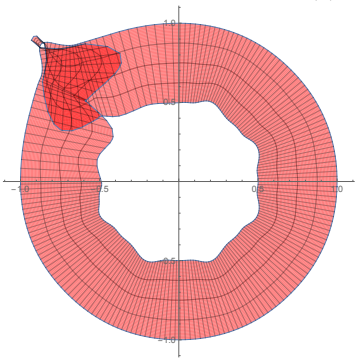
The goal of this section is to obtain a smooth quasi-normal vector for a boundary curve with fine details, as illustrated in Figure 5a. Such a quasi-normal vector can be computed as the normal vector of a smoother curve as it is shown in blue in Figure 5b. To this end we go through the following steps, given a boundary curve $\mathbf{C}_B$:

- We approximate $\mathbf{C}_B$ by a smoother curve $\mathbf{C}_S$, see Figures 5a and 5b.
- We compute the normal vectors $\mathbf{n}_B$ and $\mathbf{n}_S$ to the curves $\mathbf{C}_B$ and $\mathbf{C}_S$, respectively.
- If, for each t, the normal vector $\mathbf{n}_S(t)$ is a valid quasi-normal direction at $\mathbf{C}_B(t)$, we set $\mathbf{q}(t) = \mathbf{n}_S(t)$.
- If this is not the case, we repeat the process with another approximation $\mathbf{C}_S^*$ with $\|\mathbf{n}_S^* - \mathbf{n}_B\| < \|\mathbf{n}_S - \mathbf{n}_B\|$.

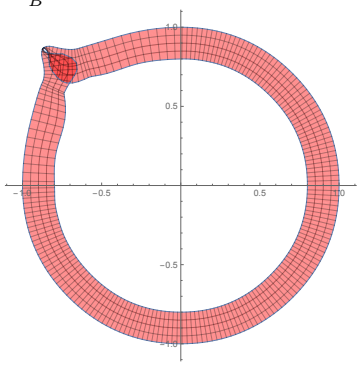
For a better understanding we present the following example.



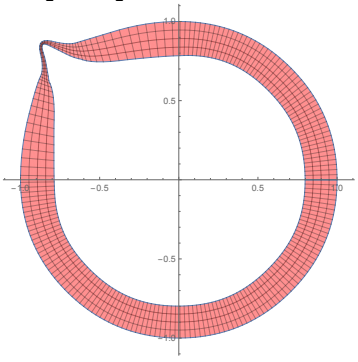
(a) Planet B-612 curve $\mathbf{C}_B^2$.



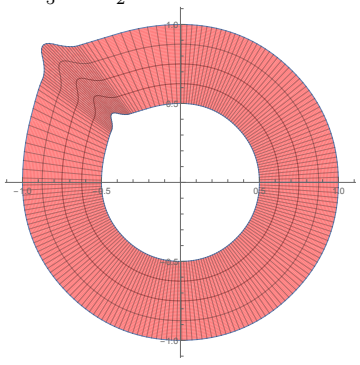
(b) $d = \frac{1}{2}$, $c = \frac{1}{2}$, $\alpha = \ \beta = \ 0$, and $\mathbf{q} = \mathbf{n}_B$.



(c) $d = \frac{1}{5}$, $c = \frac{1}{2}$, $\alpha = \ \beta = \ 0$, and $\mathbf{q} = \mathbf{n}_B$.

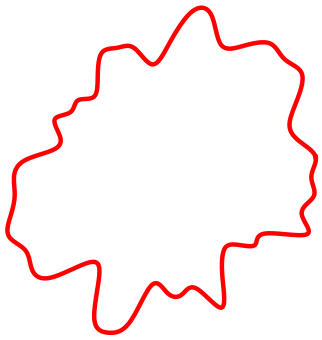


(d) $d = \frac{1}{5}$, $c = \frac{1}{2}$, $\beta = \ 10^{-6}$, and $\mathbf{q} = \mathbf{n}_B$.

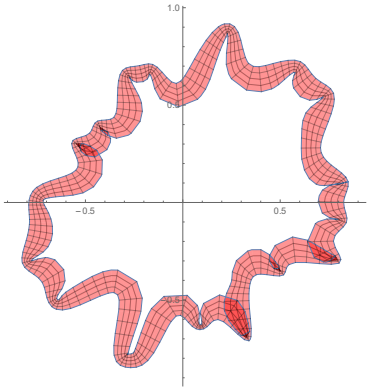


(e) $d = \frac{1}{2}$, $c = \frac{1}{2}$, $\alpha = \ \beta = \ 0$, and $\mathbf{q} = -\mathbf{C}_B^2$.

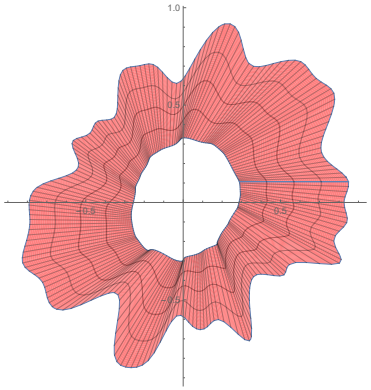
Figure 3: Different parameterizations of a domain inside the curve $\mathbf{C}_B^2$ shaped like planet B-612.



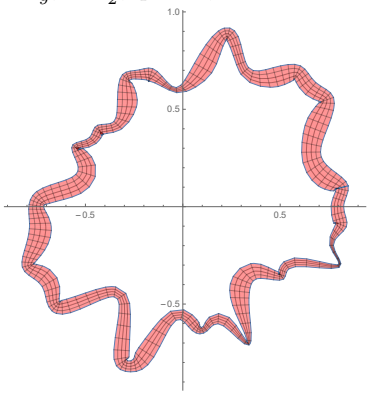
(a) Star-shaped curve.



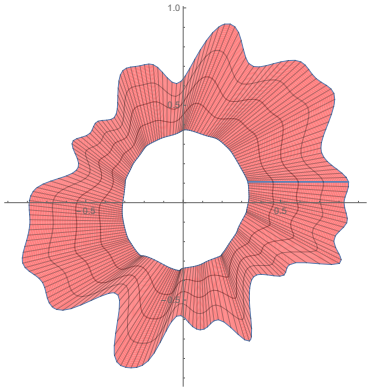
(b) $d = \frac{1}{9}$, $c = \frac{1}{2}$, $\mathbf{q} = \mathbf{n}_B$, and $\alpha = 3 \times 10^{-9}$.



(c) $d = 1$, $c = \frac{1}{2}$, $\mathbf{q} = \mathbf{n}_C$, and $\alpha = 9 \times 10^{-3}$.



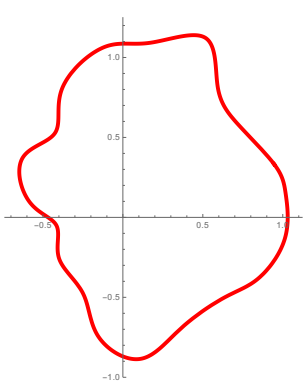
(d) $d = \frac{1}{9}$, $c = \frac{1}{2}$, $\mathbf{q} = \mathbf{n}_B$, and $\alpha = 3 \times 10^{-9}$.



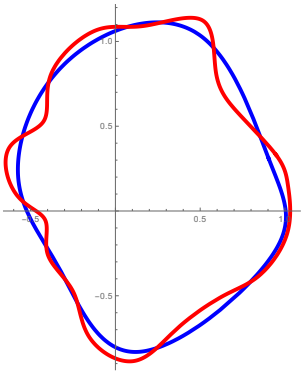
(e) $d = 1$, $c = \frac{1}{2}$, $\mathbf{q} = \mathbf{n}_C$, and $\alpha = 9 \times 10^{-5}$.

Figure 4: Applying the generalized offsetting algorithm on a star-shaped boundary curve. $\mathbf{n}_B$ stands for the normal vector of the outer curve.

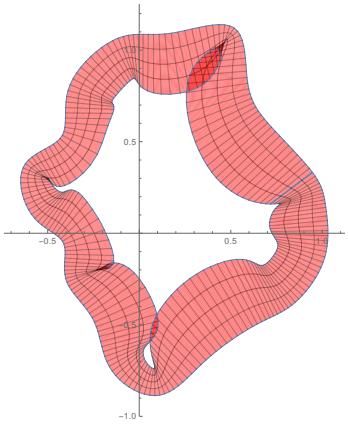
Example 4. We consider the boundary curve $\mathbf{C}_B$ shown in Figure 5a. Moreover, in Figure 5b, the boundary curve $\mathbf{C}_B$ is depicted together with a regularized, smoother curve $\mathbf{C}_S$. The corresponding normal vectors are denoted by $\mathbf{n}_B$ and $\mathbf{n}_S$, respectively. The curve $\mathbf{C}_S$ is an approximation of $\mathbf{C}_B$, having fewer geometric details. In Figures 5c and 5d we depict the parameterizations where $\mathbf{q}$ is chosen to be $\mathbf{q} = \mathbf{n}_B$ (left) and $\mathbf{q} = \mathbf{n}_S$ (right). In the latter case, the resulting patch parameterization is regular, and without any self-intersection. Moreover, the hole in the middle can be covered easily using a multi-cell domain.



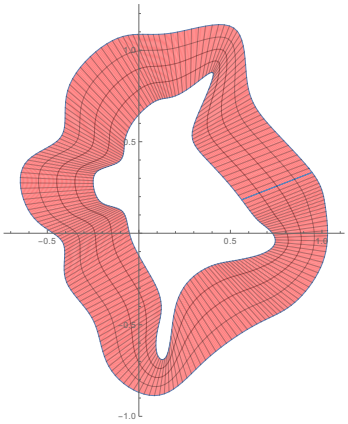
(a) The given boundary curve $\mathbf{C}_B$.



(b) $\mathbf{C}_B$ plotted together with the smoother curve $\mathbf{C}_S$.



(c) $d = \frac{2}{5}$, $c = \frac{1}{2}$, $\mathbf{q} = \mathbf{n}_B$, and $\alpha = 10^{-7}$.



(d) $d = \frac{2}{5}$, $c = \frac{1}{2}$, $\mathbf{q} = \mathbf{n}_S$, and $\alpha = 10^{-7}$.

Figure 5: Constructing a quasi-normal vector using a smooth approximation of a given boundary curve.

4. Treatment of corners on the boundary

In this section, we consider boundary curves with corners, both convex and non-convex. See Figure 6 for a visualization. We explain how to extend the generalized offsetting algorithm to

such curves. The algorithm yields a set of patches for each curve segment and each corner. The collection of patches can then be interpreted as a single ring-shaped spline manifold.

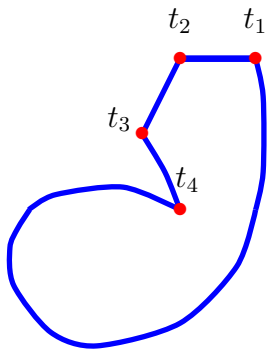


Figure 6: An example of a boundary curve $\mathbf{C}_B$ with corners.

We assume to have given a 1-periodic curve $\mathbf{C}_B \in (\mathbf{S}_h^p)^2$ with corners t_i, for $1 \leq i \leq n$. Each corner corresponds to a knot of multiplicity p. The corners split the curve into segments $\mathbf{C}_{B,i}$ with $t \in [t_{i-1}, t_i]$, i.e.,

$$\begin{aligned} &\mathbf{C}_{B,1}(t) \colon t \in [t_1, t_2] \\ &\mathbf{C}_{B,2}(t) \colon t \in [t_2, t_3] \\ &\vdots \\ &\mathbf{C}_{B,n}(t) \colon t \in [t_n, t_1 + 1]. \end{aligned} \tag{11}$$

We assume that no segment $\mathbf{C}_{B,i}$ contains any additional corners, i.e., $\mathbf{C}_{B,i} \in C^1([t_i, t_{i+1}])$ and $\|\mathbf{C}_{B,i}\| > 0$. To obtain a ring-shape parameterization for such a curve we need to go through the following steps, which are also visualized in Figure 7:

- For each segment $\mathbf{C}_{B,i}$ we define a corresponding parameter interval $\mathbf{X}_i^F$, with

$$\mathbf{X}_i^F = \begin{cases} [t_i + \delta, t_{i+1} - \delta] & \text{if both } t_i \text{ and } t_{i+1} \text{ are convex,} \\ [t_i + \delta, t_{i+1}] & \text{if } t_i \text{ is convex and } t_{i+1} \text{ is non-convex,} \\ [t_i, t_{i+1} - \delta] & \text{if } t_i \text{ is non-convex and } t_{i+1} \text{ is convex,} \\ [t_i, t_{i+1}] & \text{if both } t_i \text{ and } t_{i+1} \text{ are non-convex.} \end{cases}$$

  We restrict the spline space $\mathbf{S}_h^p$ to $\mathbf{X}_i^F$, which is denoted by $\mathbf{S}_h^p|_{\mathbf{X}_i^F}$. We have given a quasi-normal vector $\mathbf{q}_i$ for each segment $\mathbf{C}_{B,i}$. The quasi-normals are not continuous at the corners, i.e., in general $\mathbf{q}_{i-1}(t_i) \neq \mathbf{q}_i(t_i)$.

- If t_i is a convex corner, we construct a patch parameterization $\mathbf{P}_i$ covering a neighborhood of the corner as a Coons patch interpolating the boundary curves

$$\begin{aligned}\mathbf{P}_i(u,0) &= \mathbf{C}_{B,i-1}(t_i - u\cdot\delta)\\ \mathbf{P}_i(u,1) &= \mathbf{C}_{B,i}(t_i+\delta) + u\cdot\mu_i(t_i+\delta)\,\mathbf{q}_i(t_i+\delta)\\ \mathbf{P}_i(0,v) &= \mathbf{C}_{B,i}(t_i + v\cdot\delta)\\ \mathbf{P}_i(1,v) &= \mathbf{C}_{B,i-1}(t_i-\delta) + v\cdot\mu_{i-1}(t_i-\delta)\,\mathbf{q}_{i-1}(t_i-\delta).\end{aligned} \tag{12}$$

- If t_i is non-convex, we construct a patch parameterization $\mathbf{P}_i$ as a parallelogram from given quasi-normal vectors $\mathbf{q}_i(t_i)$ and $\mathbf{q}_{i-1}(t_i)$.

- We apply the generalized offsetting algorithm to the segment $\mathbf{C}_{B,i}$ on the parameter interval $\mathbf{X}_i^F$ and obtain a parameterization $\tilde{\mathbf{F}}_i : [0,1]\times\mathbf{X}_i^F \to \mathbb{R}^2$. Instead of periodicity constraints, we now have to satisfy continuity constraints to couple with the corner patches.

- We define a local parameterization

$$\mathbf{F}_i(s,t) = \mathbf{C}_{B,i}(t)\cdot(1-s) + \mathbf{C}_{I,i}(t)\cdot s, \quad s\in[0,1] \text{ and } t\in\mathbf{X}_i^F, \tag{13}$$

  such that the inner curve satisfies $\mathbf{C}_{I,i}\in(\mathbf{S}_h^p|_{\mathbf{X}_i^F})^2$.

We visualize the construction in Figure 7, where the corners t_i and t_{i+1} are convex, whereas t_{i-1} is non-convex. Hence, we have $\mathbf{X}_{i-1}^F = [t_{i-1}, t_i-\delta]$ and $\mathbf{X}_i^F = [t_i+\delta, t_{i+1}-\delta]$.

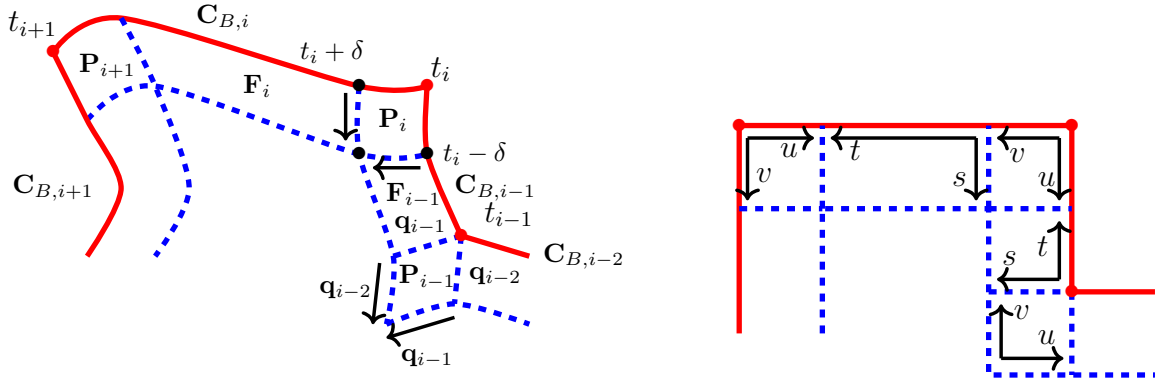
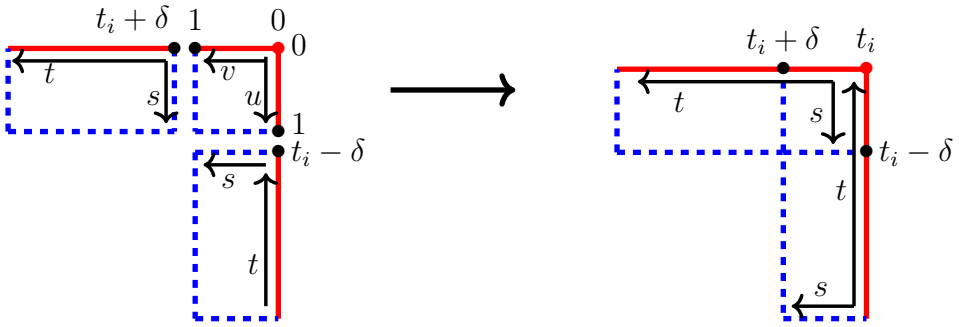


Figure 7: A collection of patches (left) and corresponding parameter domains (right).

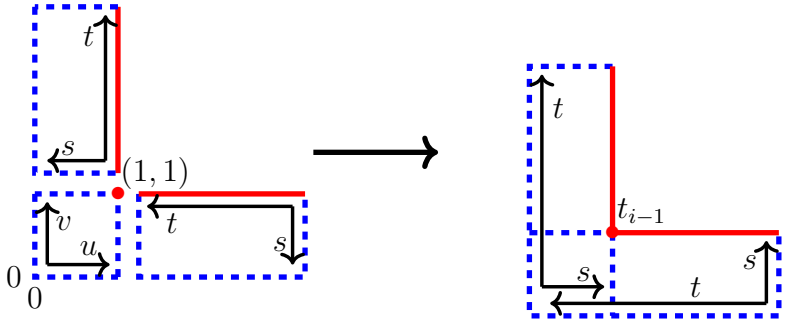
Remark 8. Note that the inner curves need to satisfy

$$\mathbf{C}_{I,i}(t_i+\delta) = \mathbf{C}_{I,i-1}(t_i-\delta) \tag{14}$$

for each convex corner t_i. For each non-convex corner t_{i-1} the angle between $\mathbf{q}_{i-2}(t_{i-1})$ and $\mathbf{q}_{i-1}(t_{i-1})$ must be greater than zero.



(a) Convex corner.



(b) Non-convex corner.

Figure 8: Gluing the local patch parameterizations.

Remark 9. The value of δ can be chosen differently for each corner and each direction. Here, for simplicity, we assumed that δ is the same for all cases.

As output we obtain a ring-shape patch as shown in Figure 7 (left). This domain can be computed by creating a sequence of patches that are glued appropriately using a manifold-like structure. The continuity of the corresponding parameter domains is visualized in Figure 7 (right). Such a configuration can be interpreted as a single spline manifold as defined in [35]. This process is visualized for convex corners in Figure 8a and for non-convex corners in Figure 8b. For details on the construction we refer to [36].

5. Overlapping multi-patch formulation and isogeometric discretization

In this section we assume, for simplicity, that the domain Ω is constructed as the union of a periodic ring-shaped patch $\Omega^{\mathcal{R}}$ and a rectangular patch $\Omega^{\mathcal{C}}$, i.e.,

$$\Omega = \Omega^{\mathcal{R}} \cup \Omega^{\mathcal{C}}.$$

As a model problem, we consider a Poisson problem on the domain Ω, with zero Dirichlet boundary conditions, which can be represented by a variational formulation as follows.

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

$$a(u,v) = \ell(v) \quad \forall v \in H_0^1(\Omega), \tag{15}$$

where

$$a(u,v) = \int_\Omega \nabla u \, \nabla v \, \mathrm{d}\boldsymbol{\xi} \quad \textit{and} \quad \ell(v) = \int_\Omega f v \, \mathrm{d}\boldsymbol{\xi}. \tag{16}$$

We adopt the notation and definitions from [2]. For $k \in \{\mathcal{R}, \mathcal{C}\}$ we define the following extension operator

$$M^k v = \begin{cases} v & \text{on } \Gamma_C^k \\ 0 & \text{on } \Gamma_D^k, \end{cases} \tag{17}$$

where, Γ_C^k and Γ_D^k are called the coupling and Dirichlet boundaries, respectively.

For $(u^\mathcal{R}, u^\mathcal{C}), (v^\mathcal{R}, v^\mathcal{C}) \in H^1(\Omega^\mathcal{R}) \times H^1(\Omega^\mathcal{C})$, we define

$$\mathcal{A}((u^\mathcal{R}, u^\mathcal{C}), (v^\mathcal{R}, v^\mathcal{C})) = a^\mathcal{R}(u^\mathcal{R}, v^\mathcal{R}) + a^\mathcal{C}(u^\mathcal{C}, v^\mathcal{C})$$

and

$$\mathcal{L}(v^\mathcal{R}, v^\mathcal{C}) = \ell^\mathcal{R}(v^\mathcal{R}) + \ell^\mathcal{C}(v^\mathcal{C}),$$

where

$$a^k(u,v) = \int_{\Omega^k} \nabla u \, \nabla v \, \mathrm{d}\boldsymbol{\xi} \quad \text{and} \quad \ell^k(v) = \int_{\Omega^k} f v \, \mathrm{d}\boldsymbol{\xi}, \quad k \in \{\mathcal{R}, \mathcal{C}\}. \tag{18}$$

Moreover, we introduce functions $u_0^k \in H_0^1(\Omega^k)$, satisfying

$$u^k = u_0^k + M^k u^{k'} \quad k \in \{\mathcal{R}, \mathcal{C}\}, \tag{19}$$

where k' such that $\{k, k'\} = \{\mathcal{R}, \mathcal{C}\}$. In [2] it was shown that equation (19) is solvable under mild conditions on $M^\mathcal{R}$ and $M^\mathcal{C}$. We obtain the following coupled problem.

Problem 2. *Find $(u_0^\mathcal{R}, u_0^\mathcal{C}) \in H_0^1(\Omega^\mathcal{R}) \times H_0^1(\Omega^\mathcal{C})$ and $(u_M^\mathcal{R}, u_M^\mathcal{C}) \in H^1(\Omega^\mathcal{R}) \times H^1(\Omega^\mathcal{C})$ such that*

$$\begin{aligned} \mathcal{A}((u_0^\mathcal{R} + u_M^\mathcal{R}, u_0^\mathcal{C} + u_M^\mathcal{C}), (v_0^\mathcal{R}, v_0^\mathcal{C})) &= \mathcal{L}(v_0^\mathcal{R}, v_0^\mathcal{C}), \quad \forall (v_0^\mathcal{R}, v_0^\mathcal{C}) \in H_0^1(\Omega^\mathcal{R}) \times H_0^1(\Omega^\mathcal{C}) && (20) \\ u_M^\mathcal{R} - M^\mathcal{R}(u_0^\mathcal{C} + u_M^\mathcal{C}) &\equiv 0 && (21) \\ u_M^\mathcal{C} - M^\mathcal{C}(u_0^\mathcal{R} + u_M^\mathcal{R}) &\equiv 0, && (22) \end{aligned}$$

where we obtained the coupling constraints (21) and (22) by replacing $M^k u^{k'}$ with u_M^k in (19).

In the following we summarize the isogeometric discretization of Problem 2. Since $\Omega^{\mathcal{R}}$ is a periodic ring-shaped patch and $\Omega^{\mathcal{C}}$ is rectangular, we use standard periodic isogeometric functions for the discretization of the functions corresponding to $\Omega^{\mathcal{R}}$ and standard B-spline basis functions for the functions corresponding to the rectangular patch $\Omega^{\mathcal{C}}$. We assume that $\mathbf{G}^{\mathcal{R}}$ is a spline geometry mapping, which is periodic in the first direction, mapping the parameter domain $\widehat{\Omega}^{\mathcal{R}} = [0,1[\times]0,1[$ onto the physical subdomain $\Omega^{\mathcal{R}}$, i.e., $\Omega^{\mathcal{R}} = \mathbf{G}^{\mathcal{R}}(\widehat{\Omega}^{\mathcal{R}})$. We assume that the rectangular patch $\Omega^{\mathcal{C}}$ has a parameterization $\mathbf{G}^{\mathcal{C}} :]0,1[^2 \to \Omega^{\mathcal{C}}$, with $\mathbf{G}^{\mathcal{C}} \in \mathbb{P}^{(1,1)}$. The isogeometric spaces can then be defined as

$$\begin{aligned} V_h^{\mathcal{R}} &= \operatorname{span}\left\{\varphi \in L^2(\Omega^{\mathcal{R}}) : \varphi \circ \mathbf{G}^{\mathcal{R}} \in \mathcal{S}^p_{h,\text{per}} \times \mathcal{S}^p_h\right\}, \\ V_h^{\mathcal{C}} &= \operatorname{span}\left\{\varphi \in L^2(\Omega^{\mathcal{R}}) : \varphi \circ \mathbf{G}^{\mathcal{C}} \in \mathcal{S}^p_h \times \mathcal{S}^p_h\right\}, \end{aligned} \tag{23}$$

where $\mathcal{S}^p_{h,\text{per}}$ denotes a periodic spline space. We define the interior functions for each subdomain as

$$\begin{aligned} V_{0h}^{\mathcal{R}} &= V_h^{\mathcal{R}} \cap H_0^1(\Omega^{\mathcal{R}}), \\ V_{0h}^{\mathcal{C}} &= V_h^{\mathcal{C}} \cap H_0^1(\Omega^{\mathcal{C}}). \end{aligned}$$

The spaces of coupling functions are defined as follows

$$\begin{aligned} V_{ch}^k &= \operatorname{span}\left\{\beta_i^k \in V_h^k \mid i \in \mathcal{I}_c^k\right\} \subset H^1(\Omega^k), \text{with} \\ \mathcal{I}_c^k &= \left\{i \in \mathcal{I}^k \mid \operatorname{supp}\beta_i^k \cap \Gamma_C^k \neq \emptyset\right\}, \end{aligned}$$

where $k \in \{\mathcal{R}, \mathcal{C}\}$. Hence, Problem 2 can be discretized as follows

Problem 3. *Find* $(u_{0h}^{\mathcal{R}}, u_{0h}^{\mathcal{C}}) \in V_{0h}^{\mathcal{R}} \times V_{0h}^{\mathcal{C}}$ *and* $(u_{Mh}^1, u_{Mh}^2) \in V_{ch}^{\mathcal{R}} \times V_{ch}^{\mathcal{C}}$ *such that*

$$\mathcal{A}(u_{0h}^{\mathcal{R}} + u_{Mh}^{\mathcal{R}}, u_{0h}^{\mathcal{C}} + u_{Mh}^{\mathcal{C}}), (v_{0h}^{\mathcal{R}}, v_{0h}^{\mathcal{C}})) = \mathcal{L}(v_{0h}^{\mathcal{R}}, v_{0h}^{\mathcal{C}}) \quad \forall (v_{0h}^{\mathcal{R}}, v_{0h}^{\mathcal{C}}) \in V_{0h}^{\mathcal{R}} \times V_{0h}^{\mathcal{C}} \tag{24}$$

$$u_{Mh}^{\mathcal{R}} - M_h^{\mathcal{R}}(u_{0h}^{\mathcal{C}} + u_{Mh}^{\mathcal{C}}) = 0 \quad \text{on } \Omega^{\mathcal{R}} \tag{25}$$

$$u_{Mh}^{\mathcal{C}} - M_h^{\mathcal{C}}(u_{0h}^{\mathcal{R}} + u_{Mh}^{\mathcal{R}}) = 0 \quad \text{on } \Omega^{\mathcal{C}}. \tag{26}$$

In Problem 3, $M_h^{\mathcal{R}}$ and $M_h^{\mathcal{C}}$ are suitable discretizations of the operators $M^{\mathcal{R}}$ and $M^{\mathcal{C}}$. We assume that the discretized extension operators are *collocation-based extension operators* (CEO) as defined in [2, Section 4.4].

6. Numerical experiments

In all numerical experiments we solve the Poisson problem using the OMP method. Except for Example 5, we always consider the exact solution

$$u(x,y) = \sin(\pi x)\sin(\pi y). \tag{27}$$

The domain parameterizations were created using the generalized offsetting algorithm presented in Section 3. In Examples 5 and 6 we consider domains with a smooth boundary. In Examples 7 and 8 we consider domains with corners that are convex (inner angle $< \pi$) and non-convex (inner angle $> \pi$), respectively.

Example 5. The peanut shaped domain, illustrated in Figure 9a, is constructed as a union of two overlapping patches. The boundary curve is composed of four segments that meet C^0 and the resulting ring-shaped patch is parameterized by periodic quadratic B-spline basis functions. The hole is covered with a multi-cell domain. We employ the following exact solution

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

In Figures 9b we plot the numerical solution for a total of 1080 DOFs and using quadratic B-splines. L^2 and H^1 errors are shown in Figure 9c. According to the numerical results, the convergence rates of errors are optimal.

Example 6. We consider the domain from Example 4. The parameterization of the ring-shaped patch is the one depicted in Figure 5d. The ring-shaped patch is parameterized by periodic cubic B-splines. The resulting OMP representation is depicted in Figure 10a.

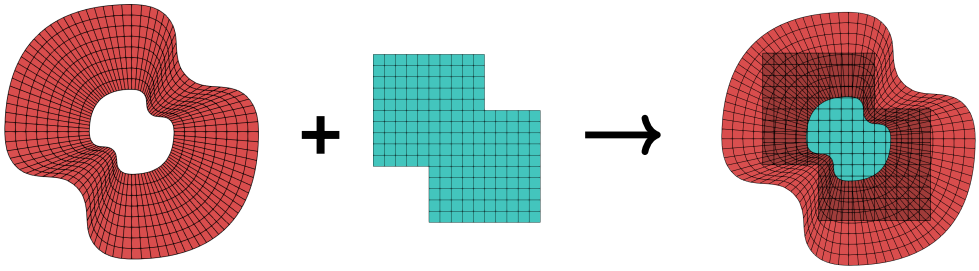
We show the numerical solution for a total of 2263 DOFs and using cubic B-splines, in Figures 10b. The L^2 and H^1 errors for discretizations of degree $p = 3$ and $p = 4$ are shown in Figure 10c. All convergence rates are optimal.

Example 7. We consider a heart shaped domain, which is illustrated in Figure 11a. In this example, the geometry is a non-convex domain with a corner near the top. The construction of the ring-shaped patch is done as explained in Section 4. The ring-shaped subdomain is parameterized with quadratic B-splines.

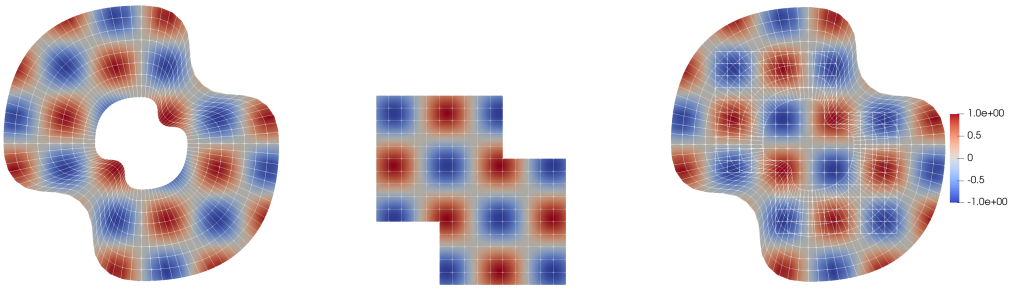
The numerical solution for a total of 1460 DOFs and using quadratic B-splines is illustrated in Figures 11b. L^2 and H^1 errors are shown in Figure 11c. The convergence rates of all errors are optimal.

Example 8. The drop shaped domain illustrated in Figure 12a, is composed of two overlapping patches. The geometry is convex with a corner on the top. The ring-shaped part is constructed with quadratic B-splines according to the strategy in Section 4.

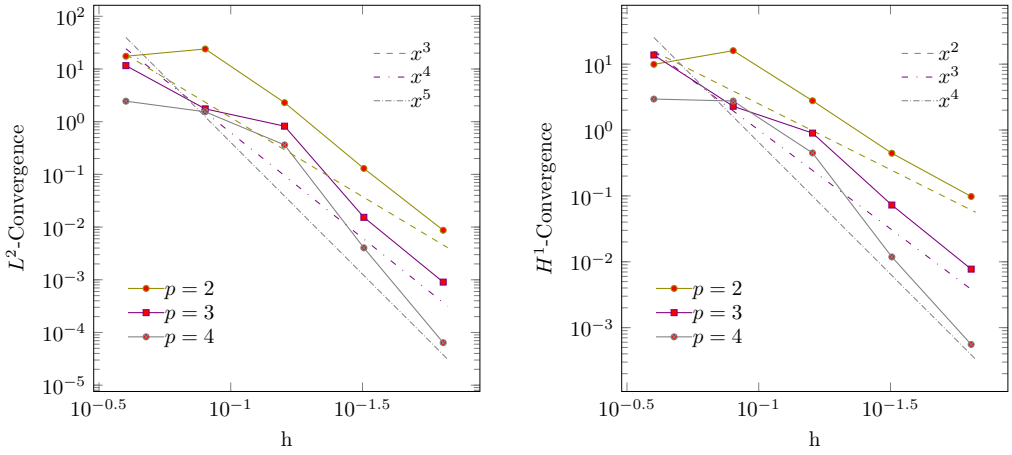
In Figures 12b we show the numerical solution for a total of 1460 DOFs and using quadratic B-splines. L^2 and H^1 errors are shown in Figure 12c. The observed convergence rates of all errors are optimal.



(a) Peanut shaped domain.

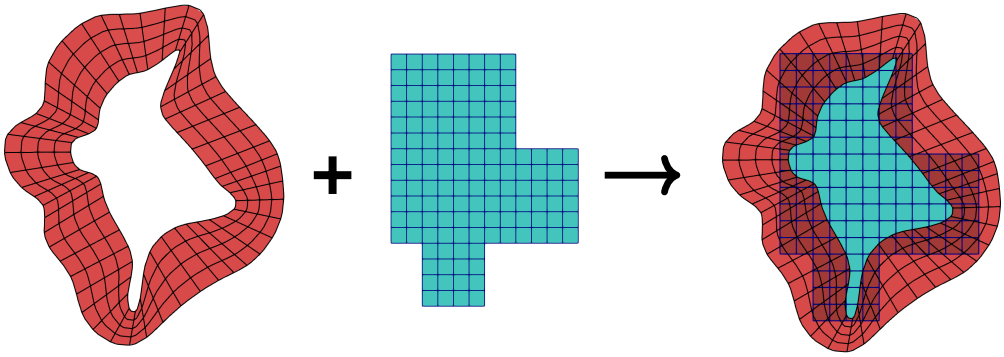


(b) Local solutions (left and center) and local solutions plotted together (right).

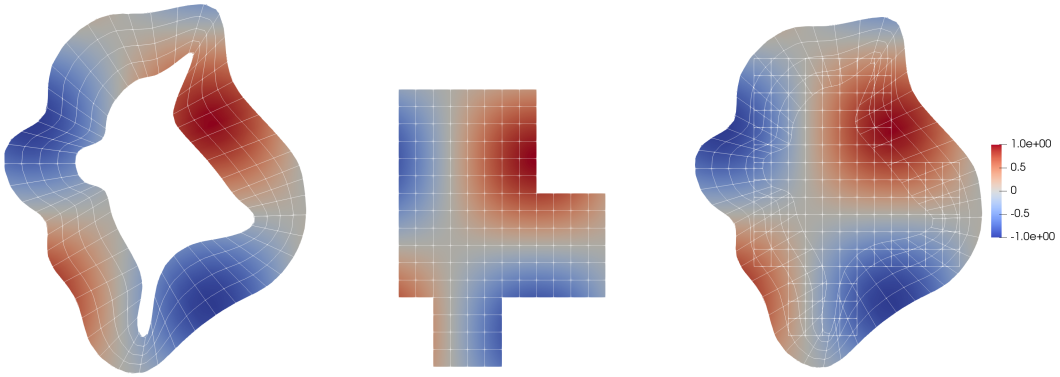


(c) L^2 (left) and H^1 (right) errors for $p = 2, 3, 4$.

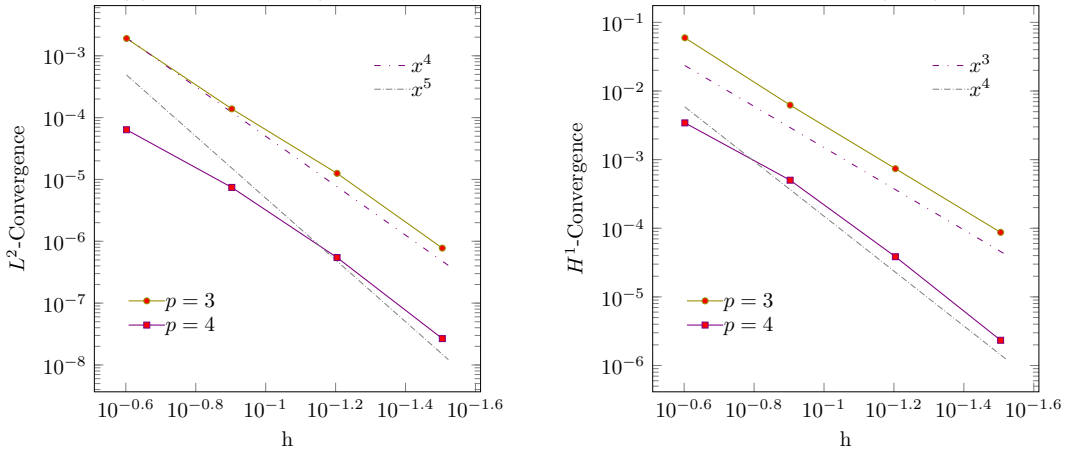
Figure 9: Numerical results for solving the Poisson problem on a peanut shaped domain.



(a) Domain with a random smooth boundary.

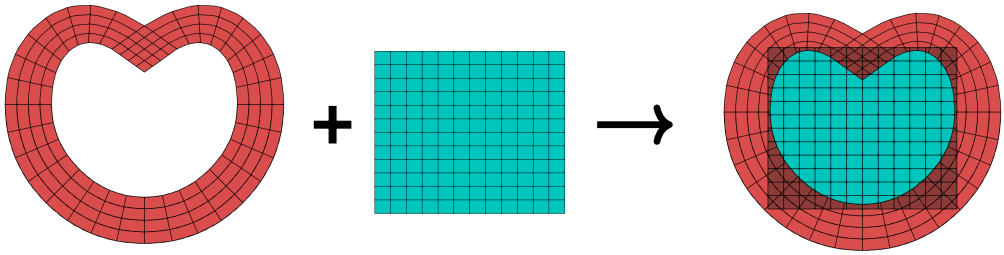


(b) Local solutions (left and middle) and all local solutions plotted together (right).

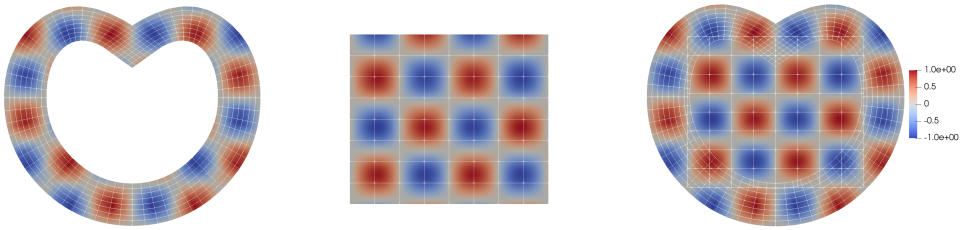


(c) L^2 (left) and H^1 (right) errors for $p = 3, 4$.

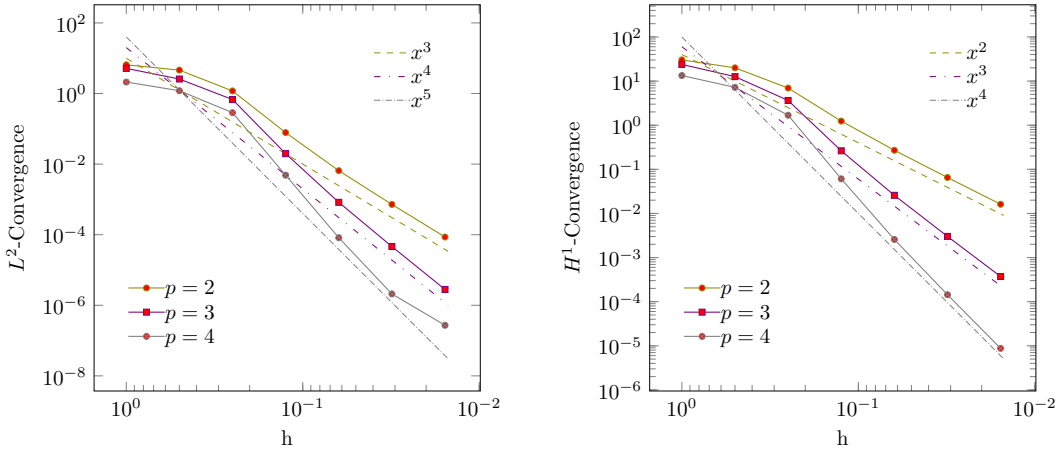
Figure 10: Numerical results for solving the Poisson problem on the domain with a random smooth boundary.



(a) Heart shaped domain.

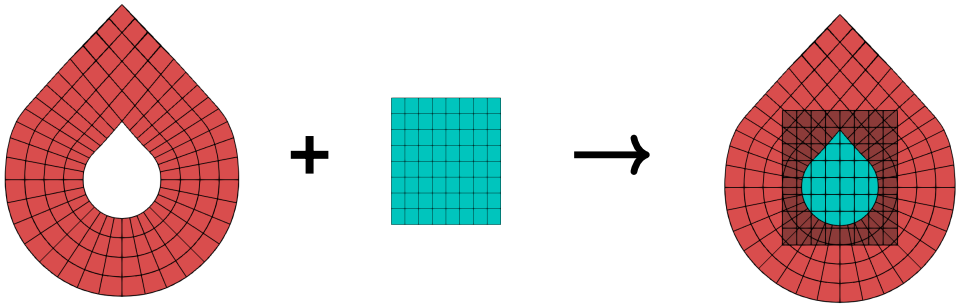


(b) Local solutions (left and middle) and all local solutions plotted together (right).

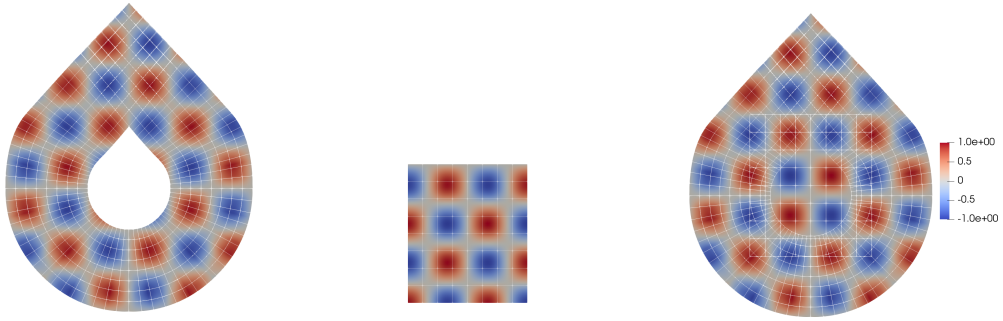


(c) L^2 (left) and H^1 (right) errors for $p = 2, 3, 4$.

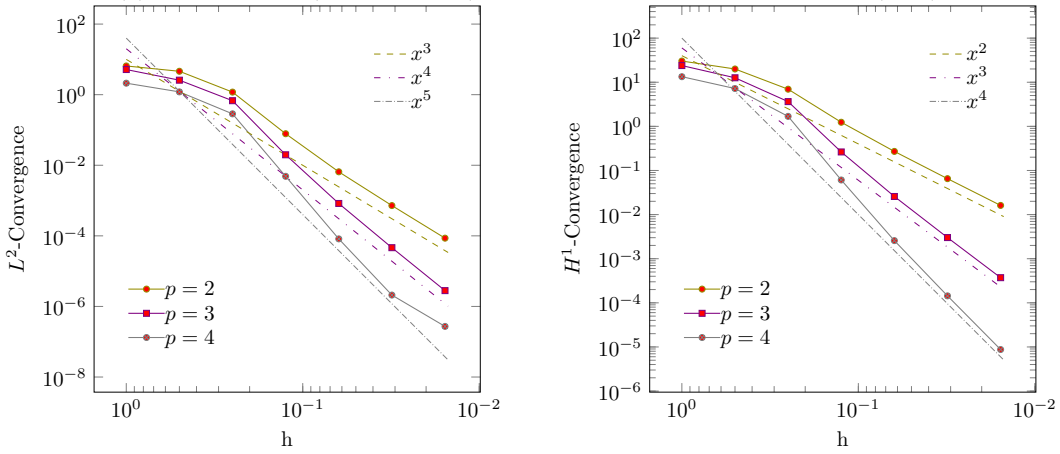
Figure 11: Numerical results for solving the Poisson problem on a heart shaped domain.



(a) Drop shaped domain.



(b) Local solutions (left and middle) and all local solutions plotted together (right).



(c) L^2 (left) and H^1 (right) errors for $p = 2, 3, 4$.

Figure 12: Numerical results for solving the Poisson problem on a drop shaped domain.

7. Conclusion

In this paper, we proposed an offset-based overlapping domain parameterization (OODP) method for IGA. In this method we generate an inner offset curve from a given regularly parameterized boundary curve by solving a regularized optimization problem. We penalize first and/or second derivatives to make the curve as smooth as possible. From this inner curve and the given outer curve we obtain a ring-shaped patch with a hole. This hole is then covered with a multi-cell domain.

We study the influence of the parameters in the proposed algorithm via several numerical examples. We show that using a quasi-normal vector, instead of the exact normal vector, for defining the inner offset curve in the algorithm generates a better parameterization for most of the considered cases. In addition, we apply the algorithm to boundary curves with convex and non-convex corners and we explain how the method can handle such cases.

Finally, we use the OMP method as introduced in [2] to solve PDEs on the proposed domain parameterizations. Note that in the numerical experiments we mostly cover the hole with simple rectangular subdomains. Such a covering is recommended since it allows for a more straightforward, more efficient implementation.

The OODP method can be applied to a wide range of given boundary curves in 2D. As we have demonstrated, it is always possible to cover complex domains with pairwise overlapping subdomains. However, in many cases, the parameterization strategy becomes more straightforward if we allow to have multiple overlaps (where three or more subdomains are overlapping). Therefore, for solving a PDE on such parameterizations, the OMP method, which is defined for pairwise overlaps, needs to be extended to configurations having multiple overlaps. We intend to extend the method in such a way in the future.

Even though we only consider B-spline parameterizations in the numerical examples, the proposed method is also applicable for NURBS boundary curves. In principle the OODP method can be generalized also to 3D domains. However, this extension is more complicated, as there are many possible configurations of boundary surfaces, which are in general given as trimmed B-spline or NURBS patches. Therefore, the parameterization of 3D domains needs further research.

8. Acknowledgments

The research was supported by the strategic program "Innovatives OÖ 2010 plus" by the Upper Austrian Government, by the FWF together with the Upper Austrian Government through the project P 30926-NBL, by Linz Institute of Technology and the government of Upper Austria through the project LIT-2019-8-SEE-116, by the Austrian Ministry for Transport, Innovation and Technology (BMVIT), the Federal Ministry for Digital and Economic Affairs (BMDW), and the

Province of Upper Austria in the frame of the COMET-Competence Centers for Excellent Technologies Program managed by Austrian Research Promotion Agency FFG, the COMET Module S3AI and by the "Austrian COMET-Programme" (Project InTribology, no. 872176).

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] S. Kargaran, B. Jüttler, S. Kleiss, A. Mantzaflaris, T. Takacs, Overlapping multi-patch structures in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 356 (2019) 325 – 353.

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

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

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

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

[7] A. Falini, B. Jüttler, THB-splines multi-patch parameterization for multiply-connected planar domains via template segmentation, Journal of Computational and Applied Mathematics 349 (2019) 390 – 402.

[8] G. Xu, M. Li, B. Mourrain, T. Rabczuk, J. Xu, S. P. Bordas, Constructing IGA-suitable planar parameterization from complex CAD boundary by domain partition and global/local optimization, Computer Methods in Applied Mechanics and Engineering 328 (2018) 175 – 200.

[9] J. Xu, F. Chen, J. Deng, Two-dimensional domain decomposition based on skeleton computation for parameterization and isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 284 (2015) 541 – 555.

[10] F. Massarwi, B. Sosin, G. Elber, Untrimming: Precise conversion of trimmed-surfaces to tensor-product surfaces, Computers and Graphics 70 (2017).

[11] K. Hui, Y.-B. Wu, Feature-based decomposition of trimmed surface, Computer-Aided Design 37 (2005) 859–867.

[12] S. Xiao, H. Kang, X.-M. Fu, F. Chen, Computing iga-suitable planar parameterizations by polysquare-enhanced domain partition, Computer Aided Geometric Design 62 (2018) 29 – 43.

[13] B. Jüttler, M. Kapl, D.-M. Nguyen, Q. Pan, M. Pauley, Isogeometric segmentation: The case of contractible solids without non-convex edges, Computer-Aided Design 57 (2014) 74 – 90.

[14] D.-M. Nguyen, M. Pauley, B. Jüttler, Isogeometric segmentation. part II: On the segmentability of contractible solids with non-convex edges, Graphical Models 76 (2014) 426 – 439.

[15] M. Pauley, D.-M. Nguyen, D. Mayer, J. Špeh, O. Weeger, B. Jüttler, The isogeometric segmentation pipeline, in: B. Jüttler, B. Simeon (Eds.), Isogeometric Analysis and Applications 2014, Springer International Publishing, 2015, pp. 51–72.

[16] D.-M. Nguyen, M. Pauley, B. Jüttler, Isogeometric segmentation: Construction of auxiliary curves, Computer-Aided Design 70 (2016) 89 – 99.

[17] M. Haberleitner, B. Jüttler, Isogeometric segmentation: Construction of cutting surfaces, Computer-Aided Design 90 (2017) 135 – 145.

[18] M. Haberleitner, B. Jüttler, Y. Masson, Isogeometric segmentation via midpoint subdivision suitable solids, Computer-Aided Design 114 (2019) 179 – 190.

[19] M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, A. V. Vuong, Swept Volume Parameterization for Isogeometric Analysis, Springer, 2009.

[20] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis, Computer-Aided Design 45 (2013) 812 – 821.

[21] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications, Computer-Aided Design 45 (2013) 395 – 404.

[22] M. Pan, F. Chen, W. Tong, Low-rank parameterization of planar domains for isogeometric analysis, Computer Aided Geometric Design 63 (2018) 1 – 16.

[23] S. Sajavičius, B. Jüttler, J. Špeh, Template Mapping Using Adaptive Splines and Optimization of the Parameterization, Springer International Publishing, 2019, pp. 217–238.

[24] J. Hinz, M. Möller, C. Vuik, An IGA framework for PDE-based planar parameterization on convex multipatch domains (2019). arXiv:1904.03009.

[25] X. Nian, F. Chen, Planar domain parameterization for isogeometric analysis based on teichmüller mapping, Computer Methods in Applied Mechanics and Engineering 311 (2016) 41 – 55.

[26] T. Martin, E. Cohen, R. M. Kirby, Mixed-element volume completion from NURBS surfaces, Computers and Graphics 36 (5) (2012) 548 – 554.

[27] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method, Journal of Computational Physics 252 (2013) 275 – 289.

[28] D.-M. Nguyen, A. Evgrafov, A. R. Gersborg, J. Gravesen, Isogeometric shape optimization of vibrating membranes, Computer Methods in Applied Mechanics and Engineering 200 (13) (2011) 1343 – 1353.

[29] J. Gravesen, A. Evgrafov, D.-M. Nguyen, P. Nørtoft, Planar parametrization in isogeometric analysis, in: Mathematical Methods for Curves and Surfaces, Springer Berlin Heidelberg, 2014, pp. 189–212.

[30] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parameterization of computational domain in isogeometric analysis: Methods and comparison, Computer Methods in Applied Mechanics and Engineering 200 (23) (2011) 2021 – 2031.

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

[32] C. Arioli, A. Shamanskiy, S. Klinkel, B. Simeon, Scaled boundary parametrizations in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 349 (2019) 576 – 594.

[33] 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.

[34] A. de Saint-Exupéry, The little prince, 1943.

[35] C. M. Grimm, J. F. Hughes, Modeling surfaces of arbitrary topology using manifolds, in: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, 1995, pp. 359–368.

[36] S. Kargaran, Overlapping multi-patch structures in isogeometric analysis, Ph.D. thesis, Johaness kepler university (2021).

Appendix A. Proof of Theorem 2

Proof. The parameterization $\tilde{\mathbf{F}}$ is given as in (4), where

$$\mathbf{C}_B = (C^1(t), C^2(t)) \text{ and } \quad \mathbf{q}(t) = (q^1(t), q^2(t)). \tag{A.1}$$

We have that $\tilde{\mathbf{F}}$ is regular, if the Jacobian determinant of the mapping is always negative. We obtain

$$\frac{\partial \tilde{\mathbf{F}}(s,t)}{\partial s} = \mu(t)\mathbf{q}(t) \text{ and } \quad \frac{\partial \tilde{\mathbf{F}}(s,t)}{\partial t} = \mathbf{C}'_B(t) + s(\mu'(t)\mathbf{q}(t) + \mu(t)\mathbf{q}'(t)), \tag{A.2}$$

Therefore, the Jacobian matrix of $\tilde{\mathbf{F}}(s,t)$ can be written as follows

$$J = \begin{pmatrix} \mu(t)\mathbf{q}(t) & \mathbf{C}'_B(t) + s(\mu'(t)\mathbf{q}(t) + \mu(t)\mathbf{q}'(t)) \end{pmatrix}. \tag{A.3}$$

We denote the determinant of J by $\det(J) = [J] = D$. In the following we obtain

$$D = \mu(t)[\mathbf{q}(t), \mathbf{C}'_B(t)] + (\mu(t)s\mu'(t)) \underbrace{[\mathbf{q}(t), \mathbf{q}(t)]}_{=0} + s\mu^2(t)[\mathbf{q}(t), \mathbf{q}'(t)]$$

which should be negative for all $s \in [0,1]^2$ and for all $t \in \mathbb{R}$, i.e.,

$$\mu(t)[\mathbf{q}(t), \mathbf{C}'_B(t)] + s\mu^2(t)[\mathbf{q}(t), \mathbf{q}'(t)] < 0. \tag{A.4}$$

Equation (A.4) is linear with respect to s. Hence, it suffices to satisfy the equation for $s = 0$ and $s = 1$. For $s = 0$ and we obtain

$$\mu(t)[\mathbf{q}(t), \mathbf{C}'_B(t)] < 0,$$

Since $[\mathbf{q}(t), \mathbf{C}'_B(t)]$ is negative, we need $\mu(t) > 0$. For $s = 1$ we get

$$\mu(t)[\mathbf{q}(t), \mathbf{C}'_B(t)] + \mu^2(t)[\mathbf{q}(t), \mathbf{q}'(t)] < 0,$$

or

$$\mu(t)[\mathbf{q}(t), \mathbf{q}'(t)] < -[\mathbf{q}(t), \mathbf{C}'_B(t)].$$

If

$$[\mathbf{q}(t), \mathbf{q}'(t)] > 0$$

we obtain

$$\mu(t) < \frac{[\mathbf{C}'_B(t), \mathbf{q}(t)]}{[\mathbf{q}(t), \mathbf{q}'(t)]}. \tag{A.5}$$

On the other hand, if

$$[\mathbf{q}(t), \mathbf{q}'(t)] \leq 0$$

we have

$$\mu(t)[\mathbf{q}(t), \mathbf{q}'(t)] \leq 0 < -[\mathbf{q}(t), \mathbf{C}'_B(t)]$$

which is satisfied for all t. This completes the proof. □