Planar Multi-Patch Domain Parameterization via Patch Adjacency Graphs

Florian Buchegger[a,*] and Bert Jüttler[b]

[a]MTU Aero Engines, Munich, Germany
[b]Institute of Applied Geometry, Johannes Kepler University, Linz, Austria

Abstract

As a remarkable difference to the existing CAD technology, where shapes are represented by their boundaries, FEM-based isogeometric analysis typically needs a parameterization of the interior of the domain. Due to the strong influence on the accuracy of the analysis, methods for constructing a good parameterization are fundamentally important. The flexibility of single patch representations is often insufficient, especially when more complex geometric shapes have to be represented. Using a multi-patch structure may help to overcome this challenge.

In this paper we present a systematic method for exploring the different possible parameterizations of a planar domain by collections of quadrilateral patches. Given a domain, which is represented by a certain number of boundary curves, our aim is to find the optimal multi-patch parameterization with respect to an objective function that captures the parameterization quality. The optimization considers both the location of the control points and the layout of the multi-patch structure. The latter information is captured by pre-computed catalogs of all available multi-patch topologies. Several numerical examples demonstrate the performance of the method.

Keywords: parameterization, multi-patch domains, isogeometric analysis

1. Introduction

The powerful framework of Isogeometric Analysis (IgA) unifies the descriptions of the geometric objects (the computational domains) and of the unknown quantities considered in numerical simulations by adopting multivariate spline functions for their representation [6]. This approach helps to deal with the problem of data exchange between the software tools used for the geometrical construction and design of CAD models and the tools used for analysis. Additionally, the higher smoothness of the spline functions has shown to be beneficial for the robustness and accuracy of the numerical simulations.

In the existing CAD technology, shapes are represented by boundary representation (B-rep) models, which contain information about the surfaces of geometric objects [5].

*Corresponding Author. Postal address: MTU Aero Engines, Dachauer Str. 665, 80995 Munich, Germany. Phone +49 89 1489 3771. Email Address: florian.buchegger@mtu.de

However, IgA typically needs a parameterization of the interior of the domain. Domain parameterization techniques, which generate domain parameterizations from boundary data, are therefore of a vital interest.

The starting point for exploring such methods is the case of a single patch, i.e., of a domain that is topologically equivalent to a d-dimensional cube of dimension $d = 2, 3$. Several techniques have been described in the literature.

These include discrete Coons patches [8], which are computationally inexpensive, or the spring model based approach [10]. A direct construction of spline parameterizations for swept volumes was established by [1]. A method for generating volumetric single-patch T-spline parameterizations of contractible objects is described in [31]. Optimal analysis-aware parameterizations were studied in [30].

For two-dimensional domains, a sequence of methods with varying levels of computational complexity has been presented in [7]: The first two methods make use of several functionals to place the inner control points, while the third method employs a harmonic mapping to guarantee regularity. Harmonic functions are used in [21] and [13] also to construct parameterizations of generalized cylinders and of contractible domains, respectively. Another method, that is specifically tailored towards single-patch NURBS-parameterization, has been developed in [23] using sequences of harmonic mappings.

When dealing with more complex geometric shapes, however, single patch representations do not provide sufficient flexibility. The use of multi-patch structures, which are obtained by gluing patches together, is a standard approach to make them more versatile.

The coupling of the patches can be performed in several ways. One may identify degrees of freedom along the interfaces. This has led to special constructions for splines on multi-patch domains, such as spline forests [27] or multi-patch B-splines with enhanced smoothness [3]. Alternatively, one may use continuity constraints and enforce them via Lagrangian multipliers. This was found to be useful for designing isogeometric solvers that exploit the power of parallel computing [14, 16]. Finally, there exist numerous techniques such as use of mortar methods, Nitsche's method, and the discontinuous Galerkin method [2, 12, 19].

The construction of multi-patch parameterizations in IgA is less well understood. For a fixed topology, an optimization-based technique has been investigated in [29]. Skeleton-based polycube constructions have been used to generate T-spline parameterizations of multi-patch-type [20]. An isogeometric segmentation pipeline, which combines domain splitting with single patch parameterization techniques, was established in [15, 22, 24]. It should be noted that there are strong relations to the problem of block-structured hexahedral mesh-generation, see [18] and the references therein. Recently, a systematic method for exploring the possible quad mesh topologies has been presented in [25].

In our present paper, which focuses on the two-dimensional case, we establish a systematic method for exploring the different possible segmentations of a domain into quadrilateral patches. Given a domain with a certain number of boundary curves, which we denote as segments, our aim is to find the optimal multi-patch parameterization with respect to an objective function that captures the parameterization quality. The optimization considers both the location of the control points and the layout of the multi-patch structure. This is achieved by using catalogs of all available multi-patch topologies.

This paper is organized as follows. First we recall the concept of a multi-patch parameterization and introduce the patch adjacency graph in Section 2. The outline of

the algorithm as well as the objective functions used for the geometry optimization are presented in the next section. Finally we demonstrated the performance of the proposed method by several computational examples in Section 4. The method used for enumerating the multi-patch topologies is reported in the appendix.

2. Preliminaries

We introduce the notion of a multi-patch parameterization and show how to capture its topology by a patch adjacency graph.

2.1. Valid multi-patch parameterizations

We consider a simply connected planar domain Ω with piecewise smooth boundary. More precisely, the boundary is represented by b parametric curves,

$$c_i : [0,1] \to \mathbb{R}^2, \quad i = 1, \ldots, b, \tag{1}$$

which are called *segments*. Note that the indices of these curves are always considered modulo b. The segments do not intersect except for the end points of neighboring segments,

$$\begin{aligned} &\forall i : c_i(1) = c_{i+1}(0) \quad \text{and} \\ &\forall i, j, \forall s, t \in [0,1) \colon c_i(s) = c_j(t) \Rightarrow (i = j \wedge s = t). \end{aligned}$$

The first property ensures that the segments form a closed loop. Without loss of generality, we assume that the segments are oriented clockwise.

Some pairs of adjacent segments may meet with tangent (G^1) continuity, i.e.,

$$\exists \lambda > 0 : \dot{c}_i(1) = \lambda \dot{c}_{i+1}(0).$$

Adjacent edges meet in *corners*, which are either *convex* if the inner angle is smaller than π or *non-convex* otherwise. The latter notion also includes the case of tangent continuity between adjacent segments.

We want to represent the domain, i.e., the interior of the loop formed by the segments, by a *multi-patch parameterization* (MPP). More precisely, the parameterization consists of p geometry patches $\boldsymbol{g}^{(j)}$.

$$\boldsymbol{g}^{(j)} \colon [0,1]^2 \to \mathbb{R}^2 \quad j = 1, \ldots, p.$$

A patch is *regular* if the determinant of the Jacobian $\nabla \boldsymbol{g}^{(j)}$ is positive. In particular, this implies that the corner angles do not exceed π. Every patch $\boldsymbol{g}^{(j)}$ has a loop of four boundary curves

$$\begin{aligned} g_1^{(i)} &\colon [0,1] \to \mathbb{R}^2 \colon t \mapsto \boldsymbol{g}^{(i)}(0,t) \\ g_2^{(i)} &\colon [0,1] \to \mathbb{R}^2 \colon t \mapsto \boldsymbol{g}^{(i)}(t,0) \\ g_3^{(i)} &\colon [0,1] \to \mathbb{R}^2 \colon t \mapsto \boldsymbol{g}^{(i)}(0,1-t) \\ g_4^{(i)} &\colon [0,1] \to \mathbb{R}^2 \colon t \mapsto \boldsymbol{g}^{(i)}(1-t,0), \end{aligned}$$

with counterclockwise orientation, see Fig. 1.

We use the opposite (i.e., clockwise) orientation for the boundary of the domain in order to prepare the definition of the adjacency relation (see below), which identifies pairs of curves with opposite orientation.

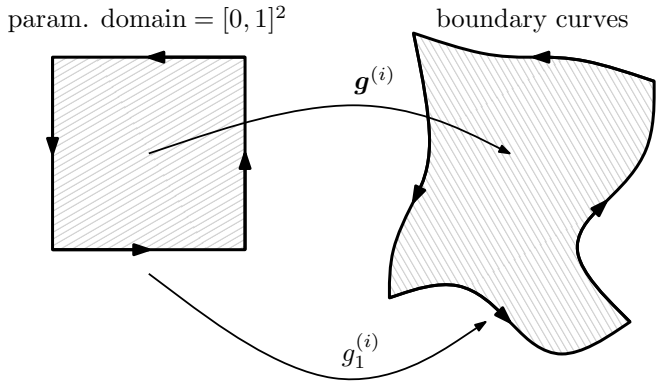


Figure 1: Parameter domain and loop of boundary curves for a patch i

All boundary curves and the segments forming the domain boundary are collected into a set

$$S = \{g_\ell^{(i)} \mid \ell = 1, \ldots, 4;\ i = 1, \ldots, p\} \cup \{c_i \mid i = 0, \ldots, b-1\}.$$

The interface relation $\sim$ on S identifies curves that possess the same parameterization in reverse orientation,

$$s \sim s' \Leftrightarrow s(t) = s'(1-t), \quad \forall t \in [0,1] \quad s, s' \in S.$$

It captures both the adjacency relation between patches and between a patch and the domain boundary. This relation is symmetric but neither reflexive nor transitive, and it does not relate pairs of segments c_i and c_j.

A set of patches $\{\boldsymbol{g}^{(j)}\}$ is called a *valid* multi-patch parameterization if it satisfies the following three conditions:

(P) All patches are regular and their interiors are mutually disjoint.

(R1) Every curve in S, which can be either a segment (on the domain boundary) or a patch boundary curve, is related to exactly one other curve in S.

(R2) There are no relations between the boundary curves of each patch.

The condition (R1) implies that the relation $\sim$ identifies pairs of curves of S for each valid MPP. Therefore S has to contain an even number of curves. Indeed, the total number of boundary curves $g_\ell^{(i)}$ of all patches is even, hence the number of segments on the domain boundary has to be even too.

2.2. The patch adjacency graph (PAG)

We introduce patch adjacency graph (PAG), also known as "dual layout", see [4]. It can be obtained from any MPP $\mathcal{M}$ and captures the information provided by the adjacency relation $\sim$. The PAG is a planar graph and its vertices are split into two groups:

- Every patch is represented by an *inner vertex*.
- Every segment of the boundary introduces a *boundary vertex*.

The total number of vertices $v = b + p$ is the sum of the number b of the boundary curves and of the number p of patches. We introduce a global numbering of the vertices:

- The boundary vertices have numbers $1, \ldots, b$, which are inherited from the associated boundary segments and ordered clockwise.
- The inner vertices have the numbers $b+1, \ldots, b+p$.

The edges of the PAG can be grouped into two categories:

- Each *inner edge* corresponds to a pair of curves of S identified by $\sim$. Recall that each inner vertex of the PAG is associated with four boundary curves, while each boundary vertex is associated with a single curve (a segment on the domain boundary). Two vertices of the PAG are connected by an inner edge, if one of the curves associated with the first one is related to one of the curves associated with the second one.
- Pairs of adjacent boundary segments define *boundary edges*.

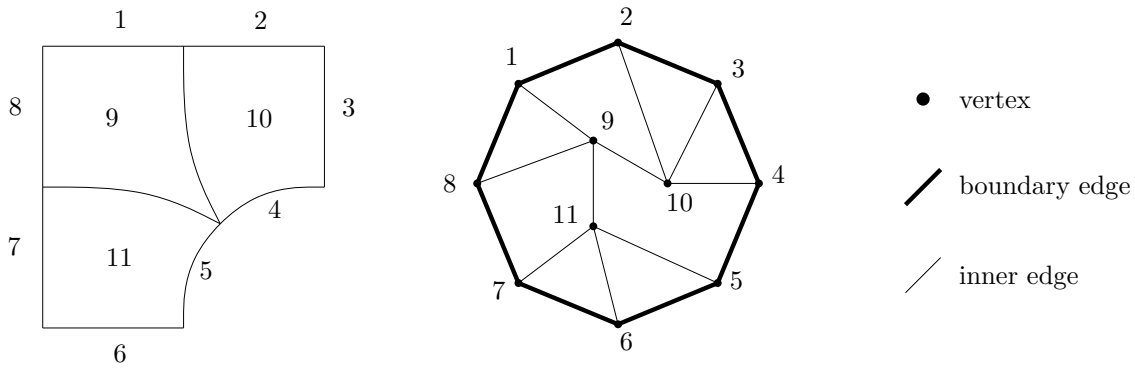


Figure 2: Multi-patch domain with associated PAG

The patch adjacency graph of a valid MPP is a planar graph that possesses the following three properties:

(G1) The inner vertices of G have valence four, while the boundary vertices possess valence three.

(G2) The subgraph which is formed by the boundary vertices is a circle. Therefore, two of the neighbors of each boundary vertex are boundary vertices, while the remaining one is an inner vertex.

(G3) The subgraph which is formed by the inner vertices is connected and does not contain any biangles or loops[1] (i.e., faces bounded by one or two edges of the graph).

The planarity of the PAG follows from the fact that the patches of $\mathcal{M}$ parameterize the physical domain without overlaps. The valency of the inner vertices is determined by the number of boundary curves of each patch. The valency of the boundary vertices follows from the construction. The connectivity of the subgraph containing the inner vertices (G3) is implied by the fact that the domain is connected. The regularity of the patches entails the second part of (G3).

[1] A loop corresponds to a patch with identified opposite boundaries since we consider regular parameterizations only. The region in its interior cannot be filled with regular patches.

It should be noted that PAGs with multiple edges exist, see Fig. 3. Later we will introduce a boundary compatibility condition, which is required at non-convex vertices of the domain.

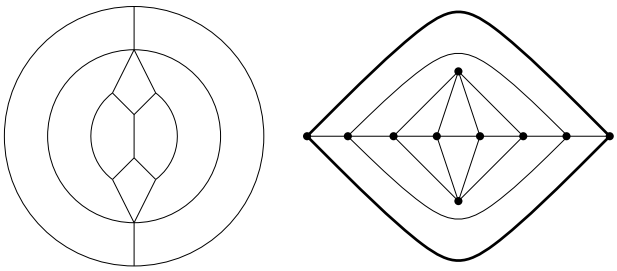


Figure 3: MPP with associated PAG with two edges between two vertices

Each PAG is represented by five functions, which we denote by I, E, N, W, and S. The first function

$$I : \{1, \ldots, b\} \to \{b+1, \ldots, b+p\}$$

assigns the unique inner vertex to each boundary vertex, cf. (G2), while the remaining four functions

$$E, N, W, S : \{b+1, \ldots, b+p\} \to \{1, \ldots, b+p\}.$$

specify the four neighbors (East, North, West, South) of each inner vertex.

For certain ranges of b and p we created catalogs of PAGs satisfying (G1-G3). This is described in more detail in the appendix.

3. Parameterization algorithm

Given a collection of boundary segments, a PAG is said to be *valid* if adjacent boundary vertices that correspond to edges enclosing a non-convex vertex are adjacent to different inner vertices,

$$\angle(\dot{c}_j(1), \dot{c}_{j+1}(0)) \geq 0 \quad \Rightarrow \quad I(j) \neq I(j+1). \tag{2}$$

The validity is required to obtain regular parameterizations in the vicinity of non-convex boundary vertices, see Fig. 4.

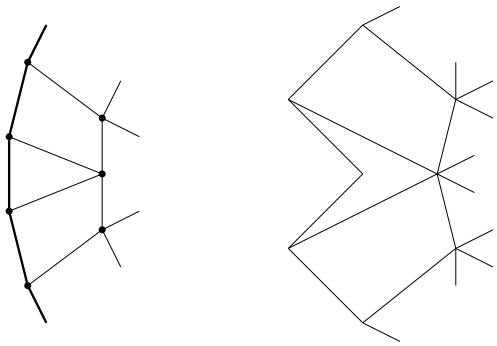


Figure 4: Violation of the boundary compatibility condition (left: PAG, right: associated patch layout) leads to a singular parameterization.

We consider all valid PAGs for the given value of b and for all $p \leq p_{\max}$, where the maximum number of patches $p_{\max}$ is chosen by the user. For each valid PAG we compute a candidate parameterization in three steps:

1. Build the control point structure of the MPP defined by the PAG.
2. Find initial values of the inner control points by minimizing the objective function (5). These values are computed by solving a linear system.
3. Find optimal values of control points by minimizing the objective function (6) via nonlinear optimization, using the previously computed result as initial solution.

Among all results we then select the one which is regular and provides the smallest value of the non-linear objective function (6). Notice that the two objective functions (5) and (6) will be defined later in Sections 3.2 and 3.3, respectively.

3.1. Building the MPP control point structure

For simplicity we choose all patches to have same knot vectors and degrees and therefore the control points can be denoted as

$$d_{i,j}^k, \quad i,j = 0,\ldots,n, \quad k = b+1,\ldots,b+p.$$

We further assume that all segments are spline curves, which have the same degree and the same knot vectors as the patches. If this is not the case, we approximate the boundary of the domain by appropriate curves, in order to fulfill this requirement. Their control points are denoted as

$$d_{i,0}^k, \quad i = 0,\ldots,n, \quad k = 1,\ldots,b, \tag{3}$$

where the second lower index is introduced in order to simplify the notation.

To keep preserve the domain boundary, fixed values are assigned to the boundary control points of the adjacent patches,

$$\begin{aligned} d_{n,n-i}^{I(k)} &= d_{i,0}^k \text{ if } E(I(k)) = k, \\ d_{i,n}^{I(k)} &= d_{i,0}^k \text{ if } N(I(k)) = k, \\ d_{0,i}^{I(k)} &= d_{i,0}^k \text{ if } W(I(k)) = k, \\ d_{n-i,0}^{I(k)} &= d_{i,0}^k \text{ if } S(I(k)) = k. \end{aligned}$$

Furthermore, we identify interface control points of adjacent patches. This leads to the equation

$$d_{n,i}^k = d_{n-i,0}^{k'} \text{ if } k' = E(k) \text{ and } k = S(k'). \tag{4}$$

and 15 similar ones, which cover the 16 different cases of patch-patch contact.

3.2. Constructing the initial solution

We collect the coordinates of control points of all patches in a vector $\mathbf{d}$. Some of these coordinates are determined by the conditions (3). After omitting them we obtain the sub-vector $\mathbf{d}^*$, which contains the unknowns of the optimization problem. In addition, we denote with $\mathbf{d}^k$ the coordinates of the control points of patch no. k.

We use a simple linear direct method (see [7]) to obtain an initial solution. More precisely, we solve the optimization problem

$$\sum_{k=b+1}^{b+p} \lambda_\ell \mathcal{Q}_\ell(\mathbf{d}^k) + \lambda_u \mathcal{Q}_u(\mathbf{d}^k) \to \min_{\mathbf{d}^*} \tag{5}$$

which is obtained by combining the parametric length functional

$$\mathcal{Q}_\ell(\mathbf{d}^k) = \int_{\mathbf{\Omega}} \|\boldsymbol{g}_u^{(k)}\|^2 + \|\boldsymbol{g}_v^{(k)}\|^2 \mathrm{d}u\,\mathrm{d}v$$

with the uniformity functional

$$\mathcal{Q}_u(\mathbf{d}^k) = \int_{\mathbf{\Omega}} \|\boldsymbol{g}_{uu}^{(k)}\|^2 + 2\|\boldsymbol{g}_{uv}^{(k)}\|^2 + \|\boldsymbol{g}_{vv}^{(k)}\|^2 \mathrm{d}u\,\mathrm{d}v$$

using user-defined non-negative weights λ_ℓ and λ_u. The choice of the weights will be discussed in Section 4.2.

In addition, the solution has to satisfy the adjacency conditions (4), which are simply enforced by identifying control points. (One might use Lagrangian multipliers instead, but this would produce a bigger system with a matrix that is not guaranteed to be positive definite anymore). Due to the simple nature of these functionals, the solution of the optimization problem is found easily by solving a linear system using a direct solver.

We obtain an initial parameterization, which is, however, not always satisfactory.

3.3. Non-linear optimization

In order to improve the quality of the MPP we consider the more general objective function

$$\sum_{k=b+1}^{b+p} \lambda_\ell \mathcal{Q}_\ell(\mathbf{d}^k) + \lambda_u \mathcal{Q}_u(\mathbf{d}^k) + \lambda_o \mathcal{Q}_o(\mathbf{d}^k) + \lambda_s \mathcal{Q}_s(\mathbf{d}^k) + \lambda_a \mathcal{Q}_a(\mathbf{d}^k) \to \min_{\mathbf{d}^*} \tag{6}$$

which is obtained by additionally considering the orthogonality functional

$$\mathcal{Q}_o(\mathbf{d}^k) = \int_{\mathbf{\Omega}} (\boldsymbol{g}_u^{(k)} \cdot \boldsymbol{g}_v^{(k)})^2 \mathrm{d}u\,\mathrm{d}v,$$

the skewness functional

$$\mathcal{Q}_s(\mathbf{d}^k) = \int_{\mathbf{\Omega}} \left(\frac{(\boldsymbol{g}_u^{(k)} \cdot \boldsymbol{g}_v^{(k)})^2}{(\boldsymbol{g}_u^{(k)} \cdot \boldsymbol{g}_u^{(k)})(\boldsymbol{g}_v^{(k)} \cdot \boldsymbol{g}_v^{(k)})} \right)^2 \mathrm{d}u\,\mathrm{d}v,$$

and the area functional

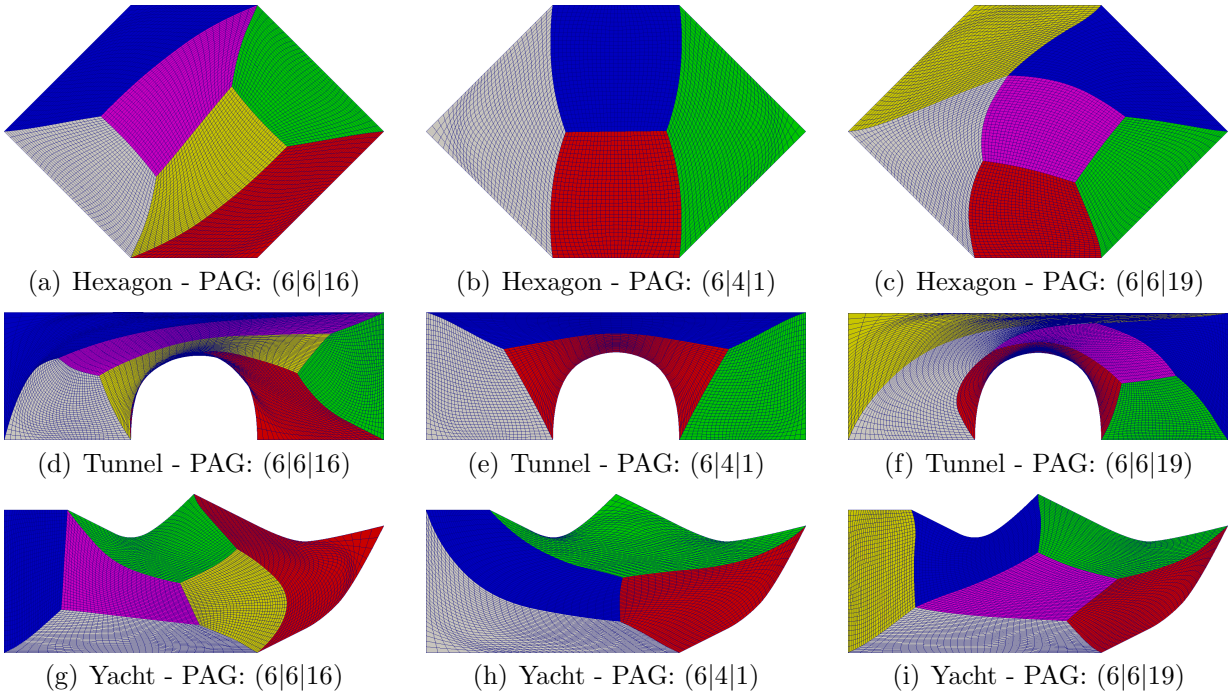
$$\mathcal{Q}_a(\mathbf{d}^k) = \int_{\mathbf{\Omega}} \det(\boldsymbol{g}_u^{(k)}, \boldsymbol{g}_v^{(k)})^2 \mathrm{d}u\,\mathrm{d}v.$$

The first two measures were already used in [7]. In addition we consider the third functional, which measures the squared determinant of the geometry mapping. This penalizes fold-overs and encourages a uniform size of the elements.

Similar to [7] we apply an iterative Gauss-Newton method to find the solution of the optimization problem, which is defined by the objective function (6) and the constraints (4). The result of the simpler problem (5) is used to initialize the iteration process.

4. Computational results

We demonstrate the performance of our method by several computational examples.



(a) Hexagon - PAG: (6|6|16) (b) Hexagon - PAG: (6|4|1) (c) Hexagon - PAG: (6|6|19)

(d) Tunnel - PAG: (6|6|16) (e) Tunnel - PAG: (6|4|1) (f) Tunnel - PAG: (6|6|19)

(g) Yacht - PAG: (6|6|16) (h) Yacht - PAG: (6|4|1) (i) Yacht - PAG: (6|6|19)

Figure 5: Multi-patch parameterizations created from different PAGs

4.1. Influence of the chosen PAG

First we explore the influence of the selected PAG to the properties of the final multi-patch parameterization. Figure 5 shows three examples and the resulting MPPs for three different choices of the PAGs. The geometries of the examples represent a simple hexagon shape (first row), a tunnel cross section (second row) and the profile of a yacht (third row). For all three geometries, the boundary is defined by six segments, and all segments meet in convex vertices only. Thus, we can consider the full range of PAGs in all these examples.

Each segment is given by a quadratic B-spline curve with 5 control points and an uniform knot-vector. Consequently, we use spline patches with 25 control points to parameterize the domain. All shapes are of similar size and are contained in the bounding box $[0, 12]^2$. We are interested in a solution with a maximal number of 6 patches, therefore 34 PAGs are considered. For the non-linear optimization we choose the weights $\lambda_o = 0.1$, $\lambda_u = 0.1$ and $\lambda_a = 2$. The remaining weights are set to 0. For each geometry we pick the optimal result and compare it to the result of the other shapes with the same PAG. The results are summarized in Table 1. The PAGs are identified by a label of the form

(number of segments b|number of patches p|PAG id).

4.2. Further examples

Table 2 and Figures 6–10 present further examples. Details of the Flow Passage example show the behavior of the parameterization near the leading edge (Fig. 8(c)), the trailing edge (Fig. 8(d)) and the extraordinary vertex in the inside of the domain (Fig. 8(e)).

For all these examples, we report the number of segments on the boundary b, the maximum number of patches considered p_{max}, the number of control points (cps) used for each patch, the number of valid PAGs, the id of the best PAG, the value of the objective

name	criterion	PAG number (segments|patches|id)		
		(6|6|16)	(6|4|1)	(6|6|19)
Hexagon	optimal solution ?	**yes**	no	no
	value of the objective function (6)	1463.8	2159.4	1470.8
	regularity	yes	yes	yes
Tunnel	optimal solution ?	no	**yes**	no
	value of the objective function (6)	1073.2	886.2	1016.5
	regularity	no	yes	no
Yacht	optimal solution ?	no	no	**yes**
	value of the objective function (6)	825.6	1192.3	703.6
	regularity	yes	no	yes

Table 1: Comparison of the examples in Fig. 5

function and the total computing time (excluding the generation of the list of PAGs) in the Table. The computations were done on a 64-Bit Debian Linux laptop with an Intel i7-4500U quad core CPU with 1.80 GHz and 16 GB of memory.

We scaled all examples to make them fit into the box $[0, 12] \times [0, 12]$ and used the weights $\lambda_o = 0.5$, $\lambda_u = 0.1$ and $\lambda_a = 2$ in the Jet, Hammer, Car and Puzzle Piece 1 examples. For the Muffin example the weights were chosen as $\lambda_a = 5$ and $\lambda_\ell = 1$. The Puzzle Piece 2 was computed using $\lambda_\ell = \lambda_s = 0.05$ and $\lambda_a = 5$ and the Flow-Passage used $\lambda_o = \lambda_u = \lambda_s = 1$ and $\lambda_a = 5$. All remaining weights were set to zero.

For the Jet example, we investigated the influence of the starting solution, reporting two different initial solutions corresponding to different weights in the objective function (5) in Fig. 9. We chose $\lambda_\ell = 1$ and $\lambda_u = 0$ for the left figure, while we used $\lambda_\ell = \lambda_u = 1$ for the right one. However, both starting points for the non-linear optimization led to the same final result. A similar behavior was observed in other examples.

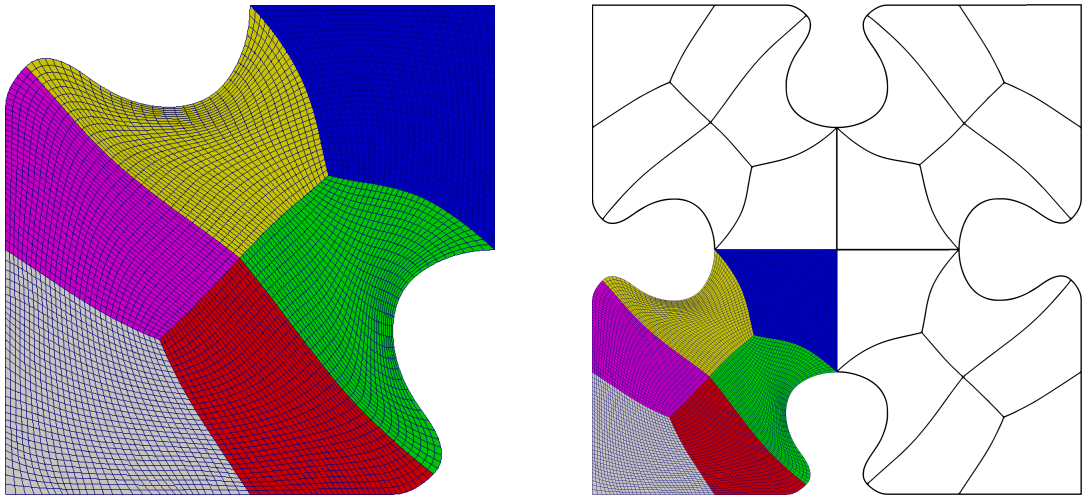
In our experiments, the choice of the weights for the non-linear optimization function (6) seems to be quite robust for domains that are sufficiently simple. For more complicated examples, we used a catalog of combinations of weights to select suitable values. This approach, however, leads to a larger computation time as the number of optimization problems is increased.

If self-intersections are likely to appear, then using the area functional with a relatively large weight helps significantly to obtain regular solutions. See [7] for a more detailed discussion of the influence of the remaining weights. It should be noted that all functionals encourage the use of several smaller patches instead of a few larger ones, since splitting a patch decreases the (total) value of the objective function. This effect, however, is limited by the fact that the segments on the boundary cannot be subdivided during the optimization. One might compensate it by using more complicated weights, but this was not found to be necessary in our examples.

It should be noted that a large number of PAGs had to be considered for the Flow Passage example, thereby increasing the computation time. This is due to the fact that this shape has more convex vertices on the boundary, hence more PAGs are valid.

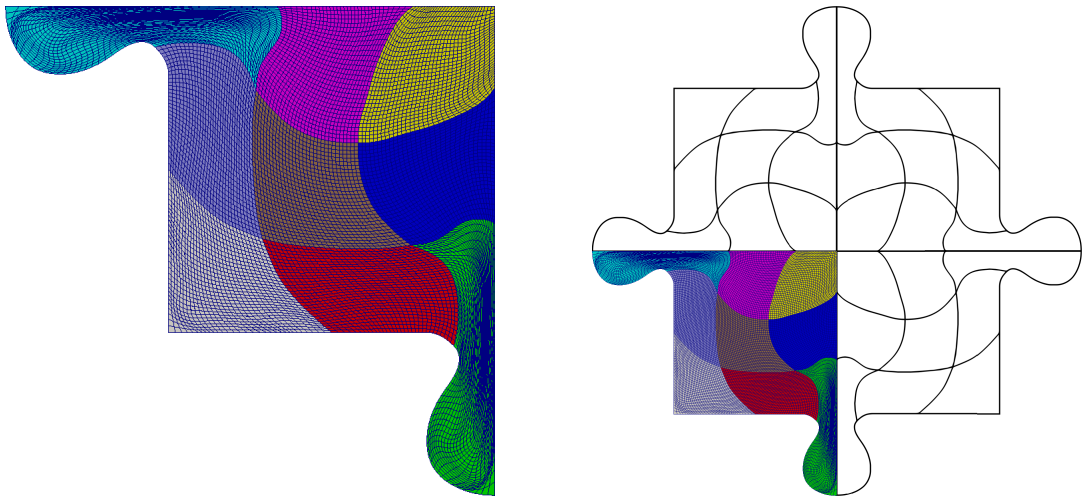
name	b	p_{max}	# cps per patch	# valid PAGs	best PAG	value of (6)	time (in sec)
Jet	6	7	25	100	(6|7|20)	704.1	54.9
Hammer	12	7	25	38	(12|7|488)	1465.3	22.0
Muffin	6	8	25	14	(6|8|157)	5467.5	8.3
Car	12	8	25	110	(12|8|4778)	329.3	71.8
First Puzzle Piece	8	6	36	8	(8|6|25)	5323.7	6.4
Second Puzzle Piece	12	9	36	45	(12|9|8829)	3127.5	65.6
Flow Passage	12	9	16	2903	(12|9|7354)	1776.1	1227.4

Table 2: Further examples



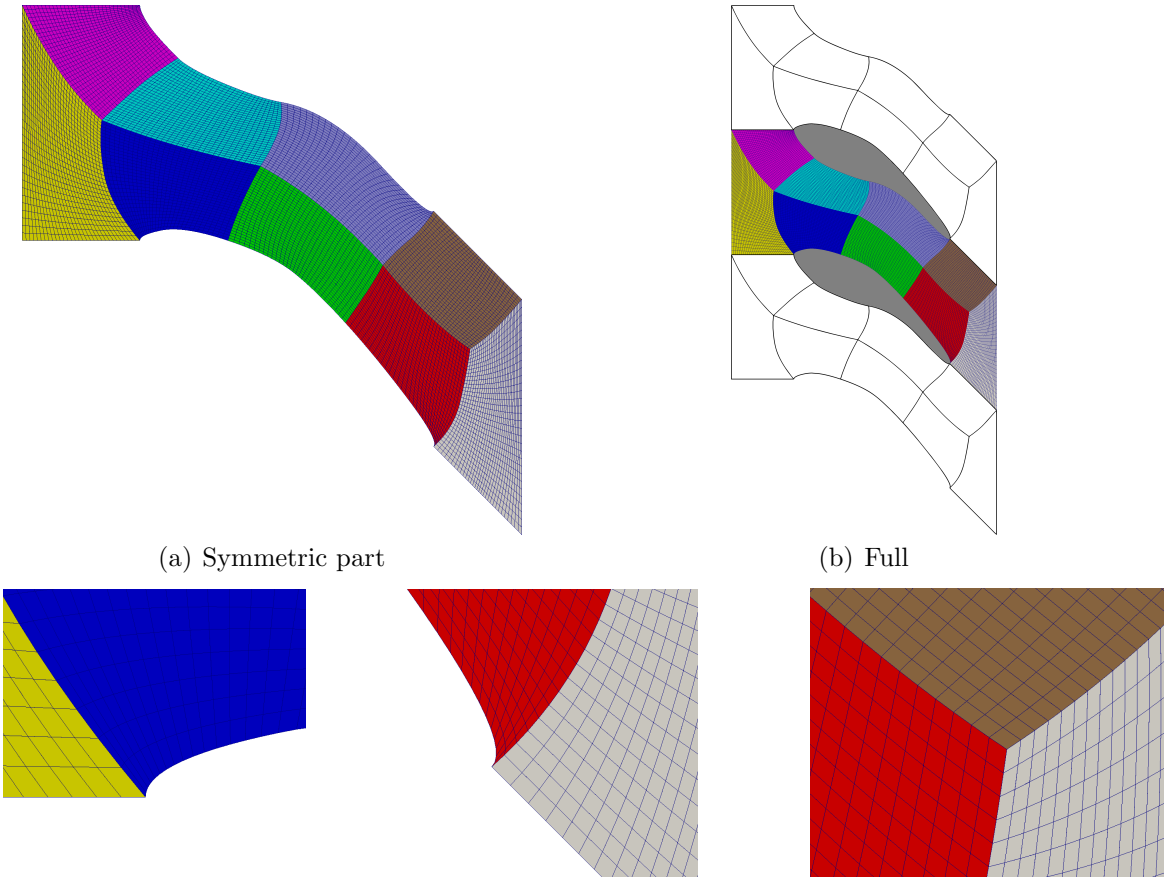
(a) Symmetric part (b) Full

Figure 6: First puzzle piece



(a) Symmetric part (b) Full

Figure 7: Second puzzle piece



(a) Symmetric part (b) Full

(c) Detail - Leading edge (d) Detail - Trailing edge (e) Detail - EV

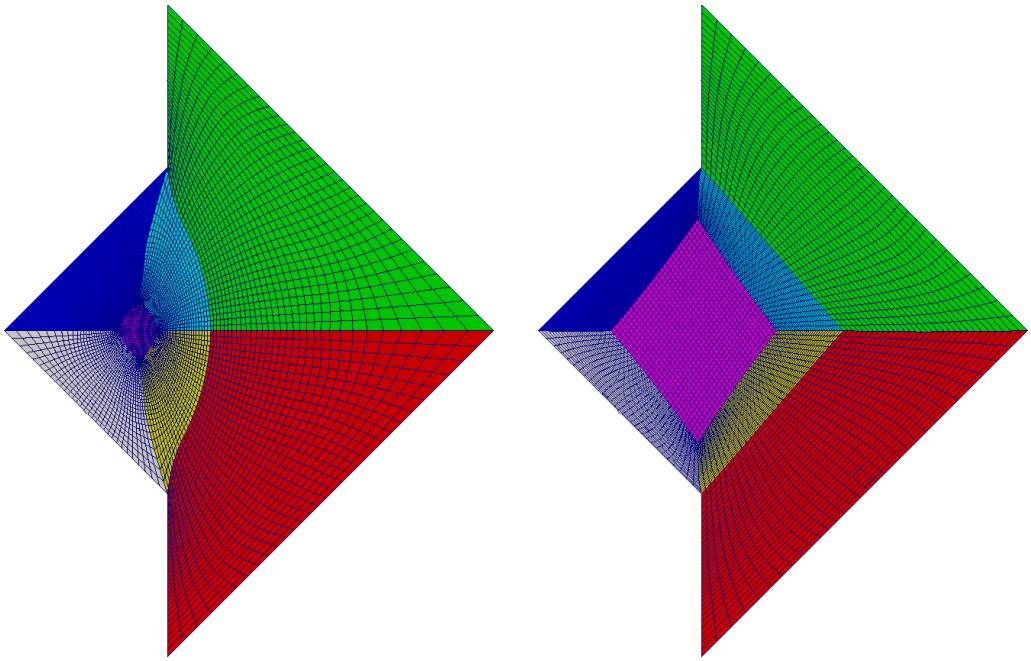
Figure 8: Flow-Passage of a compressor blade

5. Conclusion

We presented a method for parameterizing a planar shape which is represented by a number of boundary segments. Our algorithm identifies the optimal multi-patch parameterization with respect to a quality measure given by an objective function. All different possible segmentations of the domain into quadrilateral patches are systematically explored and a Gauss-Newton method is used to find the optimal placement of the inner control points for each segmentation. The multi-patch topologies, which are suitable for the given boundary segments are taken from a catalog. Selecting the configuration with the lowest value of the objective function gives the desired parameterization.

The method works well for examples with a modest number of patches and boundary segments. It is clear that it will not be suitable for larger number of patches, simply since the size of the catalog grows too fast. Nevertheless, we feel that the method can be valuable when dealing with certain classes of engineering shapes, such as the flow passage of a compressor blade. As an advantage, our method is guaranteed to find the optimal parameterization (in the sense that it provides the minimal value of the objective function) among all possible multi-patch topologies with a certain number of patches.

The functionals presented in this paper work well but do not guarantee a regular solution. This issue can be solved by using more complicated functionals ([10, 28]). We did not consider these functionals in our work since they require more sophisticated optimization



(a) $\lambda_\ell = 1, \lambda_u = 0$ (b) $\lambda_\ell = 1, \lambda_u = 1$

Figure 9: Two different initial solutions of the Jet example

techniques.

Future work will focus on the extension to the 3D case and – closely related to it – on more advanced methods for creating and representing the catalogs of available multi-patch topologies. In the 3D case, the patch layout cannot be captured by a planar graph, requiring a more complicated structure instead. Moreover the enumeration of this more complicated structure is not yet understood. The quality functionals can be generalized to the 3D case [11]. Also, we plan to identify the essential parts of these catalogs, which are determined by the distribution of the extraordinary vertices.

Another interesting topic is the development of heuristic algorithms that only generate a relevant subset of the catalog, thus decreasing the number of multi-patch layouts and therefore the computing time of the method.

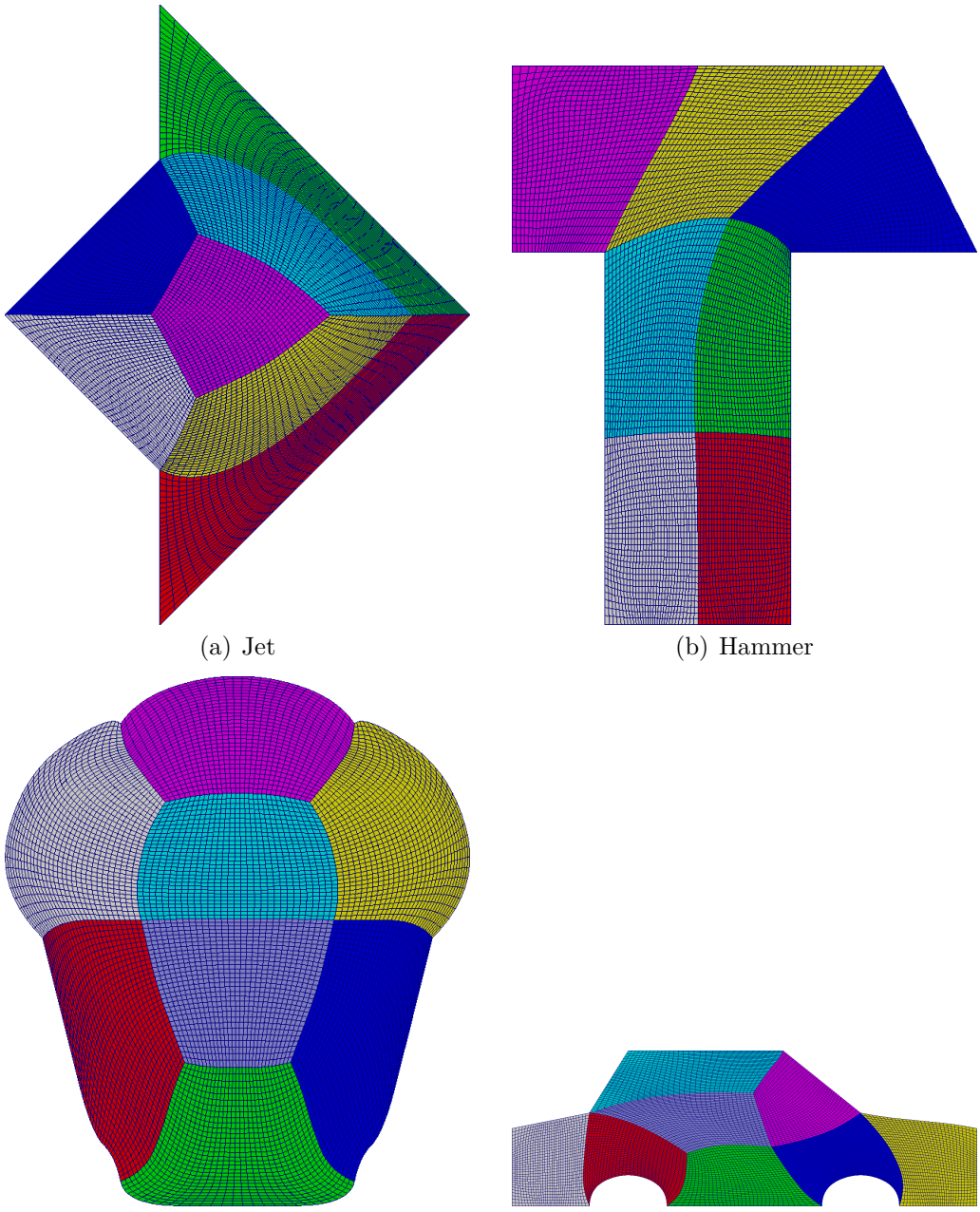
Additionally, we will explore applications to isogeometric mesh generation in the context of aircraft engine design.

Appendix A. Catalog of PAGs

We present a simple algorithm to construct all PAGs $(G_j), j \in \mathfrak{J}$, which satisfy the properties (G1) to (G3) for a given number of boundary segments. Starting from the given boundary segments c_i, the boundary vertices and edges of G can be constructed easily. The remaining challenge is to complete the interior of the graph according to the properties (G1) to (G3).

Appendix A.1. Existence of valid PAGs

First we explore the existence question for valid PAGs. Recall that a PAG is valid if it satisfies the condition formulated in Eq. (2). We already observed in Section 2.1 that



(a) Jet (b) Hammer

(c) Muffin (d) Car

Figure 10: Several results of the optimization process

an even number of boundary segments is a necessary condition for the existence of a valid multi-patch parameterization. This condition is also sufficient:

Lemma 1. *A valid patch graph G^v exists iff the number of given boundary segments b is even.*

Proof. For $b = 2$ and $b = 4$ the graphs can be given explicitly in Fig. A.11. If $b > 4$, a construction is used to achieve a patch graph. All b boundary vertices are placed on a circle. A smaller copy of this circle with its vertices is moved to a concentric position, see Fig. A.12. All boundary vertices are connected to the replicated vertices on the smaller circle. Another smaller, concentric circle is added. $\frac{b}{2}$ inner vertices are positioned on this circle in way, that always two vertices on the middle circle can be connected to them in a planar way. Finally we note that the resulting PAGs are always valid, since no pair of adjacent boundary vertices are connected to the same inner vertices. □

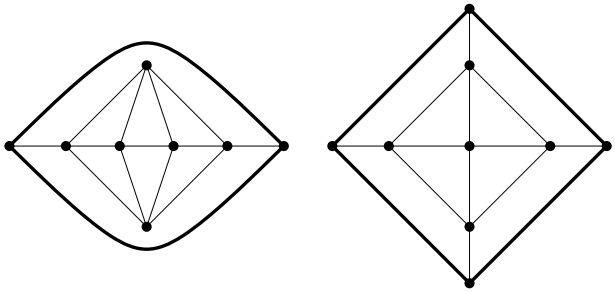


Figure A.11: PAGs for $b = 2$ and $b = 4$

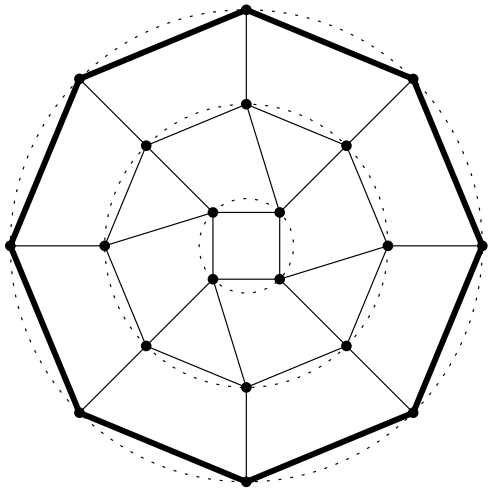


Figure A.12: Construction used in lemma 1 for 10 boundary segments

It should be noted that this construction produces a PAG with $b + \frac{b}{2}$ inner vertices. However, valid PAGs with a lower number of inner vertices may exist, depending the number of non-convex boundary vertices.

Appendix A.2. Enumeration of PAGs

The enumeration of planar graphs with certain properties is a well-studied problem in combinatorics and graph theory, see [9] and the references therein.

The number of graphs grows very fast with the number of vertices: The asymptotic behavior of the number of connected planar graphs with n vertices is given in [9] as

$$an^{-7/2}b^n n!$$

with certain constants a and b.

Our simple enumeration algorithm is based on a list of complete ternary trees, which can be generated easily [17].

First we initialize the circle of b boundary vertices and create one leaf connected to each boundary vertex except for the last one. These leaves are numbered by 1 to $b-1$ clockwise. We now attach a ternary tree with p nodes to the last boundary vertex and number the leaves of the tree by 1 to $2p+1$ counterclockwise, see Fig. A.13.

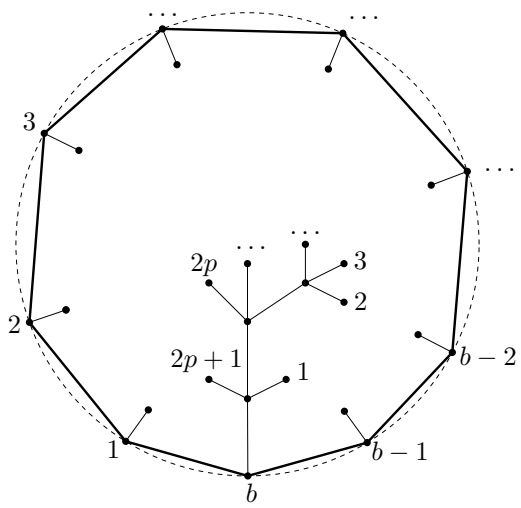


Figure A.13: Enumeration of PAGs using ternary trees

A PAG is now obtained by creating connections between boundary leaves and tree leaves, and connections between tree leaves, while maintaining the planarity of the graph.

1. Firstly we connect each boundary leaf no. i to tree leaf no. f_i. This can be represented by a strictly increasing sequence of numbers $(f_i)_{i=1,..,b-1}$, in which f_1 and every difference $f_{i+1} - f_i$ are odd numbers.
2. Secondly we connect each of the remaining tree leaves with another tree leaf, using assignment functions

   $$a_i \colon \{f_i + 1, \ldots, f_{i+1} - 1\} \to \{f_i + 1, \ldots, f_{i+1} - 1\}$$

   satisfying the conditions

   $$\begin{aligned} a_i(x) = y \Rightarrow (x \neq y \wedge a_i(y) = x)\,, \\ a_i(x_1) = y_1 \wedge a_i(x_2) = y_2 \Rightarrow x_1 \leq x_2, y_2 \leq y_1 \vee y_1 \leq x_2, y_2 \leq x_1 \vee \\ x_2 \leq x_1, y_1 \leq y_2 \vee y_2 \leq x_1, y_1 \leq x_2. \end{aligned}$$

   While the first condition ensures, that always pairs of tree leafs are matched, the second one makes sure, that the connections can be realized in a planar way.
   Geometrically, these assignments correspond to a selection of non-intersecting diagonals of $(f_{i+1} - f_i - 1)$ vertices on a circle numbered by $\{f_i + 1, \ldots, f_{i+1} - 1\}$, such that each vertex belongs to exactly one diagonal.

# patches	# boundary segments 2	4	6	8	10	12
1	0	1	0	0	0	0
2	0	0	3	0	0	0
3	0	0	2	12	0	0
4	0	0	3	18	55	0
5	0	1	6	36	132	273
6	5	4	20	84	330	910
7	9	39	66	252	880	2730
8	17	116	382	825	2698	8191
9	99	511	1476	3792	8969	26423

Table A.3: Number of different PAGs for given numbers of boundary edges and patches

3. Finally we replace all triples of three adjacent edges with two inner valence 2 vertices by a single edge.

We briefly discuss the complexity of this procedure: The number t_p of ternary trees with p nodes can be calculated using the Pólya Enumeration Theorem [26]. We arrive at the recursion

$$t_1 = 1$$
$$t_p = \left(\sum_{i+j+k=p-1} t_i t_j t_k + 3 \sum_{i+j=p-1} t_i t_j + 3 t_{p-1} \right).$$

The first numbers in this sequence are $1, 3, 12, 55, 273$. The number of sequences considered in the first step is bounded by $\binom{2p+1}{b-1}$, and the number of possible pairings in step 2 does not exceed $\frac{M!}{2^{M/2}(M/2)!}$ with $M = 2p - b + 2$. Note that M is even since b is even. The total cost of the algorithm is thus bounded by the product of these three numbers. Clearly, the costs grow quickly with p and b, hence the algorithm can be used for relatively small numbers only.

The procedure gives a catalog of PAG, but each PAG may be present several times. We eliminate duplicates by assigning a unique numbering to the inner vertices and comparing edge lists. *Note that structurally equivalent PAGS with different numbering of the boundary vertices are considered as different.* This is useful as these graphs give different patch structures (except for highly symmetric shapes). Moreover, some of these PAGs may be valid for a given collection of segments, while others may be invalid. This depends on the location of non-convex boundary vertices.

All PAGs obtained from the same ternary tree are different. At most $\binom{p-1+M/2}{M/2}$ different ternary trees can be obtained from a given PAG, hence the number of duplicates is bounded by this number.

As an example, Fig. A.14 shows all PAGs for $b = 6$, $p = 2, \ldots, 6$. These PAGs were a subset of the catalog of PAGs used in the example in Section 4.1. There is only one class of structurally equivalent PAGs for each of the cases $N = 2, \ldots, 5$, but several ones for $N = 6$.

Table A.3 reports the numbers of different PAGs for various values of the number boundary segments b and the number of patches p.

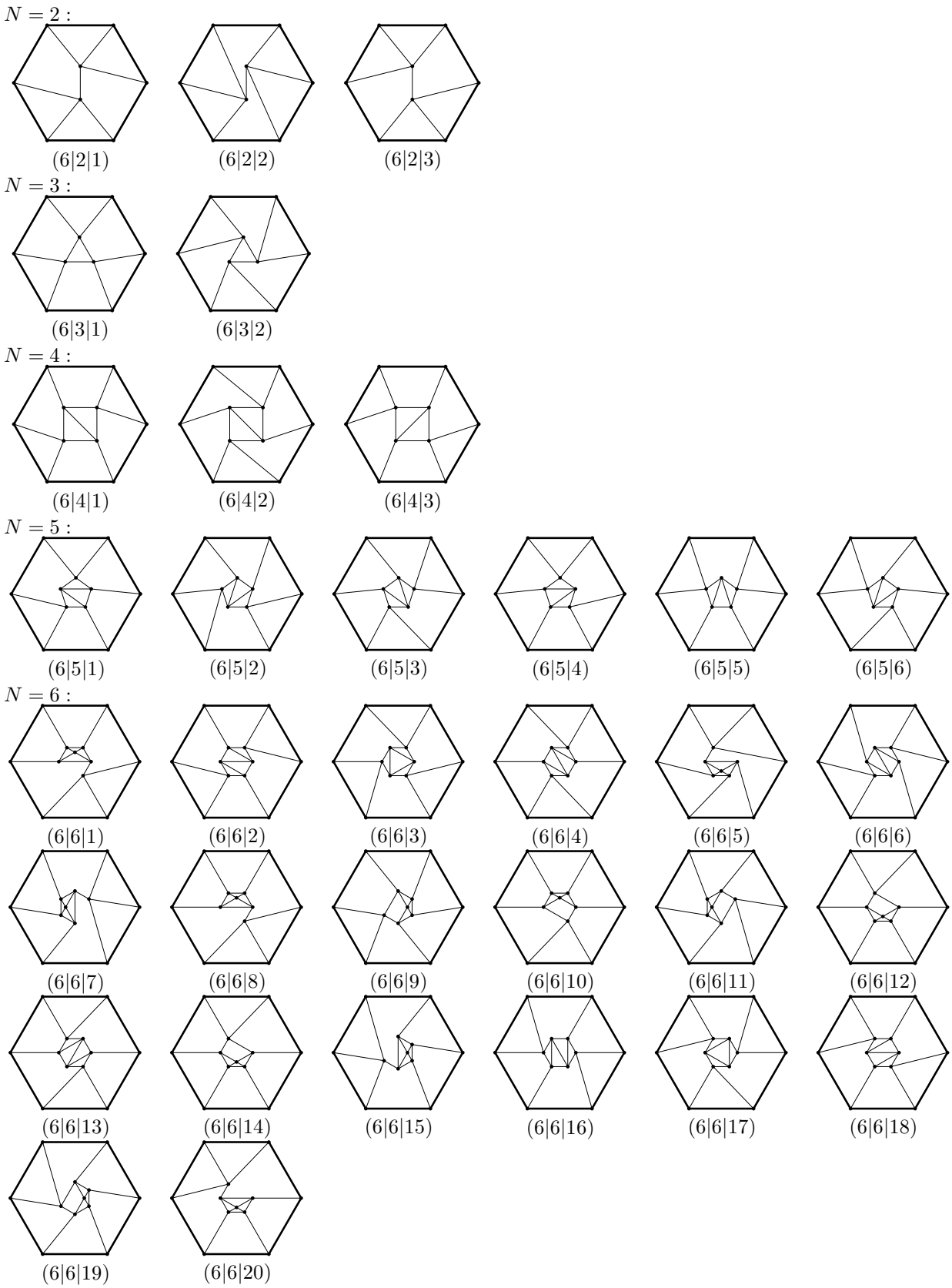


Figure A.14: The PAGs for 6 segments and 2 to 6 patches.

Acknowledgment

Supported by the European Commission through FP7, Projects EXAMPLE, GA no. 324340 and MOTOR, GA no. 678727.

References

[1] M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, and A. V. Vuong. Swept volume parameterization for isogeometric analysis. In *Mathematics of Surfaces XIII*, pages 19–44, Berlin, 2009. Springer.

[2] E. Brivadis, A. Buffa, B. Wohlmuth, and L. Wunderlich. Isogeometric mortar methods. *Comp. Meth. Appl. Mech. Engrg.*, 284:292–319, 2015.

[3] F. Buchegger, B. Jüttler, and A. Mantzaflaris. Adaptively refined multi-patch B-splines with enhanced smoothness. *Applied Mathematics and Computation*, 272, Part 1:159 – 172, 2016.

[4] M. Campen, D. Bommes, and L. Kobbelt. Dual loops meshing: Quality quad layouts on manifolds. *ACM Trans. Graph.*, 31(4):110:1–110:11, July 2012.

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

[6] J. Cottrell, T. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, 2009.

[7] A. Falini, J. Špeh, and B. Jüttler. Planar domain parameterization with THB-splines. *Computer Aided Geometric Design*, 35-36:95–108, 2015.

[8] G. Farin and D. Hansford. Discrete Coons patches. *Comp. Aided Geom. Des.*, 16(7):691–700, 1999.

[9] O. Giménez and M. Noy. Counting planar graphs and related families of graphs. In *Surveys in combinatorics 2009*, pages 169–210. Cambridge Univ. Press, 2009.

[10] J. Gravesen, A. Evgrafov, D.-M. Nguyen, and P. Nørtoft. Planar Parametrization in Isogeometric Analysis. In *Mathematical Methods for Curves and Surfaces*, pages 189–212. Springer, 2014.

[11] D. Großmann. *Volumetric Geometry Reconstruction and Isogeometric Sumulation of Turbine Blades for Aircraft Engines*. PhD thesis, Universität Linz, 2012.

[12] Y. Guo and M. Ruess. Nitsche's method for a coupling of isogeometric thin shells and blended shell structures. *Comp. Meth. Appl. Mech. Engrg.*, 284:881–905, 2015.

[13] V. Gupta, H. Voruganti, and B. Dasgupta. Domain mapping for volumetric parameterization using harmonic functions. *Computer-Aided Design and Applications*, 11(4):426–435, 2014.

[14] C. Hofer and U. Langer. Dual-primal isogeometric tearing and interconnecting solvers for multipatch dg-iga equations. Technical Report 43, NFN Geometry + Simulation, 2015. Available at `www.gs.jku.at/gs_pub.shtml`.

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

[16] S. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. IETI – Isogeometric tearing and interconnecting. *Comp. Meth. Appl. Mech. Engrg.*, 247-248:201–215, 2012.

[17] D. E. Knuth. *The Art of Computer Programming, Vol.1: Fundamental Algorithms*. Addison-Wesley, Mass, 1968.

[18] N. Kowalski, F. Ledoux, and P. Frey. Block-structured hexahedral meshes for CAD models using 3d frame fields. *Procedia Engineering*, 82:59 – 71, 2014. 23rd International Meshing Roundtable (IMR23).

[19] U. Langer, A. Mantzaflaris, S. Moore, and I. Toulopoulos. Multipatch discontinuous Galerkin isogeometric analysis. Technical Report 18, NFN Geometry + Simulation, 2014. Available at `www.gs.jku.at/gs_pub.shtml`.

[20] L. Liu, Y. Zhang, Y. Liu, and W. Wang. Feature-preserving T-mesh construction using skeleton-based polycubes. *Computer-Aided Design*, 58:162 – 172, 2015. Solid and Physical Modeling 2014.

[21] T. Martin, E. Cohen, and M. Kirby. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. In *Proc. ACM Symp. on Solid and Physical Modeling*, pages 269–280, New York, 2008.

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

[23] T. Nguyen and B. Jüttler. Parameterization of contractible domains using sequences of harmonic maps. In *Curves and Surfaces*, pages 501–514. Springer, 2012.

[24] M. Pauley, D.-M. Nguyen, D. Mayer, J. Špeh, O. Weeger, and B. Jüttler. The isogeometric segmentation pipeline. In B. Jüttler and B. Simeon, editors, *Isogeometric Analysis and Applications 2014*, volume 107 of *LNCSE*, pages 51–72. Springer, 2015.

[25] C.-H. Peng, M. Bartoň, C. Jiang, and P. Wonka. Exploring quadrangulations. *ACM Transactions on Graphics*, 33(1), 2014. Article no. 12.

[26] G. Pólya. Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen. *Acta Mathematica*, 68(1):145–254, 1937.

[27] M. Scott, D. Thomas, and E. Evans. Isogeometric spline forests. *Comp. Meth. Appl. Mech. Engrg.*, 269:222–264, 2014.

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

[29] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Comp.-Aided Design*, 45:395–404, 2013.

[30] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Optimal analysis-aware parameterization of computational domain in 3d isogeometric analysis. *Comp.-Aided Design*, 45(4):812–821, 2013.

[31] Y. Zhang, W. Wang, and T. Hughes. Conformal solid T-spline construction from boundary T-spline representations. *Computational Mechanics*, 51(6):1051–1059, 2013.