JOHANNES KEPLER
UNIVERSITY LINZ

Approximate Geometric Continuity for Numerical Simulation and Surface Reconstruction on Multi-patch Domains

Submitted by
Agnes Seiler, MSc

Submitted at
Doktoratskolleg "Computational Mathematics"

Supervisor and
First Examiner
Univ.-Prof. Dr. Bert Jüttler

Second Examiner
Prof. Laureano Gonzalez-Vega

January 2019

Doctoral Thesis
to obtain the academic degree of
Doktorin der Naturwissenschaften
in the Doctoral Program
Naturwissenschaften

**JOHANNES KEPLER
UNIVERSITY LINZ**
Altenbergerstraße 69
4040 Linz, Österreich
www.jku.at
DVR 0093696

Abstract

In the context of geometric modeling and isogeometric analysis, more complex geometries are represented as multi-patch domains that consist of several tensor-product spline patches. Naturally, smoothness across the patch interfaces is an important issue, for the design both of the multi-patch surface itself as well as for functions that are defined on the entire domain. This gives rise to different coupling methods, some of which we study in this thesis.

Exact parametric smoothness is hard to achieve and in most cases requires restrictive assumptions on the parameterization of the geometry. Moreover, in real-world applications approximate smoothness often is sufficient. We consider coupling approaches that work on general domains but only provide approximately $\mathcal{C}^1$-smooth isogeometric functions or approximate $\mathcal{G}^1$-smoothness of a multi-patch surface across patch interfaces.

There are two main possibilities how to encourage smoothness of an approximate solution to a partial differential equation on a multi-patch domain: Firstly, one can make use of the fact that the given problem is fulfilled on all individual patches. The patch-wise terms are summed up and different smoothness penalty terms can be added. One instance of such a method is the discontinuous Galerkin method, to which we devote the first part of this thesis. The discretized problem contains integrals of test functions along the patch interfaces. Their evaluation is crucial. Without matching interface parameterizations, two main difficulties arise in this framework, which we tackle with reparameterizations and suitable quadrature techniques.

Secondly, one can use globally smooth functions on the whole domain as test functions in a continuous Galerkin scheme. Here, the essential part is the construction of such functions. In the second part of this thesis we present an approach to the construction of approximately $\mathcal{C}^1$-smooth isogeometric functions. Starting from globally $\mathcal{C}^0$-smooth functions, the central idea is to bound their gradient jumps across the patch interface. Numerical examples suggest that the resulting

functions are sufficiently smooth to solve higher-order problems such as the biharmonic equation and maintain full approximation power.

Finally, the third part of this thesis considers smooth transitions between surface patches. In order to improve the overall smoothness of a multi-patch spline surface, we consider the simultaneous approximation of point and normal data. If the normal data to be approximated by one patch is taken from the boundary of its neighbors, this controls the behavior of the resulting spline patch along the boundary and ensures approximate G^1-smoothness of the composite surface.

Zusammenfassung

Im Zusammenhang mit geometrischer Modellierung und isogeometrischer Analysis werden komplexe Geometrien als Multipatch-Gebiete, die aus mehreren Tensorprodukt-B-Spline-Flächenstücken bestehen, dargestellt. Sowohl beim Entwurf eines Multipatch-Gebiets selbst als auch bei der Konstruktion von Funktionen, die auf dem gesamten Gebiet definiert sind, ist ein wichtiger Aspekt, dass die Übergänge über Schnittstellen zwischen benachbarten Flächenstücken glatt sind. Daraus ergeben sich verschiedene Kopplungsmethoden, von denen wir einige in dieser Arbeit betrachten.

Exakte parametrische Glattheit zu erreichen, ist schwierig und erfordert in den meisten Fällen strenge Annahmen an die Parametrisierung der Geometrie. Außerdem ist approximative Glattheit in vielen industriellen Anwendungen ausreichend. Wir untersuchen Kopplungsansätze, die auf allgemeinen Gebieten gelten, die aber nur approximativ $\mathcal{C}^1$-glatte isogeometrische Funktionen, bzw. approximativ $\mathcal{G}^1$-glatte Übergänge zwischen den Flächenstücken eines Multipatch-Gebiets erzeugen.

Im Wesentlichen gibt es zwei Möglichkeiten, die Glattheit einer Näherungslösung einer partiellen Differentialgleichung auf einem Multipatch-Gebiet zu fördern: Einerseits kann man die Tatsache, dass das gegebene Problem auf jedem Einzelpatch erfüllt ist, ausnutzen. Die patchweisen Formulierungen werden addiert und verschiedene Glattheitsstrafterme können zu deren Summe hinzugefügt werden. Ein Beispiel dafür ist die discontinuous Galerkin Methode, der wir uns im ersten Teil dieser Arbeit widmen. Die diskretisierte Problemstellung enthält Integrale von Testfunktionen entlang der Schnittstelle zwischen zwei Flächenstücken, deren Auswertung elementar ist. Wenn die Parametrisierungen der Schnittstelle dabei nicht übereinstimmen, entstehen zwei maßgebliche Schwierigkeiten, die wir mit einer Umparametrisierung und geeigneten Quadraturregeln beheben.

Andererseits kann man global glatte Funktionen auf dem gesamten Gebiet als Testfunktionen in einem gewöhnlichen (d.h. continuous) Galerkin Schema benutzen. Dabei besteht die wesentliche Schwierigkeit in der Konstruktion derartiger

Funktionen. Im zweiten Teil dieser Arbeit stellen wir einen Ansatz zur Konstruktion von approximativ $\mathcal{C}^1$-glatten isogeometrischen Funktionen vor. Die grundlegende Idee hierbei ist, von $\mathcal{C}^0$-glatten Funktionen auszugehen und den Sprung ihrer Gradienten über eine Patch-Schnittstelle hinweg zu beschränken. Numerische Versuche legen nahe, dass die so erzeugten Funktionen glatt genug sind, um Probleme höherer Ordnung wie die biharmonische Gleichung zu lösen und dass sie daneben die optimale Approximationsgüte besitzen.

Zu guter Letzt befassen wir uns im dritten Teil dieser Arbeit mit glatten Übergängen zwischen Flächenstücken. Um die Glattheit einer zusammengesetzen Fläche zu erhöhen, betrachten wir Punkt- und Normalenapproximation auf einem einzelnen Patch gleichzeitig. Wenn die anzunähernden Normalendaten vom Rand eines benachbarten Flächenstücks stammen, steuert dieser Vorgang das Verhalten des Splinepatches entlang seines Randes und stellt die approximative Glattheit der zusammengesetzten Fläche sicher.

Acknowledgments

First of all, I want to thank Prof. Bert Jüttler for his supervision and guidance, for being available on short notice when a problem seemed unsolvable, for being full of ideas how the problem could be solved and for passing on not only expertise but also motivation. Thank you for thoroughly revising reports, papers and this thesis. At the same time I owe my thanks to Prof. Laureano Gonzalez-Vega for reviewing this thesis.

During my stay in Munich and beyond that time, Dr. David Großmann supported my work substantially, for which I would like to thank him.

The support by the Seventh Framework Program of the EU, project EXAMPLE (GA no. 324340), and the Austrian Science Fund (FWF) through the DK W1214-04 is gratefully acknowledged. I would like to add that Prof. Peter Paule and Dr. habil. Veronika Pillwein invested great effort in bringing together numerical analysis and symbolic computation and that countless talks and seminars in mixed groups broadened my mathematical horizon.

Besides my colleagues at the Institute of Computational Mathematics and my office mates Katharina, Simon, Barbara and Martin, I would especially like to thank my colleagues at the Institute of Applied Geometry and at the DK for numerous interesting talks and discussions, as well as for making the work environment so pleasant by mixing mathematical and other topics. I would like to name Gabi and Monika in particular, for being extraordinarily welcoming and helpful and creating an atmosphere that makes you look forward to coming to the office.

Finally, I want to express my gratitude to my parents, my boyfriend Felix and his family, as well.

Contents

1 Introduction **9**

2 Preliminaries **19**

3 Approximate Smoothness for Numerical Simulation **21**

3.1 The model problem and the derivation of a dG scheme 21
3.2 Non-matching parameterizations along the interface 24
3.3 DG-IgA discretization of the model problem 25
3.4 Integrals along interfaces 28
3.5 Finding the reparameterizations 30
3.6 Numerical integration 32
    3.6.1 Gauss quadrature with exact splitting 32
    3.6.2 Gauss quadrature with uniform splitting 34
    3.6.3 Adaptive Gauss quadrature 34
3.7 Numerical results 36
    3.7.1 Reference results 36
    3.7.2 Influence of the quadrature rule 37
    3.7.3 Influence of the reparameterization 38
    3.7.4 Comparison of exact and adaptive quadrature 39
3.8 Summary 41

4 Approximately $\mathcal{C}^1$-smooth Isogeometric Functions **43**

4.1 Preliminaries 45
4.2 Results for $\mathcal{B}_1$ 47

- 4.2.1 Construction of approximately smooth functions 47
- 4.2.2 Properties of the function space 48
- 4.3 Results for $\mathcal{B}_2$ 50
  - 4.3.1 Construction of approximately smooth functions 50
  - 4.3.2 Properties of the function space 50
- 4.4 Numerical examples 52
  - 4.4.1 Approximation power 53
  - 4.4.2 Dimension of the space 63
- 4.5 Summary 63

5 Approximate Geometric Smoothness for Surface Reconstruction 67

- 5.1 Simultaneous approximation of point and normal data 68
- 5.2 Existence of a solution and convergence rates 72
- 5.3 Numerical experiments 76
  - 5.3.1 Least-squares fitting: synthetic data 76
  - 5.3.2 Least-squares fitting: industrial data 77
  - 5.3.3 Norm-like functions 81
- 5.4 Summary 82
- 5.5 Appendix: Proof of Lemma 4. 83

6 Conclusion and Future Work 87

List of Figures 91

List of Tables 95

Declaration of Authorship 105

Curriculum Vitae 107

Chapter 1

Introduction

Partial differential equations (PDEs) are a powerful mathematical tool with which many different physical phenomena can be modeled, such as diffusion, heat distribution, elasticity, fluid-structure interaction and electro-magnetic fields, just to name a few. There are only a few types of boundary or initial value problems for which a closed-form solution can be found, i.e., in most cases it is not possible to solve the given problem analytically. Thus, approximate solutions are constructed. Therefore, the problem is discretized on a finite dimensional space. The approximate solution is controlled by finitely many unknowns, called degrees of freedom.

Intensively studied methods for solving partial differential equations numerically include finite difference schemes, finite element or finite volume methods and boundary element methods. They share the property of approximating the original geometry on which the problem is to be solved. In finite element methods, the domain is divided into simple forms, e.g. triangles. Based on these, finitely many basis functions are constructed and finally the approximate solution is represented as a linear combination of the basis functions. The unknown coefficients of the basis functions in the solution are the degrees of freedom that we mentioned earlier.

A drawback of this method is the loss of accuracy when approximating the geometry. Furthermore, for complicated geometries the triangulation step may take much more time than the analysis. In real world applications, first the geometry is modeled with computer aided design (CAD) techniques, using B-splines or NURBS. Afterwards, it is approximated by a triangulation which is then used for analysis purposes. All calculations are therefore carried out with respect to an incorrect representation of the geometry.

Consequently combining the design and the analysis steps is an immense step forwards. Isogeometric Analysis (IgA), which was introduced in 2005 by Hughes et. al [14, 15], is an approach to solving a PDE approximately, that closes the gap between modeling the geometry and numerical simulation. Its main idea is to use the same functions twice: once in the parameterization of the geometry, and once as test functions in a Galerkin scheme. B-splines and NURBS possess the same useful properties as finite elements: although their support is typically larger than the support of finite elements, the system matrix will be sparse and they enjoy the same approximation power as finite element functions with respect to the degree [3].

A key ingredient of the IgA discretization is the parameterization of the physical domain, called geometry mapping, that transforms the parameter domain into the physical domain. Often, the geometry mapping is a tensor product parameterization, where the parameter domain is a unit square or a unit cube. Consequently, more complex domains have to be divided into several single patches, forming a multi-patch geometry. This is mostly the case in industrial applications, where the computational domains for example can be cars, motors, turbines or parts thereof. In this context smoothness of the global solution across patch interfaces is an important issue. The patch-wise solutions need to coincide along the interfaces, at least approximately. The first two parts of this thesis focus on different coupling techniques for isogeometric functions on multi-patch domains. We restrict ourselves to two-patch domains in order to keep the representation simple. Moreover, this is sufficient to explain the concepts of the presented coupling methods.

There is a large variety of methods covering the coupling task. In order to point out differences between them, let us start with describing the standard procedure of deriving an approximate solution to a PDE.

First, we choose a test function space of infinite dimension, then we multiply the given problem by a function from the test function space, integrate (over the complete domain or patch-wise) and as a last step, we use integration by parts, i.e., apply Green's Theorem. This leads to the variational formulation. In the subsequent discretization step, the problem is transferred to a finite dimensional setting via the selection of a finite dimensional test function space, where the approximate solution is to be found. The multi-patch structure of the domain needs to be taken into account - either in the variational formulation of the problem or in the choice of the finite dimensional test function space.

The former possibility, i.e., including coupling conditions in the weak formulation, enables us to work with spaces of test functions, which are differentiable only on the individual patches. Such spaces are called broken Sobolev spaces. Thus, (weak) differentiability of the test functions across the interface is not guaranteed. The order of the Sobolev spaces on the single patches must be adapted to the order of the problem.

Standard coupling methods from finite element methods carry over to the isogeometric approach. Coupling terms are added to the weak form of a partial differential equation. For instance, the mortar method [4, 5, 7], Nitsche mortaring [52, 62] or the discontinuous Galerkin method [46, 54] rely on the fact that the original problem is fulfilled on each subdomain and that the solution can be coupled with average and jump terms. Those terms need to be adapted to the order of the problem, i.e. for second order problems the jump of function values is used, fourth order problems require to bound the jump of normal derivatives. Analogously, via the relation between a coercive bilinear form of an elliptic problem and its equivalent quadratic optimization problem, methods from non-linear optimization can be applied. Here, discontinuities in the ansatz and test functions are either directly penalized or smoothness is enhanced via Lagrange multipliers [27]. Let us have a closer look at some of the mentioned methods.

- Mortar methods use weak continuity conditions on the interfaces. They require the jump of the mortar functions to be orthogonal in L^2 to a space of trace test functions on the interface. Subsequently, these test functions are interpreted as Lagrange multipliers. The mortar methods for spectral and finite elements are explained in more detail in [5] and transferred to the isogeometric case in [7]. The choice of a finite dimensional subspace of the trace test function space is crucial and changes the resulting method.

- Nitsche's method can be seen to lie inbetween mortar and penalty methods. Originally it was introduced to weakly enforce Dirichlet boundary conditions, but in the same fashion, it can be used for interpatch coupling. The main idea is to replace the Lagrange multipliers of the mortar method by the normal flux and to add a term to ensure the coercivity of the resulting bilinear form.
- The discontinuous Galerkin (dG) method encourages $\mathcal{C}^1$-smoothness by penalizing the jump of the gradient of the solution across the interface. Its variational form is derived by rewriting the problem formulation on the individual patches, followed by adding a penalty term to restore coercivity of the problem and possibly a symmetrization term. Contrary to continuous Galerkin methods (see below), dG methods allow for hanging nodes and hence also for adaptive refinement strategies. Furthermore, different polynomial degrees can be used on different patches.

Discontinuous Galerkin methods

In the first part of this thesis, we focus on dG methods and consider the following example problem on a domain $\Omega \subseteq \mathbb{R}^2$ that consists of two patches, i.e.,

$$\bar{\Omega} = \bar{\Omega}^1 \cup \bar{\Omega}^2, \ \Omega^1 \cap \Omega^2 = \emptyset.$$

We denote the partition of Ω by $\mathcal{T} = \{\Omega^1, \Omega^2\}$. The Poisson problem with homogeneous Dirichlet boundary conditions consists in finding $u : \Omega \to \mathbb{R}$ such that

$$\begin{aligned} -\Delta u &= f \text{ on } \Omega \\ u &= 0 \text{ on } \partial\Omega \end{aligned} \tag{1.1}$$

for the a function $f \in L^2(\Omega)$. For this second order problem, the required broken Sobolev space is given by

$$H^s(\mathcal{T}) := \{f \in L^2(\Omega) : f|_{\Omega^1} \in H^s(\Omega^1), f|_{\Omega^2} \in H^s(\Omega^2)\} \tag{1.2}$$

with $s > \frac{3}{2}$, where

$$H^s(\Omega^k) = \{u \in L^2(\Omega^k) : D^\alpha u \in L^2(\Omega^k) \, \forall \, 0 \leq |\alpha| \leq s\}$$

denotes the standard Sobolev space on Ω^k.

The publications [12, 54] provide a general description of dG techniques in the context of finite elements, which have been transferred to isogeometric discretizations in [9, 44, 45, 46]. A more detailed derivation of the dG formulation of (1.1) is given at the beginning of Chapter 3. For the time-being let us just state the symmetric interior penalty dG method for (1.1):

Find $u \in H^s(\mathcal{T})$ such that

$$\begin{aligned}\sum_{k=1}^{2} \int_{\Omega^k} \nabla u \nabla v - \sum_{k=1}^{2} \sum_{\gamma \in \partial\Omega^k} \int_{\gamma} \left(\{\nabla u \cdot n\}[v] + \{\nabla v \cdot n\}[u] + \delta[u][v] \right) \\ = \int_{\Omega} f v \quad \forall v \in H^s(\mathcal{T}),\end{aligned} \tag{1.3}$$

where $\{\cdot\}$ and $[\cdot]$ denote the average and jump operator, respectively, and δ is a penalty parameter. Subsequently, (1.3) is discretized by means of isogeometric function spaces on both patches. The resulting approximate solution to (1.1) will be approximately, but not exactly parametrically smooth.

To find a solution to the weak form (1.3), it is discretized by means of isogeometric function spaces. The evaluation of the integrals on the interface in (1.3) is crucial: Since the interface is formed by two boundary edges of two different patches, the parameterizations of the two edges need to be taken into account. If they are different from each other, the evaluation becomes considerably more complicated.

So far, only matching interface parameterizations have been studied in the context of dG-IgA methods. More precisely, whenever two patches meet in an interface, then the parameterizations restricted to these interfaces are assumed to be identical (possibly after affine transformations of the parameter domains), see [44, 45, 46, 64]. On the one hand, this limitation provides the advantage that the elements of the patches on both sides of the interface are perfectly matching, which significantly simplifies the implementation of such methods. On the other hand, it substantially complicates the creation of multi-patch parameterizations.

As notable exceptions we mention the recent publications [29, 30], where the authors study gaps and overlaps at the interfaces. While the theory presented in these papers does not require any assumptions regarding matching interfaces, such conditions are assumed to be satisfied in all the computational examples. More precisely, the meshes of the considered domains fulfill restrictive correspondence conditions, which are quite similar to the matching case. This is due to the lack of an implementation for the non-matching case, not to theoretical requirements [46].

Chapter 3 of this thesis is devoted to the study of non-matching parameterizations along the interface in the context of a dG-IgA discretization of a given PDE. We explain the necessary steps to evaluate the affected integrals. The results presented here were published in [57].

Approximately $\mathcal{C}^1$-smooth isogeometric functions

In the second part of this thesis we turn our attention to test function spaces of globally smooth or approximately smooth functions. Including coupling conditions in the test function space leads to conforming methods. The weak form of (1.1) is derived as follows: Let

$$\mathcal{V} := H_0^1(\Omega) = \{u \in L^2(\Omega) : \partial_x u, \partial_y u \in L^2(\Omega), u|_{\partial\Omega} = 0\}$$

denote the Sobolev space of weakly differentiable functions with measurable derivatives on Ω which fulfill the zero-boundary conditions. Multiplying (1.1) with a function $v \in \mathcal{V}$, integrating over Ω and using Green's theorem yields the following variational problem:

Find $u \in \mathcal{V}$ such that

$$\int_\Omega \nabla u \nabla v = \int_\Omega f v \quad \forall v \in \mathcal{V}. \tag{1.4}$$

The finite dimensional test function space $\mathcal{V}_h$ is chosen as a subspace $\mathcal{V}_h \subseteq \mathcal{V}$, in particular all test functions have to be weakly differentiable on the whole domain Ω. Thus, the discretized problem reads:

Find $u \in \mathcal{V}_h$ such that

$$\int_\Omega \nabla u_h \nabla v_h = \int_\Omega f v_h \quad \forall v_h \in \mathcal{V}_h. \tag{1.5}$$

Methods of this kind are also refered to as continuous Galerkin methods. The weak forms (1.4) and (1.5) are considerably simpler than the dG form (1.3). In an isogeometric setting, the challenge of this approach lies in constructing test function spaces of isogeometric functions that are smooth across patch interfaces. In recent years many publications covered this topic:

- Global $\mathcal{C}^0$-smoothness can be realized by identifying degrees of freedom along the interfaces. Obviously, this only works for matching parameterizations of the interface. For higher order problems, $\mathcal{C}^0$-continuity is not sufficient.

- Fourth order problems like the biharmonic equation require test functions from $H^2(\Omega)$. Therefore one needs to construct $\mathcal{C}^1$-smooth isogeometric functions which is remarkably more difficult. Recent results for planar domains [33, 40] are based on the relation between parametric continuity of a function and geometrical continuity of the corresponding graph surface [26]. A numerical approach to the computation of $\mathcal{C}^1$-smooth discretization is presented in [11]. The volumetric case is covered by [6]. However, the construction of such functions imposes quite restrictive conditions on the parameterization of the underlying geometry. Here, the parameterization needs to be bilinear or bilinear-like.

  In [13, 35, 36] the authors provide possibilities to work around some of these restrictions by suitably reparametrizing non-bilinearly parametrized geometries.

- For sixth order problems, e.g. the triharmonic equation, $\mathcal{C}^2$-smooth test functions are necessary. The construction and analysis of such functions for isogeometric discretizations on planar bilinearly parametrized domains can be found in [37, 38, 39].

- In contrast to the previously listed publications, the publications [10, 34] focus on the construction and analysis of smooth splines on the parameter domain, referred to as multi-patch B-splines with enhanced smoothness, which subsequently can be used in the context of IgA. They are also suitable for adaptive refinement, which is an issue that was not addressed in the other mentioned publications.

All mentioned publications refer to the exact notion of parametric or geometric smoothness, respectively. Computationally, this is a rather strict limitation to the construction of such functions, as it requires the exact, i.e., symbolic, determination of the kernel of the system matrix. Also, the analytical derivation of constructive conditions on such functions is challenging as pointed out earlier.

The fourth chapter of this thesis is devoted to the construction of spaces of approximately $\mathcal{C}^1$-smooth isogeometric functions on general domains. Since we do not make any assumptions on the parametrization of the geometry, we lose the notion of exact smoothness. Hence, similarly to the idea of dG schemes, we aim at bounding the gradient jump of such functions. The resulting spaces will then serve as discretization spaces. Consequently, no additional penalty terms or constraints are required in the variational formulation.

Our approach follows the idea that in industrial applications approximate smoothness may be sufficient for subsequent steps. Furthermore, spaces of exactly smooth isogeometric functions can be of low dimensionality. Approximately smooth functions could overcome this problem and at the same time be more efficiently computable. The results presented in Chapter 4 are published in [58].

Spline surface reconstruction

The third part of this thesis is devoted to a slightly different field, namely spline surface reconstruction.

Spline surface fitting is a well-established technique for surface reconstruction from point data [20, 28]. It is frequently used in industrial applications, where the scanning of mechanical components results in point cloud data. For subsequent processes it is often necessary to find a surface that represents the point data. Typically, the triangulated data sets are segmented and the resulting pieces are parameterized and fitted individually. After the fitting step, the collection of the approximating patches forms the geometric model that is now suitable for further processing.

The importance of fitting in an industrial context, i.e., as a part of reverse engineering [61], is evident from the substantial number of publications on this topic. Besides algebraic and implicit surfaces [32] as well as subdivision surfaces, see [48, 50], which appear to be not fully supported by all existing software tools [59], B-spline and NURBS surfaces are discussed in the literature. We restrict ourselves to this class of surfaces.

There are various ways to approach the fitting problem. The intuitive least squares method for B-splines, which reduces to solving a linear system of equations, depends on finding a valid parameterization and knot vectors, which requires powerful optimization methods [23]. When using NURBS, one also has the possibility to choose the weights, although this is hardly done in practice.

Advanced techniques include iterative methods for B-splines such as applying a quasi-Newton method for active curves and surfaces [53], adaptations of iterative geometric interpolation and approximation algorithms [42], iterative approaches for NURBS that avoid solving a linear system at all [47], formulations of the fitting problem as more general constrained optimization problems [22] and hybrid optimization algorithms for NURBS [63]. Several publications also focus on improving the computational efficiency, such as [8]. Finally we mention that progressive iterative approximation [18, 19] is a common technique.

Achieving smoothness across patch interfaces is of fundamental importance for generating a high-quality geometric model. Approximate methods for the coupling of patches across interfaces are treated e.g. by [51, 59], whereas exact methods are discussed in the recent book of [41].

In Chapter 5, we consider the spline fitting problem on a single patch including normal boundary conditions which makes our approach also suitable for generating multi-patch models, in which the boundary conditions may arise from data of neighboring patches. In this context, it is often more important to achieve approximate G^1-smoothness across interfaces between patches than to approximate given point data along interfaces as good as possible. Also, real-world data most likely will contain measurement errors, making it less reasonable to enforce a highly precise fitting result along the interfaces. Mathematically speaking, we simultaneously approximate point and normal data, where the latter are suitably weighted. The term containing the normal data can be seen in analogy to the penalty terms mentioned in the context of coupling an approximate solution to a

PDE on a multi-patch domain. It penalizes deviations, i.e., jumps, in the normals between neighboring patches.

Since all patches are considered separately, a geometric model produced by our fitting procedure will be only approximately G^0- (and, in addition to it, approximately G^1-) smooth. In certain applications, such as numerical flow simulations, it is desirable to maintain global C^0-smoothness; this can be achieved by simply identifying boundary control points of adjacent patches and solving the fitting problem on all patches simultaneously, as pointed out earlier. The results we present in this part of the thesis were published in [56].

Outline

This thesis is organized as follows: In Chapter 2 we introduce the common notation for the remainder of this thesis. Chapter 3 addresses the discontinuous Galerkin method. We will recall the basic principle, introduce the notion of non-matching interfaces and explain how to tackle the computational difficulties arising therefrom. The construction of approximately $\mathcal{C}^1$-smooth isogeometric functions is described in Chapter 4. We then leave the isogeometric field to focus on spline surface reconstruction. The simultaneous fitting of point and normal data by B-splines will by presented in Chapter 5. Finally, Chapter 6 concludes the thesis and points out future work.

Chapter 2

Preliminaries

Throughout Chapters 3 and 4 of this thesis we consider a planar multi-patch domain $\Omega \subseteq \mathbb{R}^2$ that consists of the two non-overlapping single patches Ω^1, Ω^2 such that

$$\bar{\Omega} = \bar{\Omega}^1 \cup \bar{\Omega}^2 \text{ and } \Omega^1 \cap \Omega^2 = \emptyset. \tag{2.1}$$

We will refer to the partition of Ω as $\mathcal{T} = \{\Omega^1, \Omega^2\}$. The single interface between Ω^1 and Ω^2 will be denoted by $e = \bar{\Omega}^1 \cap \bar{\Omega}^2$. We assume that the two patches only share this edge. We define the set of all edges

$$\Gamma = \bigcup_{k=1}^{2} \partial\Omega^k \tag{2.2}$$

of the multi-patch domain. It is the disjoint union of the set of the interface edge

$$\Gamma_C = \{\gamma \in \Gamma : \gamma \subseteq \bar{\Omega}^1 \cap \bar{\Omega}^2\} = \{e\} \tag{2.3}$$

and the set of boundary edges

$$\Gamma_D = \{\gamma \in \Gamma : \gamma \subseteq \bar{\Omega}^k \cap \partial\Omega\,,\, k = 1, 2\}\ . \tag{2.4}$$

The parameterization of the domain Ω is described by tensor-product spline functions, either patch-wise as used in Chapter 3 or by a global geometry map as in Chapter 4. For their construction we will use open knot vectors. Thus, the boundary curves are spline curves whose control points are the outer control points of the patch itself.

On the physical domain we define isogeometric functions as concatenation of tensor-product B-splines and the geometry map. A more detailed description is given in the following chapters, as we will work with two slightly different constructions. More precisely, in Chapter 3, we consider isogeometric functions that are defined on single patches, whereas in Chapter 4 we need the isogeometric functions to be globally $\mathcal{C}^0$-smooth.

We will refer to the standard Sobolev space on Ω either by $H^s(\Omega)$ or by $H^s_0(\Omega)$, depending on whether the functions in the Sobolev space shall fulfill zero boundary conditions, i.e.

$$H^s(\Omega) = \{f \in L^2(\Omega) : D^\alpha f \in L^2(\Omega)\, \forall\, 0 \leq |\alpha| \leq s\} \tag{2.5}$$

and

$$H^s_0(\Omega) = \{f \in H^s(\Omega) : f|_{\partial\Omega} = 0\}. \tag{2.6}$$

Furthermore, in the context of multi-patch domains, the notion of broken Sobolev spaces is important. In the two-patch case this space can be denoted by

$$H^s(\mathcal{T}) = \{f \in L^2(\Omega) : f^1 = f|_{\Omega^1} \in H^s(\Omega^1), f^2 = f|_{\Omega^2} \in H^s(\Omega^2)\}. \tag{2.7}$$

Analogously to the non-broken case, zero boundary conditions can be incorporated. The superscript of a functions refers to its restriction to the patch indicated by the superscript, i.e. $f^k = f|_{\Omega^k}, k = 1, 2$.

For each patch index k, any function $v \in H^1(\Omega^k)$ has a well-defined trace along any edge $\gamma \in \partial\Omega^k$. Hence a function $v \in H^1(\mathcal{T})$ defines *two* traces on the interface $e = \bar{\Omega}^1 \cap \bar{\Omega}^2$, which we denote as $v^1|_e$ and $v^2|_e$, respectively. We use them to define the average

$$\{v\}^e = \frac{1}{2}\left(v^1|_e + v^2|_e\right) \tag{2.8}$$

and the jump

$$[v]^e = v^1|_e - v^2|_e \tag{2.9}$$

across the interface e. These definitions are further extended to boundary edges $\gamma \in \Gamma_D$,

$$\{v\}^\gamma = v^k|_\gamma \text{ and } [v]^\gamma = v^k|_\gamma,\ k = 1, 2\ . \tag{2.10}$$

Chapter 3

Approximate Smoothness for Numerical Simulation

In this chapter we study certain aspects of the isogeometric discontinuous Galerkin (dG) discretization, in particular the computation of the stiffness matrix elements in the case of non-matching interface parameterizations. Before pointing out the relevance of non-matching parameterizations we shortly recall the derivation of the dG scheme for a simple model problem. Hereby, we restrict ourselves to a two-patch case due to better readability. All observations generalize directly to domains with more than two patches.

3.1 The model problem and the derivation of a dG scheme

A detailed derivation of the dG discretization is given in [54]. The adaptation to the isogeometric setting is discussed in the thesis [9], which also comments on the choice of the the penalty parameter δ (see below), and in the recent article [44]. We include the derivation in this thesis in order to make this thesis self-contained and facilitate the comparison of the dG method with other coupling techniques.

Given a two-patch domain $\bar{\Omega} = \bar{\Omega}^1 \cup \bar{\Omega}^2 \subseteq \mathbb{R}^2$ with interface e, we consider the

Poisson problem

$$\text{Find } u : \begin{cases} -\nabla \cdot (\alpha \nabla u) & = f \text{ on } \Omega \\ u & = 0 \text{ on } \partial\Omega \ , \end{cases} \tag{3.1}$$

where f is given and $\alpha > 0$ is the known diffusion coefficient. Theoretically, α can be piecewise constant, i.e., it may take different values on every single patch. However, in order to keep the presentation simple, we restrict ourselves to the case where $\alpha \equiv 1$ on Ω.

Starting from (3.1) on a single patch Ω^k, we multiply with a test function $v \in \mathcal{V} := H^s(\mathcal{T})$ with $s > \frac{3}{2}$, integrate over Ω^k and apply integration by parts, i.e., Green's theorem in the form

$$-\int_D w\Delta v = \int_D \nabla v \nabla w - \int_{\partial D} (\nabla v \cdot n_D) w$$

for a bounded domain D, $v \in H^2(D)$ and $w \in H^1(D)$. This results in

$$\int_{\Omega^k} \nabla u \nabla v - \int_{\partial\Omega^k} (\nabla u \cdot n) v = \int_{\Omega^k} f v \, \forall v \in \mathcal{V}.$$

We sum over both patches and get

$$\sum_{k=1}^{2} \int_{\Omega^k} \nabla u \nabla v - \sum_{k=1}^{2} \sum_{\gamma \in \partial\Omega^k} \int_\gamma (\nabla u \cdot n^k) v = \int_\Omega f v \, \forall v \in \mathcal{V}.$$

Note that the middle terms contains twice an integral over the interface $e = \bar{\Omega}^1 \cap \bar{\Omega}^2$. Since the outer normal vector n^2 of the second patch fulfills $n^2 = -n^1$ for the outer normal vector n^1 of the first patch, we can rewrite this term as

$$\sum_{\gamma \in \Gamma_D} \int_\gamma (\nabla u \cdot n) v + \int_e \left((\nabla u^1 - \nabla u^2) \cdot n^1 \right) v.$$

By means of the jump operator this equals

$$\sum_{\gamma \in \Gamma} \int_\gamma [(\nabla u \cdot n) v].$$

A short computation confirms that this again equals

$$\sum_{\gamma \in \Gamma} \int_\gamma \{\nabla u \cdot n\}[v].$$

Thus, the first version of the dG variational form, which is equivalent to (3.1), reads as follows:

Find $u \in \mathcal{V}$ such that

$$\sum_{k=1}^{2} \int_{\Omega^k} \nabla u \nabla v - \sum_{\gamma \in \Gamma} \int_{\gamma} \{\nabla u \cdot n\}[v] = \int_{\Omega} f v \, \forall v \in \mathcal{V}.$$

The corresponding bilinear form is not symmetric and, more important, not coercive. Therefore, we add the terms

$$\begin{aligned}
&\int_e \frac{\delta}{|\gamma|}[u][v],\\
&-\int_e \{\nabla v \cdot n\}[u],\\
&\sum_{\gamma \in \Gamma_D} \int_{\gamma} (\nabla v \cdot n) u = \sum_{\gamma \in \Gamma_D} \int_{\gamma} \{(\nabla v \cdot n)\}[u] \text{ and}\\
&\sum_{\gamma \in \Gamma_D} \int_{\gamma} \frac{\delta}{|\gamma|} uv = \sum_{\gamma \in \Gamma_D} \int_{\gamma} \frac{\delta}{|\gamma|}[u][v],
\end{aligned}$$

which are consistent since for the exact solution all these terms vanish, as the exact solution is smooth across the interface and vanishes on $\partial\Omega$. The first term is a penalty term, penalizing jumps of the test functions weighted with the penalty parameter δ which has to be chosen sufficiently large to restore coercivity. The value $|\gamma|$ measures the length of the edge γ. The second and the third term together symmetrize the bilinear form. The third and the fourth term weakly impose the Dirichlet boundary conditions. The final problem is given as follows:

Find $u \in \mathcal{V}$ such that

$$\sum_{k=1}^{2} \int_{\Omega^k} \nabla u \nabla v - \sum_{\gamma \in \Gamma} \int_{\gamma} \left(\{\nabla u \cdot n\}[v] + \{\nabla v \cdot n\}[u] - \frac{\delta}{|\gamma|}[u][v] \right) = \int_{\Omega} f v \, \forall v \in \mathcal{V}. \tag{3.2}$$

It is also possible to impose the Dirichlet conditions strongly via an L^2 projection. In case of inhomogeneous boundary conditions the problem can then be homogenized. When both u and the test functions v vanish on the boundary $\partial\Omega$, the dG variational form only contains integrals on the patches Ω^k and the interface e.

Finding a solution to the final problem is called the symmetric interior penalty dG method for the Poisson problem. An approximate solution is constructed by discretizing the problem with respect to an isogeometric setting, and evaluating the discrete terms. This is where non-matching parameterizations make a significant difference.

3.2 Non-matching parameterizations along the interface

Recent work has motivated us to investigate the effect of non-matching interface parameterizations in the context of discontinuous Galerkin - Isogeometric Analysis (dG-IgA). We aim to give a complete description of the necessary computational steps for applying the theoretical results of [29, 30, 44, 45, 46, 64] to the case of non-matching parameterizations at the interfaces. In order to keep the presentation simple, we restrict ourselves to planar two-patch domains and we assume that the interfaces are geometrically matching, thus they have neither overlaps nor gaps. However, it is clear that the results from [29, 30] apply to the non-matching case also, as the theory presented there is sufficiently general.

More precisely, the assembly of the local stiffness matrices derived from the dG bilinear form requires the computation of integrals of the type

$$\int_e D\, b_i^k(\boldsymbol{x}) D'\, b_j^\ell(\boldsymbol{x}) d\boldsymbol{x} \ , \tag{3.3}$$

where D, D' are differential operators. As we shall see, non-matching interface parameterizations give rise to two problems that need to be treated separately.

The first one concerns the evaluation of $b_i^k(\boldsymbol{x})$ and $b_j^\ell(\boldsymbol{x})$ at the same position $\boldsymbol{x}$ on the interface. Due to the use of non-matching parameterizations, a point $\boldsymbol{x}$ will have two possibly different preimages in the parameter domains of the two patches joined by the interface respectively. To identify pairs of corresponding preimages we use reparameterizations of the preimages of the interface. We also investigate the influence of the quality of the reparameterization on the accuracy of the overall result.

The second problem is related to the use of numerical integration methods. We need to find a quadrature method whose exactness does not depend on the smoothness of the integrands. We present different approaches, one resulting from dividing the element on which quadrature is performed and another one making use of automatized element splitting. The performance of both approaches is explored in numerical experiments.

The remainder of this chapter is structured as follows: We describe the discrete problem we will focus on in the next section. We then state the two issues of evaluation and numerical integration, as described above. Section 3.5 treats the first problem of finding suitable reparameterizations, while Section 3.6 is devoted to the different quadrature techniques. Results of numerical experiments are presented in Section 3.7. Finally we conclude the chapter.

3.3 DG-IgA discretization of the model problem

The isogeometric discretization of the given model problem stated previously is derived as follows. Each physical subpatch Ω^k of the domain is parameterized by an associated geometry mapping G^k which maps the parameter domain $\hat{\Omega}^k = [0,1]^2$ to Ω^k by means of tensor-product B-splines β_i^k, $k = 1, 2$, where the subscript i refers to the index of the respective basis function. The superscript k indicates the patch index. Thus, G^k is given in the form

$$G^k(\xi,\eta) = \sum_{i \in \mathcal{I}^k} P_i^k \beta_i^k(\xi,\eta), (\xi,\eta) \in \hat{\Omega}^k, \tag{3.4}$$

where $P_i^k \in \mathbb{R}^2$ are the two-dimensional control points for patch Ω^k and $\mathcal{I}^k$ is the index set corresponding to the basis functions defined on $\hat{\Omega}^k$. A sketch of the setting is shown in Figure 3.1. With the parameterization, the set Γ of all edges can equivalently be written as

$$\Gamma = \bigcup_{k=1}^{2} \{G^k([0,1],0),\, G^k([0,1],1),\, G^k(0,[0,1]),\, G^k(1,[0,1])\}. \tag{3.5}$$

The knot vectors of the B-splines β_i^k split the parameter domain into elements, and their images induce a mesh on the physical domain. The maximal element

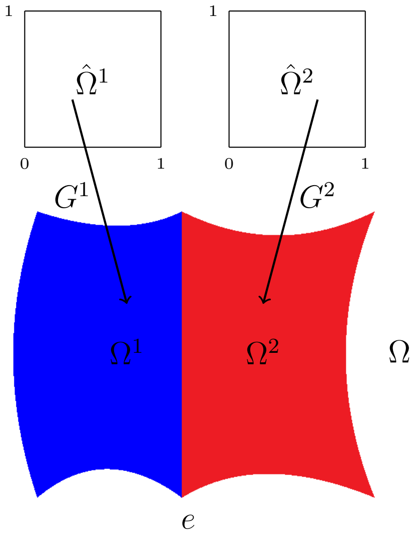


Figure 3.1: Multi-patch domain with two patches Ω^1, Ω^2, one interface e and geometry mappings G^1, G^2.

side length will be denoted by h. We do not assume that the knot vectors of the patches are identical.

An isogeometric basis function b_i^k on the physical patch Ω^k is the push-forward of a B-spline β_i^k defined on the parameter domain $\hat{\Omega}^k$,

$$b_i^k(\boldsymbol{x}) = \begin{cases} \left(\beta_i^k \circ \left(G^k\right)^{-1}\right)(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \Omega^k \ , \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

The dG-IgA discretization space is spanned by the isogeometric basis functions on both patches, i.e., it considers the subspace

$$\mathcal{V}_h = \operatorname{span}\{b_i^k \ : \ i \in \mathcal{I}^k, \ k = 1, 2\} \subseteq H^1(\mathcal{T}) \ , \tag{3.7}$$

of the broken Sobolev space. The dG-IgA discretization

$$\text{find } u \in \mathcal{V}_h : \quad a(u,v) = F(v) \quad \forall v \in \mathcal{V}_h \tag{3.8}$$

of the Poisson problem (3.1) uses the bilinear form

$$a(u,v) = \sum_{k=1}^{2} a_1^k(u,v) - \sum_{\gamma\in\Gamma} \left(a_{2,1}^\gamma(u,v) + a_{2,2}^\gamma(u,v)\right) + \sum_{\gamma\in\Gamma} a_3^\gamma(u,v) \tag{3.9}$$

with

$$a_1^k(u,v) = \int_{\Omega^k} \nabla u \cdot \nabla v \mathrm{d}\Omega \ , \tag{3.10}$$

$$a_{2,1}^\gamma(u,v) = \int_\gamma \{\nabla u \cdot n\}^\gamma [v]^\gamma d\gamma \ , \ a_{2,2}^\gamma(u,v) = \int_\gamma \{\nabla v \cdot n\}^\gamma [u]^\gamma \mathrm{d}\gamma \ , \tag{3.11}$$

$$a_3^\gamma(u,v) = \int_\gamma \frac{\delta}{h} [u]^\gamma [v]^\gamma \mathrm{d}\gamma \tag{3.12}$$

and the linear form

$$F(v) = \int_\Omega f v \mathrm{d}\Omega \ . \tag{3.13}$$

The bilinear and linear forms coincide with the terms in (3.2). We introduce symbols for the single terms of the bilinear form to facilitate later referencing. The second group of terms $a_{2,1}^\gamma$ and $a_{2,2}^\gamma$ considers normal vectors $n = n_e$ of the interface e, which need to comply with the chosen orientation of the edges (determined by the patch numbering). The last terms a_3^γ in the bilinear form are the penalty terms mentioned before, which involve the sufficiently large parameter δ. They depend on the element size h, i.e. on the length of the knot spans[1].

The discretization (3.8) defines the associated *dG norm*

$$\|u\|_{dG}^2 = \sum_{k=1}^{2} a_1^k(u,u) + \sum_{\gamma\in\Gamma} a_3^\gamma(u,u) \ , \tag{3.14}$$

where in $a_1^k(u,u)$ the gradient of u is restricted to Ω^k, see again [44].

The coefficients u_i^k of the approximate solution

$$u_h = \sum_{k=1}^{2} \sum_{i\in\mathcal{I}^k} u_i^k b_i^k \tag{3.15}$$

are found by solving the linear system $Su = b$ with

$$\begin{aligned} S &= \left(s_{(i,k),(j,\ell)}\right)_{(i,k),(j,\ell)} \ , \\ b &= \left(b_{(j,\ell)}\right)_{(j,\ell)} \ , \\ u &= \left(u_i^k\right)_{(i,k)} \ , \end{aligned}$$

---

[1] For simplicity we consider uniform knots only. If this is not the case then one may consider quasi-uniform knots instead, choosing a parameter that controls the size of all knot spans.

where

$$\begin{aligned} s_{(i,k),(j,\ell)} &= a\left(b_i^k, b_j^\ell\right)\ ,\ i \in \mathcal{I}^k\ ,\ j \in \mathcal{I}^\ell\ ,\ k,\ell = 1,2,\ \text{and} \\ b_{(j,\ell)} &= F\left(b_j^\ell\right)\ ,\ j \in \mathcal{I}^\ell\ ,\ \ell = 1,2\ . \end{aligned}$$

3.4 Integrals along interfaces

Evaluating the forms in (3.9) involves integrals along interfaces, which pose considerable difficulties. We discuss the evaluation of these quantities in more detail, considering again the domain shown in Figure 4.1. As a representative example we shall focus on $a^e_{2,1}$, i.e., we consider the specific edge $\gamma = e$. All observations generalize directly to other terms.

In this situation we obtain

$$\begin{aligned} a^e_{2,1}(u,v) = \int_e & \left(\nabla u^1|_e \cdot n\right) v^1|_e + \left(\nabla u^2|_e \cdot n\right) v^1|_e \\ & - \left(\nabla u^1|_e \cdot n\right) v^2|_e - \left(\nabla u^2|_e \cdot n\right) v^2|_e \mathrm{d}e\ . \end{aligned}$$

The stiffness matrix is a combination of several matrices, each of which is contributed by one of the four forms in (3.9) defining it. In particular we focus on the contribution of $a^e_{2,1}$.

Taking into account that

$$\begin{aligned} b_i^2|_{\Omega^1} &= 0\ ,\quad \nabla b_i^2|_{\Omega^1} = 0 \quad \forall i \in \mathcal{I}^2\ , \\ b_i^1|_{\Omega^2} &= 0\ ,\quad \nabla b_i^1|_{\Omega^2} = 0 \quad \forall i \in \mathcal{I}^1\ , \end{aligned}$$

we find that only the expressions

$$a^e_{2,1}\left(b_i^k, b_j^\ell\right) = (-1)^{\ell+1} \int_e \left(\nabla b_i^k|_{\Omega^k} \cdot n\right) b_j^\ell|_{\Omega^\ell} \mathrm{d}e \tag{3.16}$$

contribute to the element $s_{(i,k),(j,\ell)}$ of the stiffness matrix.

In order to compute these values we use an appropriate numerical quadrature rule, which means that we have to evaluate these products on the interface e. This is no major problem if $k = \ell$ since the integral involves only one trace in this case.

However, the situation is more complicated if $k \neq \ell$ since the (possibly different) parameterizations of the interface need to be taken into account. In the remainder of this section we discuss the evaluation of $a^e_{2,1}\left(b^1_i, b^2_j\right)$ in more detail.

The interface

$$e = G^1([0,1]^2) \cap G^2([0,1]^2) = G^1(1,[0,1]) = G^2(0,[0,1]) \tag{3.17}$$

is parameterized by the restrictions

$$L = G^1|_{(G^1)^{-1}(e)} \text{ and } R = G^2|_{(G^2)^{-1}(e)} \ , \tag{3.18}$$

as sketched in Figure 3.2. These two different representations of the same interface are related by the reparameterizations

$$\lambda : [0,1] \to \{1\} \times [0,1] \tag{3.19}$$

and

$$\varrho : [0,1] \to \{0\} \times [0,1] \tag{3.20}$$

via

$$L \circ \lambda = R \circ \varrho. \tag{3.21}$$

The *construction of suitable reparameterizations* λ and ϱ is the first major problem related to the evaluation of this term. We will discuss it in the next section.

These parameterizations will be used to represent the edge as

$$e = (L \circ \lambda)([0,1]) = (G^1 \circ \lambda)([0,1]) = (G^2 \circ \varrho)([0,1]) = (R \circ \varrho)([0,1]) \ . \tag{3.22}$$

Finally we define $P = L \circ \lambda = R \circ \varrho$ and arrive at

$$\begin{aligned}
& - a^e_{2,1}\left(b^1_i, b^2_j\right) = \int_e \left(\nabla b^1_i(\boldsymbol{x})|_{\Omega^1} \cdot n(\boldsymbol{x})\right) b^2_j(\boldsymbol{x})|_{\Omega^2} \mathrm{d}\boldsymbol{x} \\
&= \int_e \left[\left(\nabla G^1(\boldsymbol{x})\right)^{-1} \nabla \beta^1_i \left((G^1)^{-1}(\boldsymbol{x})\right)|_{\Omega^1} \cdot n(\boldsymbol{x})\right] \beta^2_j \left((G^2)^{-1}(\boldsymbol{x})\right)|_{\Omega^2} \mathrm{d}\boldsymbol{x} \\
&= \int_0^1 \left[\left(\nabla G^1\left(P(t)\right)\right)^{-1} \nabla \beta^1_i \left(L^{-1}\left(P(t)\right)\right) \cdot n\left(P(t)\right)\right] \beta^2_j \left(R^{-1}\left(P(t)\right)\right) \|\dot{P}(t)\| \mathrm{d}t \\
&= \int_0^1 \left[\left(\nabla G^1\left(P(t)\right)\right)^{-1} \nabla \beta^1_i \left(\lambda(t)\right) \cdot n\left(P(t)\right)\right] \beta^2_j \left(\varrho(t)\right) \|\dot{P}(t)\| \mathrm{d}t \ .
\end{aligned}$$

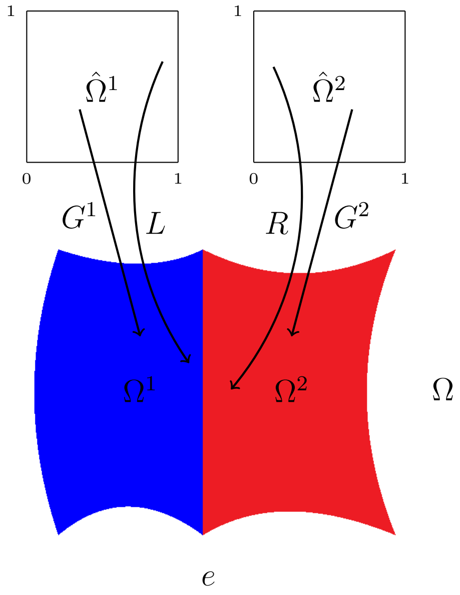


Figure 3.2: Multi-patch domain with two patches Ω^1, Ω^2 and geometry mappings G^1, G^2. The mappings L and R are the parameterizations of the interface e, i.e., the restriction of the geometry mappings to its preimage.

The integral in the last line is evaluated by a quadrature rule. However, the *choice of the quadrature rule*, which is the second major problem related to the evaluation of this term, is nontrivial and will be discussed further in Section 3.6. In fact, the choice of the rules needs to take the different knots of the functions β_i^1, β_j^2, λ and ϱ into account. While one will generally choose the same knots for λ and ϱ, the knots of β_i^1 and β_j^2 are subject to a non-linear transformation and cannot be assumed to be identical.

3.5 Finding the reparameterizations

It is quite common in the literature to assume matching parameterizations or almost matching ones, see [14, p. 4148], [15, p. 87], [44, 45, 46, 64]. In this situation, the choice of the reparameterizations λ and ϱ is trivial, as they are simply linear parameterizations (possibly reversing the orientation) of the preimages of the interface in the parameter domains. However, the restriction to matching

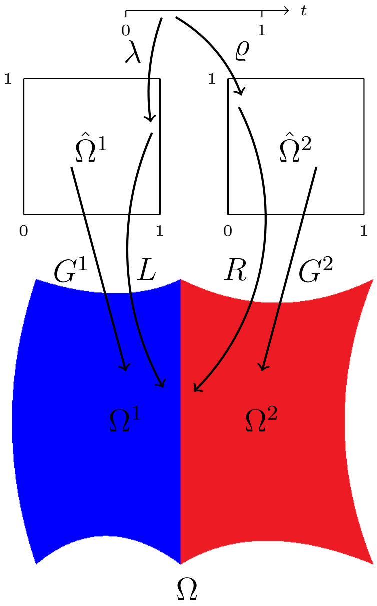


Figure 3.3: Multi-patch domain with geometry maps G^1 and G^2, their restrictions L and R to the preimages of the interface and reparameterizations λ and ϱ

parameterizations poses constraints on the construction of multi-patch parameterizations, making it essentially impossible to parameterize the individual patches separately. This fact motivates us to study the *non-matching* case.

More precisely, we consider situations where the condition (3.21) cannot be satisfied by considering linear reparameterizations λ and ϱ. Clearly, the condition does not determine λ and ϱ uniquely. We fix one of the mappings, say λ, and compute the remaining one, ϱ. Figure 3.3 visualizes the relations between the mappings.

The unknown mapping ϱ satisfies $\varrho = R^{-1} \circ L \circ \lambda$. We compute it by least-squares approximation of point samples, as follows:

1. For a given number N of samples, we evaluate

$$\varrho_i = R^{-1} \circ L \circ \lambda \left(\frac{i}{N} \right) \tag{3.23}$$

   by performing the closest point computations

$$\varrho_i = \underset{\boldsymbol{\xi} \in \{0\} \times [0,1]}{\operatorname{argmin}} \left\| L \circ \lambda \left(\frac{i}{N} \right) - R(\boldsymbol{\xi}) \right\|, \; i = 0, \ldots, N, \tag{3.24}$$

where $\|\cdot\|$ is the Euclidean norm. This formulation also applies to the case of geometrically inexact interfaces (cf. [29, 30]).

2. We choose a suitable spline space (e.g. linear, quadratic or cubic splines with a few uniformly distributed inner knots) and find the control points $c_j \in \{0\} \times [0,1]$ of the associated B-splines N_j, $j = 1, \ldots, m$, by solving the linear least-squares problem

$$\sum_{i=1}^{N} \left(\sum_{j=1}^{m} c_j N_j \left(\frac{i}{N} \right) - \varrho_i \right)^2 \longrightarrow \min, \tag{3.25}$$

cf. [20]. The influence of the choice of the spline space for ϱ will be discussed later in Section 3.7. The given reparameterization λ is chosen as a linear polynomial.

We will refer to the case where at least one of the mappings λ and ϱ is different from the identity as non-matching parameterizations at the interface.

3.6 Numerical integration

The evaluation of

$$a_{2,1}^{e}\left(b_i^1, b_j^2\right) = \int_0^1 \left[\left(\nabla G^1\left(P(t)\right)\right)^{-1} \nabla \beta_i^1\left(\lambda(t)\right) \cdot n\left(P(t)\right) \right] \beta_j^2\left(\varrho(t)\right) \|\dot{P}(t)\| \mathrm{d}t \ . \tag{3.26}$$

requires integration with respect to the parameter t, which varies in the parameter domain $[0,1]$. This is done by applying numerical quadrature and we present several strategies for doing so.

3.6.1 Gauss quadrature with exact splitting

Gauss quadrature can be applied to segments of analytic functions. Consequently, we split the parameter domain $[0,1]$ into segments (separated by junctions) where the integrand satisfies this requirement. Three types of junctions arise:

- the inverse images $\lambda^{-1}(\kappa_i^1)$ of the knots κ_i^1 that were used to define the B-splines β_j^1,

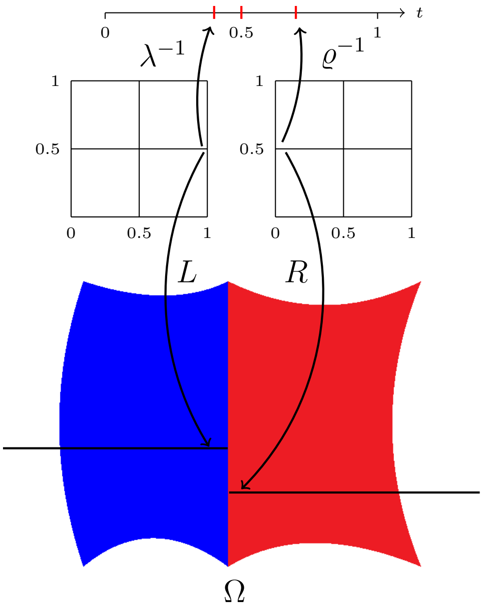


Figure 3.4: Exact splitting of a knot span and application of a quadrature rule to each subsegment

- the inverse images $\varrho^{-1}(\kappa_i^2)$ that were used to define the B-splines β_j^2, and
- the knots τ_i that were used to define the B-splines N_j in (3.25).

These types are visualized in Figure 3.4.

Consequently we perform Gauss quadrature with exact splitting by applying the following algorithm:

- Compute all junction points (all three types) in $[0, 1]$,
- sort the junction points, subdivide the domain into segments accordingly,
- subdivide the resulting segments if they are too long,
- apply a Gauss quadrature rule to each segment and sum up the contributions.

As a disadvantage, the inversion of λ and ϱ is costly and has to be done with high accuracy, as the sorting depends on it. Furthermore, the method may result in many segments of varying lengths.

We use Gauss quadrature with $p+1$ nodes per element (which exactly integrates polynomials of degree $2p+1$), where p is the degree used for defining the dG-IgA discretization, cf. [49].

3.6.2 Gauss quadrature with uniform splitting

A computationally simpler approach is to use uniform subdivision, as follows:

- Split the domain $[0,1]$ uniformly into M segments, where M is a multiple of the number of knot spans used to define the B-splines N_j in (3.25),
- apply a Gauss quadrature rule to each segment and sum up the contributions.

As we shall see later, it is mandatory to use large values of M in order to reach the desired level of accuracy. This is due to the fact that the junctions of the first two types listed in the previous section may still be located within the segments obtained by uniform splitting. On the other hand, the use of uniform refinement also creates many small segments that could be merged into larger ones without compromising the accuracy. This can be exploited by using adaptive quadrature.

3.6.3 Adaptive Gauss quadrature

We recall the main idea of adaptive quadrature, cf. [24]. In order to evaluate the integral

$$I = \int_a^b f(x)\mathrm{d}x \tag{3.27}$$

of an integrable function f over an interval $[a,b]$ adaptively one computes two different estimates I_1 and I_2 of I by using two different integration methods. One assumes that one of these estimates, say I_1, is more accurate than the other. Next, one computes the relative distance between I_1 and I_2 taking into account a given (or chosen) tolerance tol, e.g. machine precision. If the difference is small enough, one chooses I_1 as the value of the integral $\int_a^b f(x)\mathrm{d}x$. If this is not the case one splits the interval $[a,b]$ into two subintervals,

$$[a,b] = [a,m] \cup [m,b] \quad , \quad \text{where} \quad m = \frac{a+b}{2},$$

and evaluates I by summing up the two contributions. This means that one applies the procedure of computing two different estimates and checking their relative difference to both subintervals. Adaptive quadrature is therefore a recursive procedure, which is summarized in Algorithm 1.

Algorithm 1 Adaptive Quadrature: Basic routine.

AdaptQuad(f, a, b, tol)

```
1: Input: $f$ , $a$ , $b$ , tol where $f$ is an integrable function, $a$ and $b$ are the interval boundaries and tol is a given tolerance
2: Choose knots $u_i$ and weights $w_i$ , $i = 1, \dots, n$ .
3: Compute $I_1 = \sum_{i=1}^{n} w_i f(u_i)$ .
4: Choose knots $\tilde{u}_i$ and weights $\tilde{w}_i$ , $i = 1, \dots, m$ .
5: Compute $I_2 = \sum_{i=1}^{m} \tilde{w}_i f(\tilde{u}_i)$ .
6: if $|I_1 - I_2| \leq \text{tol} \cdot |I_1|$ then
7:     Return $I_1$
8: else
9:     Return
       $$\text{AdaptQuad}\left(f, a, \frac{a+b}{2}, \text{tol}\right) + \text{AdaptQuad}\left(f, \frac{a+b}{2}, b, \text{tol}\right) \; .$$
10: end if
```

Note that the stopping criterion has to be chosen with care and in fact line 6 in the algorithm is a slight oversimplification of it. See [24] for further information.

We apply this procedure to the knot spans that were used to define the B-splines N_j in (3.25). Therefore we choose I_1 as a Gauss quadrature rule with $p+1$ quadrature knots, where again p is the degree of the basis functions in the dG-IgA discretization space. For the computation of I_2 we split the interval manually into two halves, apply a a Gauss quadrature rule of the same order on both halves, and sum up. The tolerance tol is set to machine precision.

As an advantage, adaptive quadrature can be performed without inverting the reparameterizations. Moreover, it avoids the oversegmentation problem that was present for the previous approach. We observed experimentally that the adaptive procedure accurately detects the junction points and subdivides the domain

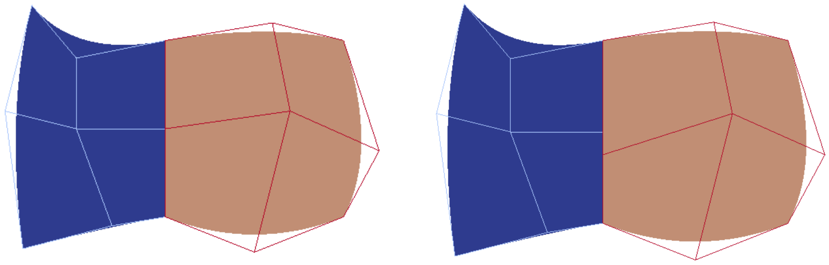


Figure 3.5: Patch and its control net. Left: matching parameterizations at the interface. Right: non-matching parameterizations at the interface.

accordingly. Clearly, the implementation is more costly and requires a recursive algorithm.

3.7 Numerical results

We examine the performance of the quadrature methods presented in Section 3.6 as well as the influence of the accuracy of the reparameterization. All experiments were performed using G+Smo[2], an object-oriented C++ IgA library named "Geometry + Simulation Modules".

3.7.1 Reference results

As a reference we will first show the convergence plot of the solution of the Poisson equation in the case of matching parameterizations, i.e. for $\lambda = \varrho = \mathrm{id}$. In this case we can restrict ourselves to a simple quadrature rule. There is no need for using more elaborate versions of numerical integration. Furthermore, since $\lambda = \varrho = \mathrm{id}$, we do not need to consider the influence of the quality of the reparameterization. More precisely, we consider the two-patch domain with biquadratic matching interface parameterizations shown in Figure 3.5, left.

Figure 3.6 demonstrates the convergence behaviour of the numerical solutions that were obtained for various values of the element size h that was used to define the dG-IgA discretization. We consider a problem with a known solution and

---

[2] G+Smo: `gs.jku.at`

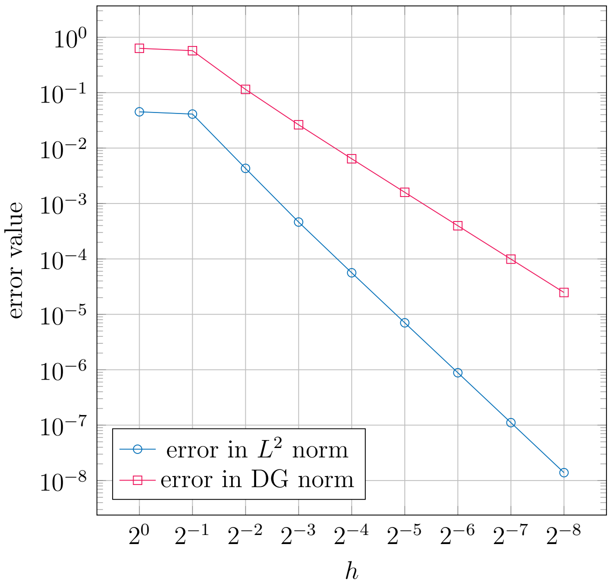


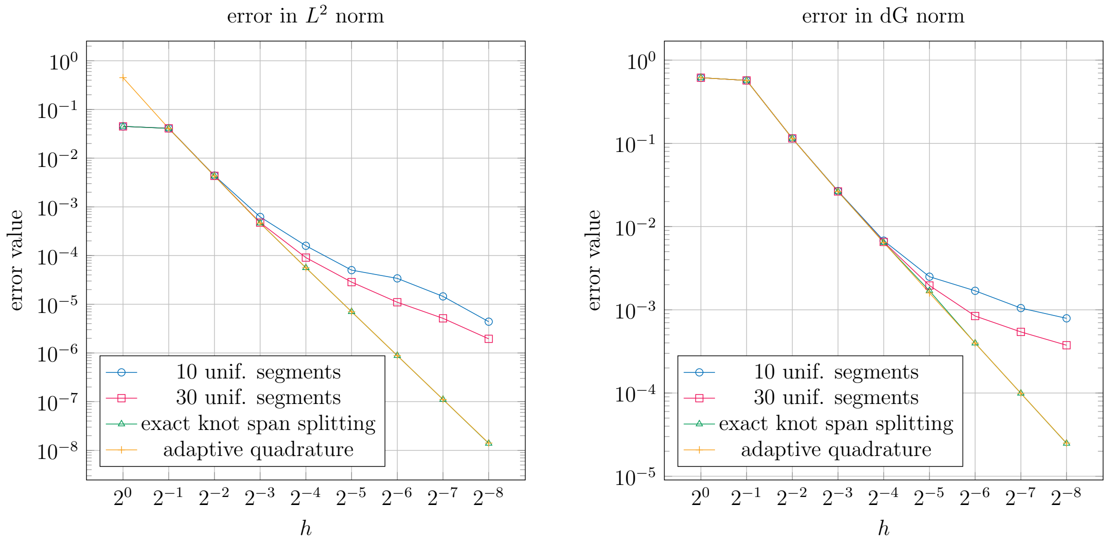
Figure 3.6: Matching parameterizations at the interface, convergence behaviour of error in different norms: L^2 norm (blue curve), dG norm (red curve).

measure the error as the difference to it. The quadrature method we used is Gauss quadrature with three quadrature knots. A convergence rate of three for the L^2 error and of two for the dG error is clearly visible. This is in accordance with the theoretical predictions, see [3, 15].

3.7.2 Influence of the quadrature rule

We now consider a different parameterization of the same computational domain, with non-matching parameterizations of the interface, see Figure 3.5, right. Again we use biquadratic patches. Now we need to use a more complicated integration technique, and we consider the three approaches that were described in Section 3.6.

Figure 3.7, left and right, visualizes the convergence behaviour measured in the L^2 and dG norms respectively. Each plot contains four curves, corresponding to four different numerical quadrature techniques. More precisely, we consider Gauss quadrature with exact splitting (green), Gauss quadrature with uniform splitting into 10 (blue) and into 30 (red) segments, and adaptive Gauss quadrature


error in L^2 norm

error in dG norm

Figure 3.7: Influence of the quadrature rule. Left: Convergence behavior of the error in L^2 norm. Right: Convergence behavior of the error in dG norm. Blue and red curves: 10 and 30 uniform segments per t-knot span. Green curves: exact splitting of the knot spans. Yellow curves: adaptive quadrature. Note that the yellow curve coincides with the green one for smaller values of h. Exact representation of the reparameterizations λ and ϱ.

(yellow). We observe that the first and the last method perform better than the results based on uniform splitting and they achieve the optimal convergence rates (compare with Figure 3.6). In particular we note that using uniform quadrature leads to a reduced order of convergence for smaller mesh sizes h. Even the use of a very fine but uniform segmentation (30 (red) instead of 10 (blue) segments) does not improve this significantly.

Based on these observations we decided to use solely adaptive and exact Gauss quadrature in the remaining example.

3.7.3 Influence of the reparameterization

Next we analyse the influence of the quality of the representation of the reparameterization. Consider again the parameterization of the domain in Figure 3.5, right, with non-matching parameterizations of the interface. We compare three different choices of the reparameterizations λ and ϱ.

For the first reparameterization, which generates the results represented by the blue curve in Figure 3.8, we choose polynomials λ and ϱ such that the equation $L \circ \lambda = R \circ \varrho$ is exactly satisfied. In this case it was possible to find such polynomials, due to the particular construction of the example. However, this would be impossible in general and it is used here to generate a reference result.

The second and third reparameterizations (red and green curves) were obtained using the Algorithm from Section 3.5 to find ϱ, while λ was chosen as a linear polynomial. The second reparameterization uses a linear spline with 8 segments and has an L^2 error of $1.3 \cdot 10^{-2}$, and the third reparameterization uses a cubic spline with 4 segments and has an L^2 error of $3.1 \cdot 10^{-15}$.

Figure 3.8 compares the errors in the L^2 (left) and dG norms (right) for the three reparameterizations. We observe that using a high quality reparameterization is essential for the convergence of the method. Depending on the accuracy of the reparameterization, h-refinement only works until it reaches a critical mesh size, where further refinement does not have any effect.

The plots show the results obtained by using adaptive Gauss quadrature. The exact method gives virtually identical results.

3.7.4 Comparison of exact and adaptive quadrature

We perform an experimental comparison of the computational complexity of exact and adaptive quadrature for the domain in Figure 3.5, right.

First we demonstrate the effect of using adaptive quadrature, by showing the automatically created splitting points in Figure 3.9. We used an accuracy of 10^{-6} instead of machine precision for this picture to obtain a clearer image. Both patches were uniformly refined into 4×4 elements by knot insertion. The mappings λ and ϱ are cubic splines on $[0, 1]$ with four knot spans of equal length. Their knots τ_i coincide with the inverse images $\lambda^{-1}(\kappa_i^1)$, as the first mapping is simply the identity. The adaptive quadrature, which is applied to the knot spans $[\tau_i, \tau_{i+1}]$, thus creates additional splitting points around the inverse images $\varrho^{-1}(\kappa_i^2)$, as shown in the Figure. In this particular case, only one splitting point (at 0.5625) is created

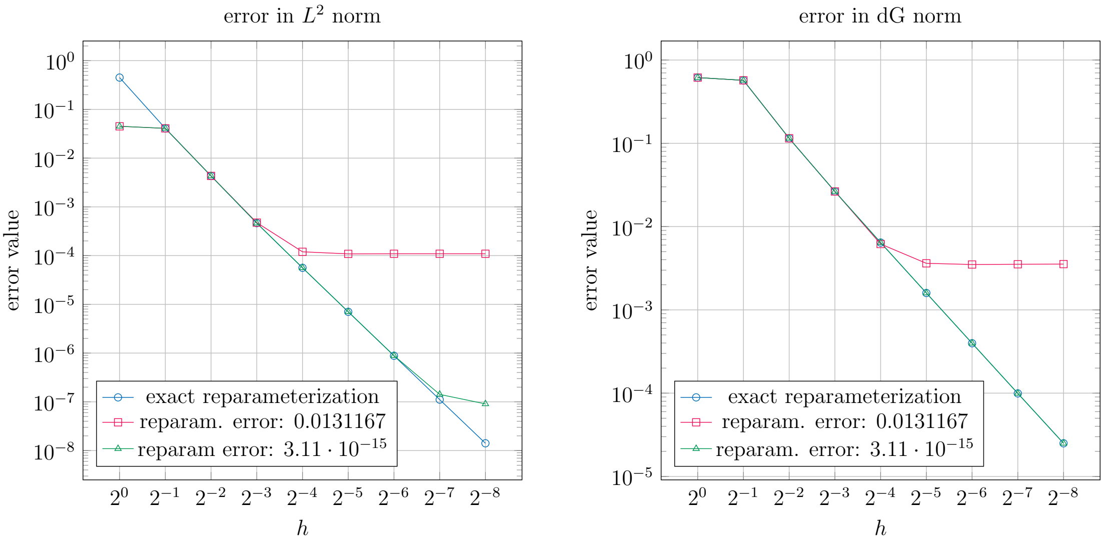


Figure 3.8: Influence of the reparameterization. Adaptive quadrature on interface integrals. Left: Convergence behaviour of the error in L^2 norm. Right: Convergence behaviour of the error in dG norm. Blue curves: Exact representation of λ and ϱ. Red curves: Approximation error of $\varrho \approx 0.0131167$. Green curves: Approximation error of $\varrho \approx 3.10616 \cdot 10^{-15}$

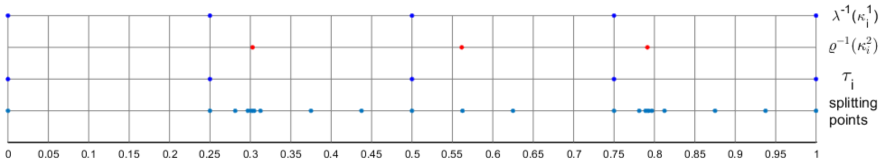


Figure 3.9: Splitting points created by adaptive quadrature - see text for details.

near $\varrho^{-1}(\kappa_2^2) = 0.5615$ since this suffices to reach the desired accuracy.

These results indicate that, unlike uniform Gauss quadrature, adaptive quadrature avoids over-segmentation of the integration domains. Still, it splits the knot spans more often than exact Gauss quadrature, which also results in a higher number of quadrature knots and thus evaluations.

In order to analyze this effect, Figure 3.10 compares the number of evaluations (i.e., quadrature knots) used by exact and adaptive Gauss quadrature for increasing numbers of elements. In addition, we also show the number of root finding operations (which are more expensive than evaluations) needed to compute the splitting points of exact Gauss quadrature. Clearly, adaptive quadrature requires more evaluations than exact splitting. However, for sufficiently fine discretizations, the number of evaluations in the interior of the patches dominates the total effort.

3.8 Summary

We used a simple model problem to investigate the complications that arise from using non-matching interface parameterizations within the framework of Isogeometric Analysis on a multi-patch domain, using discontinuous Galerkin techniques to couple terms across the interfaces. The discretized discontinious Galerkin bilinear form contains integrals of products of isogeometric functions, defined on neighboring patches, on the common interface. The interface parameterization is crucial for their evaluation. More precisely, we studied two particular problems. Firstly, we explored the use of reparameterizations to identify pairs of associated points on the common interface. This was found to be useful for correctly evaluating products of basis functions. Secondly, we addressed the construction of a suitable procedure for numerical integration, in order to compute the value of

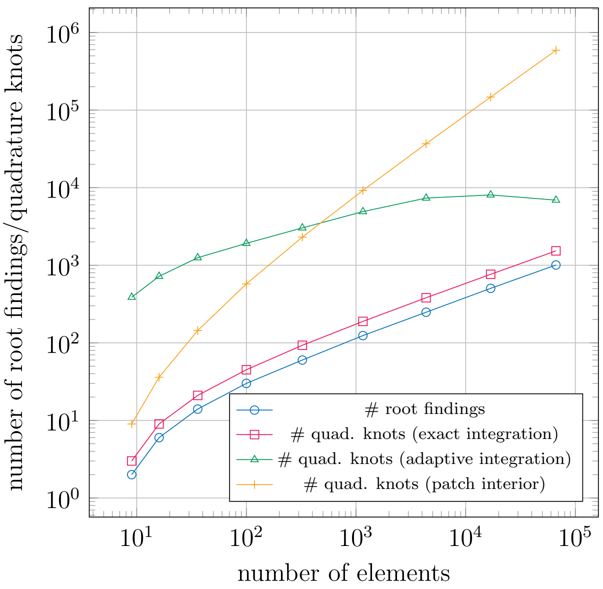


Figure 3.10: Number of quadrature knots and root finding operations needed by exact and adaptive quadrature for increasingly finer discretizations.

integrals of the named products. Using exact quadrature combined with splitting the knot spans accordingly and applying adaptive quadrature lead to similar error decays of the approximate solution. As demonstrated in our numerical experiments, both problems are important for ensuring the optimal rate of convergence for the numerical simulation. If the reparameterization or the quadrature rule is not accurate enough, the approximate solution will not converge with the optimal convergence rate. Finally we investigated the computational effort of exact and adaptive quadrature. Although the adaptive technique required more function evaluations, the overall effort on the patch interior outweighed the effort of the computations on the interface.

Chapter 4

Approximately $\mathcal{C}^1$-smooth Isogeometric Functions

In this chapter, we follow a different approach to coupling isogemetric functions across patch interfaces on multi-patch domains. In contrast to working with broken Sobolev spaces and adapting the variational formulation to couple the solution, this chapter is devoted to the construction of approximately smooth isogeometric test functions on the whole domain. In turn, changing the variational form to a patch-wise formulation is not necessary. The coupling of isogeometric functions across patch interfaces recently attracted substantial interest:

- $\mathcal{C}^0$-coupling of isogeometric functions is easily realizable by identifying the coefficients of neighboring basis functions along an interface as one degree of freedom. However, for higher order problems, $\mathcal{C}^0$-continuity of the test functions is not sufficient.

- For such differential equations, $\mathcal{C}^1$-smooth test functions are required. Their construction is considerably more complicated. Recent results rely on the relation between geometrical smoothness of a graph surface and parameterical smoothness of the underlying functions [26, 40]. However, such constructions are based on certain assumptions about the parameterization of the underlying domain, which are needed to ensure sufficient flexibility of the resulting discretizations. For instance, in [33, 40], the authors use to bilinear or bilinear-like parameterizations.

- In [13, 35, 36] it is shown that non-bilinear domains almost always can be reparameterized in order to match the necessary assumptions.
- Also [11] considers more general domains. However, the authors compute the coefficients of the isogeometric basis functions numerically and not symbolically as it is done in [33, 40].
- In a similar fashion to [33, 40], the papers [37, 38, 39] are devoted to the construction of $\mathcal{C}^2$-smooth isogeometric functions, which are required for sixth-order problems like the triharmonic equation. Again, bilinearly parameterized domains are considered.

In order to avoid the limitation to bilinear-like parameterizations, we relax the construction by considering approximate instead of exact $\mathcal{C}^1$-smoothness of isogeometric functions on multi-patch domains. This enables us to generate function spaces on general (not bilinear-like) domains. Our construction is based on suitably chosen bilinear forms. More precisely, we explore two different forms and obtain two different function spaces. Starting from globally $\mathcal{C}^0$-smooth functions, we provide bounds on the gradient jump of the corresponding approximately $\mathcal{C}^1$-smooth isogeometric functions.

The rest of this chapter is organized as follows: In Section 1 we provide the necessary notation and we introduce two different bilinear forms $\mathcal{B}_1$ and $\mathcal{B}_2$. Section 2 describes the construction of a space of approximately $\mathcal{C}^1$-smooth isogeometric functions based on $\mathcal{B}_1$ and investigates its advantages and drawbacks, in particular we will find that the resulting space suffers from not containing trivially smooth functions. In order to overcome this deficiency, in Section 3 we carry out an analogous construction based on the slightly modified bilinear form $\mathcal{B}_2$. Section 4 is devoted to numerical experiments concerning the approximation power and the dimension of the space. In particular, we will provide experiments that suggest that the functions we construct are smooth enough to solve fourth-order problems like the biharmonic equation. In Section 5 we conclude this chapter and point out possible future research.

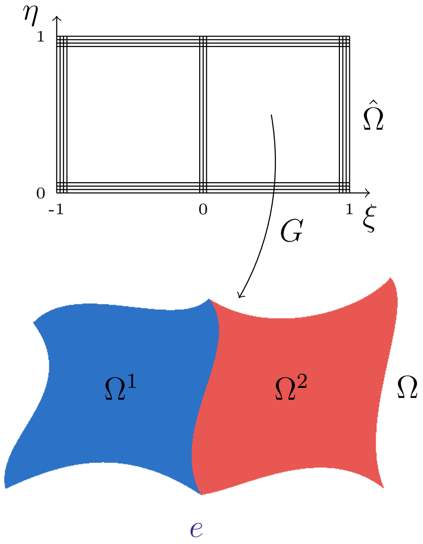


Figure 4.1: Two patch domain Ω parameterized by a bicubic geometry map G. The knot vectors are given by $[-1,-1,-1,-1,0,0,0,1,1,1,1] \times [0,0,0,0,1,1,1,1]$.

4.1 Preliminaries

Consider again a planar two-patch domain $\bar{\Omega} = \bar{\Omega}^1 \cup \bar{\Omega}^2 \subseteq \mathbb{R}^2$ with interface e between the single patches Ω^1 and Ω^2, as depicted in Figure 4.1. Although we will still work in an isogeometric setting, the notation will slightly change. In contrast to the previous chapter, here the domain will be parameterized on a single parameter domain $\hat{\Omega} = [-1,1] \times [0,1]$ by only one tensor-product B-spline geometry map G of the form

$$G : \hat{\Omega} \to \Omega : (\xi, \eta) \mapsto \sum_{i \in \mathcal{I}} P_i \beta_i(\xi, \eta). \tag{4.1}$$

Therefore, no superscript is required to indicate that patch index and we will no longer use k and ℓ for that purpose. Instead, $P_i \in \mathbb{R}^2$ are control points and β_i are tensor-product B-splines of bidegree (p_1, p_2) with global index set $\mathcal{I}$, defined by open quasi-uniform knot vectors with maximal knot span sizes h_1, h_2 in ξ and η-direction, respectively. By h we denote the maximal element length $h = \max\{h_1, h_2\}$. The multiplicities of the inner knots do not exceed $p-1$, except for the knot 0 with respect to ξ, which appears p times. The simplest instance of the knot configuration is visualized in Figure 4.1. The patch interface is

$$e = G(\{0\} \times [0,1]).$$

The associated isogeometric basis functions

$$b_i(\boldsymbol{x}) = \left(\beta_i \circ G^{-1}\right)(\boldsymbol{x}),\ i \in \mathcal{I}, \tag{4.2}$$

are collected in the vector

$$\boldsymbol{b}(\boldsymbol{x}) = (b_i(\boldsymbol{x}))_{i\in\mathcal{I}} \tag{4.3}$$

and span the isogeometric discretization space

$$\mathcal{V}_h = \operatorname{span}\{b_i : i \in \mathcal{I}\} \subseteq \mathcal{C}^0(\Omega). \tag{4.4}$$

The last inclusion holds due to the choice of the knot vector. This is another difference to the situation in Chapter 3, where we worked with broken Sobolev spaces.

Finally we recall the definition of the jump operator

$$[f] = f^1|_e - f^2|_e,$$

which is defined for any function $f \in L^2(\Omega)$ with

$$f^1 = f|_{\Omega^1} \in H^1(\Omega^1), \quad f^2 = f|_{\Omega^2} \in H^1(\Omega^2).$$

For functions with $f^1 \in H^2(\Omega^1)$ and $f^2 \in H^2(\Omega^2)$ the gradient jump is defined analogously.

We will use two different bilinear forms in order to construct approximately $\mathcal{C}^1$-smooth isogeometric functions on Ω. The first one is given by

$$\mathcal{B}_1 : \mathcal{V}_h \times \mathcal{V}_h \to \mathbb{R} : (f,g) \mapsto \varepsilon \int_\Omega f(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x} - \int_e [\nabla f(\boldsymbol{x})]^T[\nabla g(\boldsymbol{x})]d\boldsymbol{x} \tag{4.5}$$

and depends on a positive parameter ε. The second one takes the form

$$\mathcal{B}_2 : \mathcal{V}_h \times \mathcal{V}_h \to \mathbb{R} : (f,g) \mapsto \int_e [\nabla f(\boldsymbol{x})]^T[\nabla g(\boldsymbol{x})]d\boldsymbol{x}. \tag{4.6}$$

More precisely, our aim is to construct isogeometric functions with a bounded gradient jump $\|[\nabla f]\|_{L^2(e)}$, and the bilinear forms $\mathcal{B}_1$ and $\mathcal{B}_2$ are designed with this objective in mind. For the first one, a suitable value of ε has to be chosen in advance. It controls the magnitude of the bound. The spaces of approximately smooth isogeometric functions obtained by using $\mathcal{B}_1$ and $\mathcal{B}_2$ have different properties, although the constructions themselves are quite similar.

4.2 Results for $\mathcal{B}_1$

We show that the bilinear form $\mathcal{B}_1$ yields a mesh-size independent bound on the gradient jump. However, we will also see that this space lacks optimal approximation power.

4.2.1 Construction of approximately smooth functions

We consider functions $f, g \in \mathcal{V}_h$ with

$$f(\boldsymbol{x}) = u^T \boldsymbol{b}(\boldsymbol{x}), g(\boldsymbol{x}) = v^T \boldsymbol{b}(\boldsymbol{x}), \tag{4.7}$$

with coefficient vectors $u, v \in \mathbb{R}^{|\mathcal{I}|}$. Consequently, $\mathcal{B}_1(f, g)$ can be rewritten in matrix-vector-form as

$$\mathcal{B}_1(f, g) = u^T(\varepsilon M - Q)v, \tag{4.8}$$

where

$$M = (m_{i,j})_{i,j\in\mathcal{I}} \text{ with } m_{i,j} = \int_\Omega b_i(\boldsymbol{x}) b_j(\boldsymbol{x}) d\boldsymbol{x} \tag{4.9}$$

and

$$Q = (q_{i,j})_{i,j\in\mathcal{I}} \text{ with } q_{i,j} = \int_e [\nabla b_i(\boldsymbol{x})]^T [\nabla b_j(\boldsymbol{x})] d\boldsymbol{x}, \tag{4.10}$$

as confirmed by a short computation:

$$\begin{aligned}\int_\Omega f(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x} &= \int_\Omega \left(\sum_{i\in\mathcal{I}} u_i b_i(\boldsymbol{x})\right)\left(\sum_{j\in\mathcal{I}} v_j b_j(\boldsymbol{x})\right) d\boldsymbol{x} \\ &= \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I}} u_i v_j \int_\Omega b_i(\boldsymbol{x}) b_j(\boldsymbol{x}) d\boldsymbol{x}\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{I}} u_i v_j m_{i,j} \\
&= u^T M v.
\end{aligned}
$$

An analogous computation confirms $\int_e [\nabla f(\boldsymbol{x})]^T [\nabla g(\boldsymbol{x})] d\boldsymbol{x} = u^T Q v$. The matrices M and Q are symmetric positive semi-definite, as

$$u^T M u = \|f\|^2_{L^2(\Omega)} \geq 0 \tag{4.11}$$

and

$$u^T Q u = \|[\nabla f]\|^2_{L^2(e)} \geq 0. \tag{4.12}$$

Now let $0 \leq \lambda^1 \leq \ldots \leq \lambda^n$ be the non-negative eigenvalues of $\varepsilon M - Q$ in ascending order and let $c^1, \ldots, c^n$ be the corresponding eigenvectors, $n \leq |\mathcal{I}|$. The eigenvectors satisfy

$$(c^k)^T c^\ell = 0 \text{ and } (c^k)^T c^k = 1 \text{ for } k \neq \ell, 1 \leq k, \ell \leq n, \tag{4.13}$$

possibly after performing the Gram-Schmidt orthonormalization, if multiple eigenvalues are present.

We define

$$
\begin{aligned}
G_h^{1,\varepsilon} &:= \operatorname{span}\left\{ \sum_{i\in\mathcal{I}} c_i^k b_i(\boldsymbol{x}) : k = 1, \ldots, n \right\} \\
&= \left\{ \sum_{i\in\mathcal{I}} d_i b_i(\boldsymbol{x}) : d \in \operatorname{span}\left\{c^1, \ldots c^n\right\} \right\}.
\end{aligned}
\tag{4.14}
$$

as the *space of approximately* $\mathcal{C}^1$*-smooth isogeometric functions.*

4.2.2 Properties of the function space

By construction, $G_h^{1,\varepsilon}$ is a linear space. As an immediate consequence from its definition, we obtain a mesh-size independent bound on the gradient jump of functions in this space, i.e., the bound does not weaken as we refine the mesh. This is different from standard inverse estimated for isogeometric functions [3].

We denote by $\boldsymbol{C} \in \mathbb{R}^{|\mathcal{I}|\times n}$ the matrix containing the eigenvectors $c^1, \ldots, c^n$ as column vectors.

Proposition 1. *The gradient jump of any function* $f = (\boldsymbol{C}d)^T\boldsymbol{b} \in G_h^{1,\varepsilon}$ *with* $d \in \mathbb{R}^n$ *can be bounded by*

$$\|[\nabla f]\|^2_{L^2(e)} \leq \varepsilon \|f\|^2_{L^2(\Omega)} \tag{4.15}$$

Proof. We use (4.11) and (4.12) and obtain

$$\begin{aligned}
\varepsilon\|f\|^2_{L^2(\Omega)} - \|[\nabla f]\|^2_{L^2(e)} &= (\boldsymbol{C}d)^T(\varepsilon M - Q)(\boldsymbol{C}d) \\
&= d^T\boldsymbol{C}^T(\varepsilon M - Q)\boldsymbol{C}d \\
&= d^T\mathrm{diag}\left(\lambda^1, \ldots, \lambda^n\right)d \\
&= \lambda^1 d_1^2 + \ldots + \lambda^n d_n^2 \geq 0,
\end{aligned}$$

where the last inequality holds because we only consider non-negative eigenvalues λ_i. □

A further estimation of the gradient jump is possible. Let $\lambda_{\max}(M)$ denote the maximal eigenvalue of the mass matrix M. Since M is symmetric it holds that

$$x^T M x \leq x^T(\lambda_{\max}(M) \cdot I)x = \lambda_{\max}(M)\|x\|_2^2 \quad \forall x \in \mathbb{R}^{|\mathcal{I}|}, \tag{4.16}$$

if the entries of M are bounded. Recalling

$$\|f\|^2_{L^2(\Omega)} = (\boldsymbol{C}d)^T M(\boldsymbol{C}d),$$

it follows that

$$\|f\|^2_{L^2(\Omega)} \leq \lambda_{\max}(M)\|\boldsymbol{C}d\|_2^2.$$

This bound is mesh-size dependent, as the eigenvalue $\lambda_{\max}(M)$ depends on h.

The space $G_h^{1,\varepsilon}$ based on $\mathcal{B}_1$ does not necessarily contain the trivially smooth isogeometric functions b_i^k with

$$\nabla b_i^k|_e = \vec{0}.$$

We will refer to these functions as *off-interface basis functions.* Since they are constantly zero across the interface e, their gradient jumps across e are zero as well. However, their coefficient vectors with respect to the basis $\{b_i : \imath \in \mathcal{I}\}$, which are the canonical unit vectors in $\mathbb{R}^{|\mathcal{I}|}$, are not necessarily eigenvectors of $\varepsilon M - Q$. As we shall see in Section 4.4, the functions in $G_h^{1,\varepsilon}$ do not possess the same approximation power as the full space of isogeometric functions.

4.3 Results for $\mathcal{B}_2$

We study another bilinear form, which is designed to compensate the drawbacks of the first approach, in order to ensure the existence of trivially smooth functions in the resulting space of approximately smooth isogeometric functions. However, in this case we cannot expect to obtain an estimate of $\|[\nabla f]\|_{L^2(e)}$ that is independent of the mesh size.

4.3.1 Construction of approximately smooth functions

The modified space $\hat{G}_h^{1,\varepsilon}$ is constructed analogously to the procedure described in Section 4.2.1. Recall that $\mathcal{B}_2(f,f)$ can equivalently be written as

$$\mathcal{B}_2(f,f) = u^T Q u$$

for

$$f = u^T \boldsymbol{b} \in \mathcal{V}_h, \quad u \in \mathbb{R}^{|\mathcal{I}|}.$$

As explained before, the matrix Q is symmetric positive semi-definite. We choose a positive value ε. Let $\hat{\lambda}^1 \leq \ldots \leq \hat{\lambda}^{\hat{n}} \leq \varepsilon$ be the eigenvalues of Q that are bounded by ε and let $\hat{c}^1, \ldots, \hat{c}^{\hat{n}}$ be the corresponding orthonormalized eigenvectors, $\hat{n} \leq |\mathcal{I}|$. We define

$$\hat{G}_h^{1,\varepsilon} := \operatorname{span}\left\{ \sum_{i \in \mathcal{I}} \hat{c}_i^k b_i : k = 1, \ldots, \hat{n} \right\}. \tag{4.17}$$

4.3.2 Properties of the function space

Again, by construction, $\hat{G}_h^{1,\varepsilon}$ is a linear space. Moreover, all trivially smooth isogeometric functions f, i.e., the off-interface basis functions as well as constant and linear functions (which are contained in the space of isogeometric functions, due to use of the isoparametric principle), fulfill

$$\mathcal{B}(f,f) = 0.$$

Since the matrix Q is symmetric positive semi-definite, this implies that the coefficient vector of f is an element of the kernel of Q. Consequently, the corresponding coefficient vector is an eigenvector to the eigenvalue 0 of Q. Since we set $\varepsilon > 0$, all elements in the kernel will also be elements of $\hat{G}_h^{1,\varepsilon}$. This is independent of the mesh size h. As we will see, the inclusion of these functions in $\hat{G}_h^{1,\varepsilon}$ is important to achieve optimal convergence.

Subsequently, we bound the gradient jump of functions in $\hat{G}_h^{1,\varepsilon}$. We denote by $\hat{\boldsymbol{C}} \in \mathbb{R}^{|\mathcal{I}|\times\hat{n}}$ the matrix containing the eigenvectors $\hat{c}^1, \dots, \hat{c}^{\hat{n}}$ of Q as column vectors. Let $f \in \hat{G}_h^{1,\varepsilon}$, i.e. we set

$$f(\boldsymbol{x}) = (\hat{\boldsymbol{C}}d)^T \boldsymbol{b}(\boldsymbol{x}) \tag{4.18}$$

with $d \in \mathbb{R}^n$.

Theorem 2. *Let the knot vectors that define the B-splines $\{\beta_i\}_{i\in\mathcal{I}}$ be quasi-uniform. Then all functions $f \in \hat{G}_h^{1,\varepsilon}$ satisfy*

$$\|[\nabla f]\|^2_{L^2(e)} \leq \varepsilon \frac{C}{h^2} \|f\|^2_{L^2(\Omega)} \tag{4.19}$$

for a constant C that depends on the maximal spline degree p and the geometry mapping G, but not on the maximal mesh size h.

Proof. Let $f = (\hat{\boldsymbol{C}}d)^T \boldsymbol{b} \in G_h^{1,\varepsilon}$ as denoted above. Then we have

$$\begin{aligned}\|[\nabla f]\|^2_{L^2(e)} &= (\hat{\boldsymbol{C}}d)^T Q (\hat{\boldsymbol{C}}d) = d^T \hat{\boldsymbol{C}}^T Q \hat{\boldsymbol{C}} d \\ &= d^T \mathrm{diag}\left(\hat{\lambda}^1, \dots, \hat{\lambda}^{\hat{n}}\right) d \\ &= \hat{\lambda}^1 d_1^2 + \ldots + \hat{\lambda}^{\hat{n}} d_{\hat{n}}^2 \\ &\leq \varepsilon \sum_{i=1}^{\hat{n}} d_i^2 = \varepsilon \|d\|_2^2 = \varepsilon \|\hat{\boldsymbol{C}}d\|_2^2,\end{aligned}$$

where the last equality holds because $\hat{\boldsymbol{C}}$ is an orthogonal matrix.

Next, we use the stability of tensor-product B-spline bases $\{\beta_i\}_{i\in\mathcal{I}}$ [55] with stability constant D_p^2, where $p = \max\{p_1, p_2\}$, and get

$$\|[\nabla f]\|^2_{L^2(e)} \leq \varepsilon \|\hat{\boldsymbol{C}}d\|_2^2 \leq \varepsilon D_p^4 \frac{1}{h^2} \left\| \sum_{i\in\mathcal{I}} (\hat{\boldsymbol{C}}d)_i \beta_i \right\|^2_{L^2(\hat{\Omega})}. \tag{4.20}$$

We rewrite β_i in terms of the push-forward $b_i \circ G$ and obtain

$$\|[\nabla f]\|^2_{L^2(e)} \leq \varepsilon D_p^4 \frac{1}{h^2} \left\| \sum_{i\in\mathcal{I}} (\hat{\boldsymbol{C}}d)_i \, (b_i \circ G) \right\|^2_{L^2(\hat{\Omega})}, \tag{4.21}$$

which again can be rewritten and summarized as

$$\begin{aligned} \|[\nabla f]\|^2_{L^2(e)} &\leq \varepsilon D_p^4 \frac{1}{h^2} \left\| \left(\sum_{i\in\mathcal{I}} (\hat{\boldsymbol{C}}d)_i b_i \right) \circ G \right\|^2_{L^2(\hat{\Omega})} \\ &= \varepsilon D_p^4 \frac{1}{h^2} \left\| f \circ G \right\|^2_{L^2(\hat{\Omega})}. \end{aligned} \tag{4.22}$$

Now we transform the integral $\int_{\hat{\Omega}} (f \circ G)^2$ on $\hat{\Omega}$ to an integral on Ω, which yields

$$\|[\nabla f]\|^2_{L^2(e)} \leq \varepsilon D_p^4 \frac{1}{h^2} \| \det \nabla(G)^{-1} \|_{L^\infty(\Omega)} \|f\|^2_{L^2(\Omega)}. \tag{4.23}$$

Finally we set

$$C(p, G) = D_p^4 \cdot \left\| \det \nabla(G)^{-1} \right\|_{L^\infty(\Omega)}.$$

This concludes the proof. □

This result resembles standard inverse inequalities for isogeometric functions, which can be found in [3], apart from the factor ε, which is chosen in advance, and the power of h. For a sufficiently smooth function f, standard inverse inequalities lead to an estimate of the form

$$\|\nabla f\|^2_{L^2(e)} \leq \frac{C}{h^3} \|f\|^2_{L^2(\Omega^i)},$$

where C is a positive constant and Ω^i is one of the two patches adjacent to e. If we choose $\varepsilon \in \mathcal{O}(h^2)$ for the construction of $\hat{G}_h^{1,\varepsilon}$, we can eliminate the mesh-size dependence in the bound of the gradient jump. However, smaller values of ε lead to fewer functions in $G_h^{1,\varepsilon}$, which we will discuss in the following section.

4.4 Numerical examples

We consider least squares approximation, the Poisson problem as well as the biharmonic equation on a two-patch domain. In this context we are interested in the approximation power of $G_h^{1,\varepsilon}$ and $\hat{G}_h^{1,\varepsilon}$. Furthermore we will study the number of interface basis functions under uniform h-refinement.

4.4.1 Approximation power

Second order problems – such as the Poisson equation – require weakly differentiable ansatz functions on the whole domain. Thus, in an isogeometric discretization, smoothness of a higher order than $\mathcal{C}^0$-smoothness is not required. However, we include these examples to illustrate that we maintain full approximation power in these cases, although we use less basis functions than in the standard tensor-product basis. Solving the biharmonic equation on the contrary requires second order differentiable test functions. Our examples suggest that the functions in $\hat{G}_h^{1,\varepsilon}$ are sufficiently smooth.

Throughout the remainder of this section, all errors are measured patch-wisely and then summed up, e.g. we refer to

$$\|f_{\text{approx}}|_{\Omega^1} - f_{\text{exact}}|_{\Omega^1}\|_{H^1(\Omega^1)} + \|f_{\text{approx}}|_{\Omega^2} - f_{\text{exact}}|_{\Omega^2}\|_{H^1(\Omega^2)}$$

as the H^1 error and to

$$\|f_{\text{approx}}|_{\Omega^1} - f_{\text{exact}}|_{\Omega^1}\|_{H^2(\Omega^1)} + \|f_{\text{approx}}|_{\Omega^2} - f_{\text{exact}}|_{\Omega^2}\|_{H^2(\Omega^2)}$$

as the H^2 error of f_{approx}. The patch-wise splitting is not necessary for the L^2 error, as $G_h^{1,\varepsilon} \subseteq L^2(\Omega)$ and $\hat{G}_h^{1,\varepsilon} \subseteq L^2(\Omega)$.

L^2 approximation

We start with an example that identifies the limitations of the space $G_h^{1,\varepsilon}$, which is based on the bilinear form $\mathcal{B}_1$. Figure 4.2 shows the function

$$f_{\text{exact}}(x, y) = 3xy \exp(-x) \sin(\pi y), \tag{4.24}$$

which we approximate on a two-patch domain by functions in $G_h^{1,\varepsilon}$. The domain coincides with the one shown in Figure 4.1. We solve the constrained least squares fitting problem

$$\min_{f \in G_h^{1,\varepsilon}} \|f - f_{\text{exact}}\|^2_{L^2(\Omega)}.$$

The parameter ε was set to 0.5. The relative L^2 and H^1 errors are depicted in the left plot of Figure 4.3. After some refinement steps, no significant reduction of

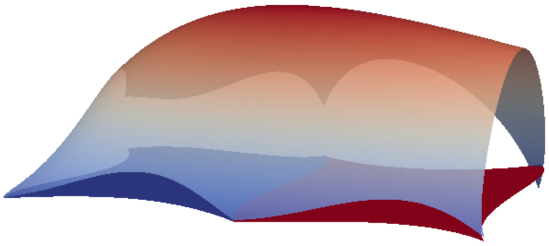


Figure 4.2: Bicubically parameterized domain (see Fig. 4.1) and transparent plot of the exact solution $3xy\exp(-x)\sin(\pi y)$.

the error is achieved. Considering the distribution of the error values in the last refinement step, shown in Figure 4.3, right, we note that the largest errors occur close to the interface and in the back corners. This is a possible indicator that the corresponding corner basis functions are not present in $G_h^{1,\varepsilon}$.

Consequently, we consider only the space $\hat{G}_h^{1,\varepsilon}$ based on the bilinear form $\mathcal{B}_2$. The following experiment shows that - in contrast to the previous approach - the functions in $\hat{G}_h^{1,\varepsilon}$ maintain the full approximation power.

Again, we choose $\varepsilon = 0.5$ and approximate the same function (4.24) on the same domain as before. We use a uniform h-refinement strategy. The relative L^2 and H^1 error values and the respective convergence rates are shown in Figure 4.4, left and right, respectively. A comparison with the reference slopes shows that the functions in $\hat{G}_h^{1,\varepsilon}$ maintain the optimal convergence rates of $p+1$ and p for the L^2 and the H^1 error, respectively.

At the finest level of refinement we used 8,840 (9,111) basis functions of degree 3 (4) with mesh size 2^{-6}. Note that this number of basis functions is slightly less than the number of original tensor-product B-splines, which is 8,978 (9,248) for degree 3 (4).

The bottom plot in Figure 4.4 depicts the solution using 8,840 basis functions of degree 3 as a patch-wise plot with added flat shading. These effects highlight the smoothness of the solution across the curved interface.

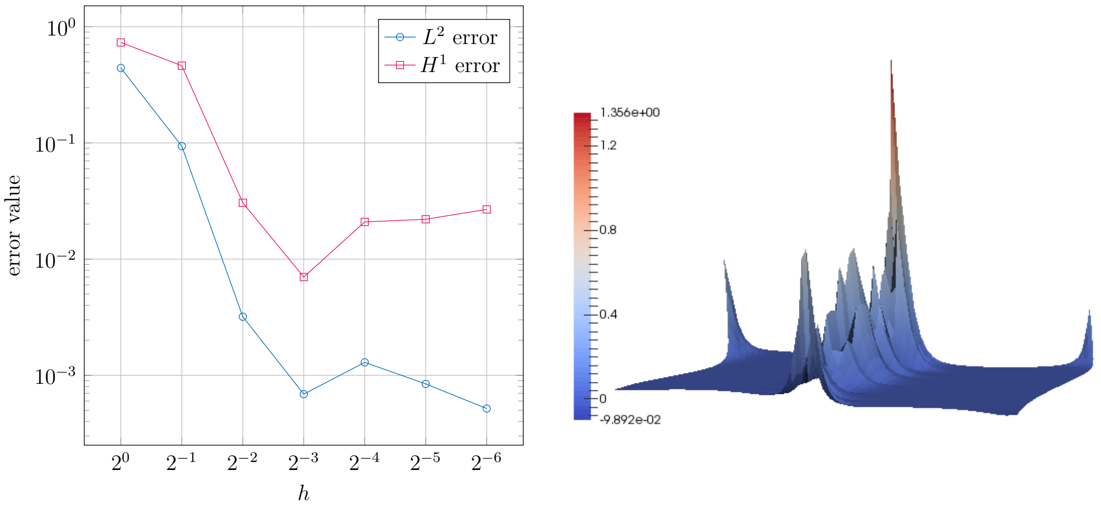


Figure 4.3: Least squares approximation with functions in $G_h^{1,\varepsilon}$ with $\varepsilon = 0.5$. Left: relative L^2 and H^1 error for the approximate solution of degree 3. Right: L^2 error values at the finest discretization step (scaled by factor 100).

Poisson problem

Solving the Poisson problem leads to very similar results. It is given by

$$\text{Find } u : \begin{cases} -\Delta u & = f \text{ on } \Omega \\ u & = 0 \text{ on } \partial\Omega \end{cases},$$

see problem (3.1). We solve this problem by means of a continuous Galerkin method, i.e., we solve the discretized weak form

$$\text{find } u \in \hat{G}_{h,0}^{1,\varepsilon} \text{ such that } \int_\Omega \nabla u(\boldsymbol{x}) \nabla v(\boldsymbol{x}) d\boldsymbol{x} = \int_\Omega f(\boldsymbol{x}) v(\boldsymbol{x}) d\boldsymbol{x} \quad \forall v \in \hat{G}_{h,0}^{1,\varepsilon}, \tag{4.25}$$

where

$$\hat{G}_{h,0}^{1,\varepsilon} = \{u \in G_h^{1,\varepsilon} : u|_{\partial\Omega} = 0\}.$$

The zero Dirichlet boundary conditions are imposed strongly in the test function space. Again we set the threshold ε to 0.5. The exact solution is given by

$$\begin{aligned} u(x,y) = 40&(0.25x + 0.75 - y)(-0.25x + 1.25 - y) \\ &(-0.25x + 0.25 - y)(0.25x - 0.25 - y)\sin(0.5\pi x). \end{aligned}$$

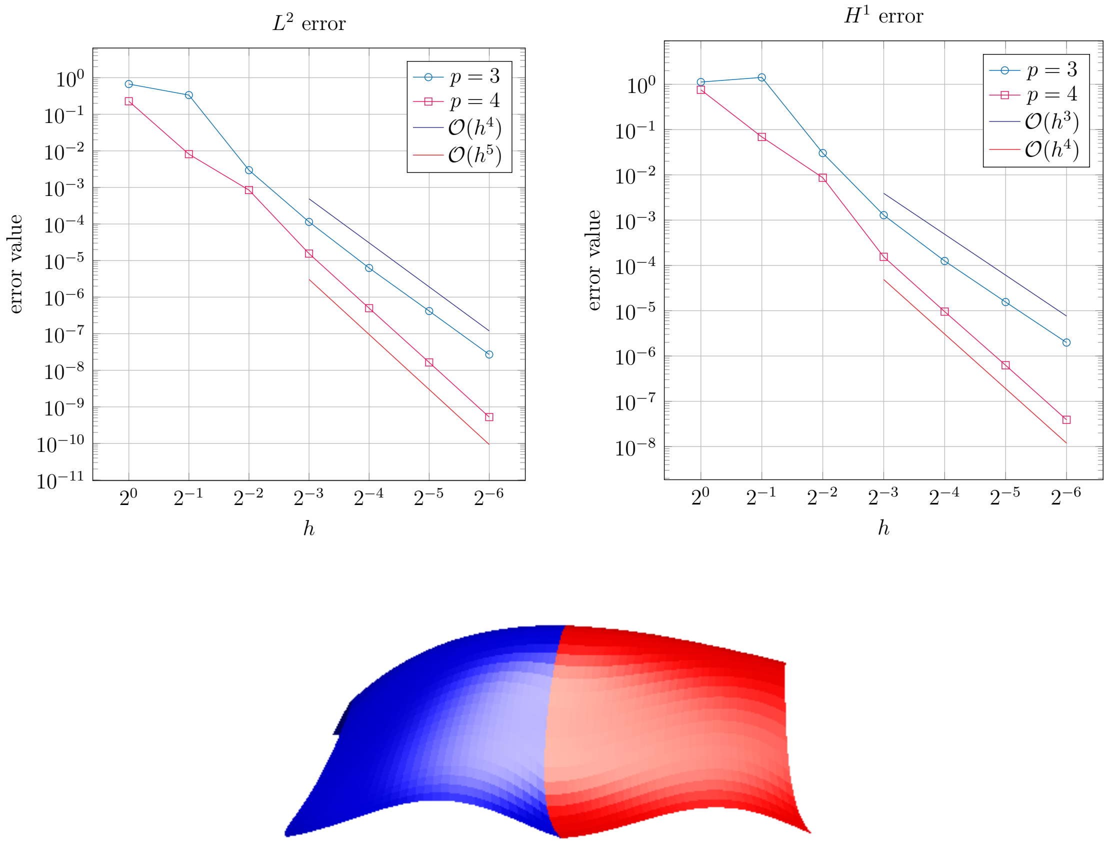


Figure 4.4: Least squares approximation with functions in $\hat{G}_h^{1,\varepsilon}$ with $\varepsilon = 0.5$. Relative L^2 (top left) and H^1 (top right) error of the solution to the fitting problem on the bi-cubic domain, see Figure 4.1. Bottom: Patch-wise representation of the solution with 8,840 basis functions of degree three for $\varepsilon = 0.5$ with flat shading.

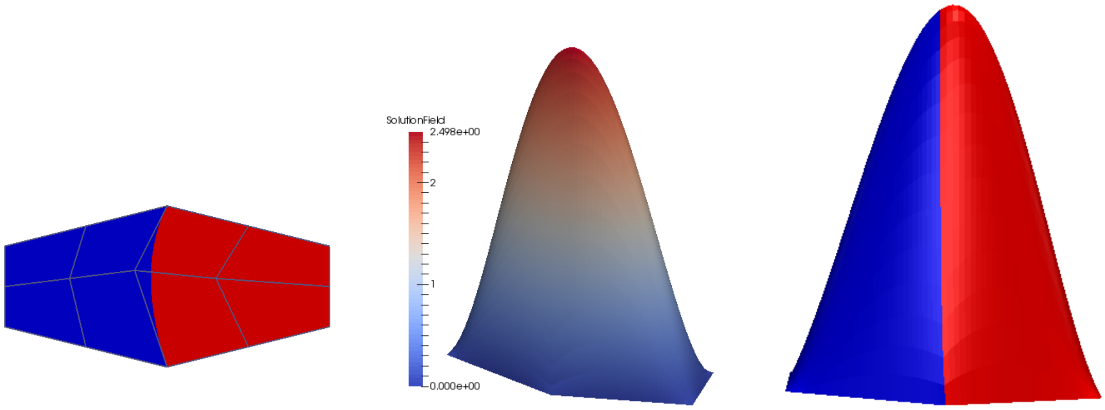


Figure 4.5: Poisson problem (4.25). Left: Domain Ω and its control net. Middle: Approximate solution with 2,048 basis functions of degree 2. Right: Patch-wise plot of the solution with flat shading.

The physical domain Ω consists of two patches with a curved interface, see Figure 4.5, left. It is biquadratically parameterized. Figure 4.5, middle, shows its solution for 2,048 basis functions of degree 2 with element size 2^{-5}, the patch-wise plot of the solution, right, emphasizes the location of the interface. The flat shading in the right plot shows that the solution is smooth in this area.

The behavior of the relative L^2 and the H^1 error are shown in Figure 4.6, left and right, respectively. We see that in both cases and for the tested degrees two, three and four of the test functions we realize optimal convergence rates. This is consistent with the L^2 approximation results.

Biharmonic equation

The examples shown previously did not require $\mathcal{C}^1$-smooth basis functions. Consequently, the approximately smooth functions we presented did not exhibit any advantage over standard $\mathcal{C}^0$-smooth isogeometric functions which can be constructed by identifying the corresponding degrees of freedom along the interface, except for the fact that we used slightly less basis functions. We now consider a fourth-order equation, where the bilinear form governing the weak formulation cannot be evaluated for only $\mathcal{C}^0$-smooth functions and thus, smoother functions are mandatory. The following examples demonstrate that - depending on the value of ε that con-

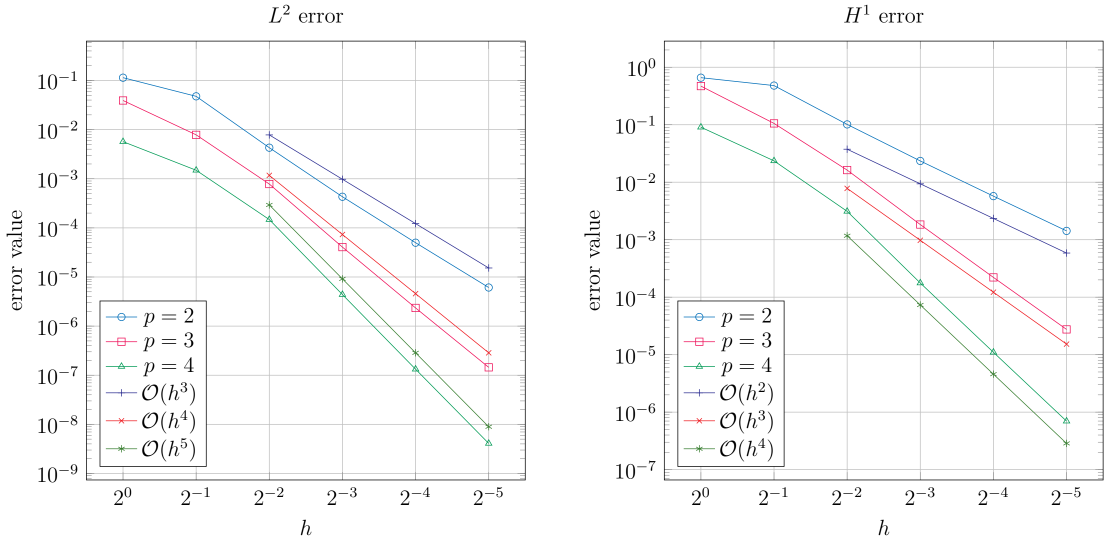


Figure 4.6: Poisson problem (4.25). Relative L^2 (left) and H^1 error (right) of the approximate solution with basis functions of different degrees in $\hat{G}^{1,\varepsilon}_{h,0}$ for $\varepsilon = 0.5$.

trols the magnitude of the jump - approximately smooth functions are suitable for solving such a higher-order problem and even to maintain full approximation power in the L^2, H^1 and H^2 error.

We consider the discretized weak form:

$$\text{Find } u \in \hat{G}^{1,\varepsilon}_{h,\mathbf{0}} \text{ such that } \int_\Omega \Delta u(\boldsymbol{x})\Delta v(\boldsymbol{x}) d(\boldsymbol{x}) = \int_\Omega f(\boldsymbol{x})v(\boldsymbol{x}) \quad \forall v \in \hat{G}^{1,\varepsilon}_{h,\mathbf{0}}, \tag{4.26}$$

where

$$\hat{G}^{1,\varepsilon}_{h,\mathbf{0}} = \{u \in G^{1,\varepsilon}_{h} : u|_{\partial\Omega} = (\nabla u \cdot n)|_{\partial\Omega} = 0\},$$

of the biharmonic equation

$$\begin{aligned} \Delta^2 u &= f \text{ on } \Omega \\ u = \frac{\partial u}{\partial n} &= 0 \text{ on } \partial\Omega. \end{aligned} \tag{4.27}$$

Again, we impose the boundary conditions strongly in the test function space and solve (4.26) by means of the Galerkin method. The right-hand side f is obtained from the exact solution $(1 - \cos(2\pi x))(1 - \cos(2\pi y))$. The domain Ω is a square, which is split into two patches with a curved interface, see Figure 4.7, left. Figure

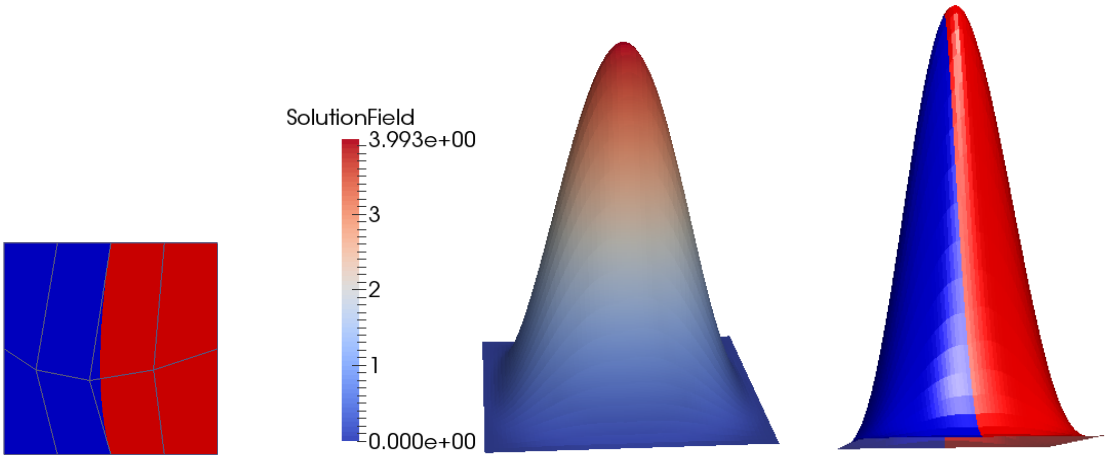


Figure 4.7: Biharmonic equation (4.26): Domain with its control net (left), approximate solution with 2.101 basis functions of degree four in $\hat{G}_h^{1,\varepsilon}$ with $\varepsilon = h^2$ (middle) and patch-wise plot of the approximate solution with flat shading (right).

4.7, middle, depicts the solution for 2101 basis functions of degree four and element size $h = 2^{-5}$ and $\varepsilon = h^2$. The flat shading demonstrates the smoothness of the solution across the interface in the patch-wise plot on the right.

We consider the decay of the relative error for different degrees of the basis functions, starting with degree $p = 3$. The plots in Figure 4.8 show that the optimal approximation order with respect to the L^2 (top left), H^1 (top right) and H^2 (bottom) norm is reached for $\varepsilon = C \cdot h^k$ for $k \leq 2$, but not for $k = 3$. This effect is best visible in the L^2 error.

The situation is slightly different for $p = 4$. Here, the optimal approximation order with respect to the L^2 (top left), H^1 (top right) and H^2 (bottom) norm is reached for $\varepsilon = C \cdot h^k$ for $k = 2, 3$, but neither for $k \leq 1$ nor for $k \geq 4$, see Figure 4.9. Finally, the optimal approximation order for $p = 5$ with respect to the L^2 (top left), H^1 (top right) and H^2 (bottom) norm is reached for $\varepsilon = C \cdot h^k$ for $k = 3$, but neither for $k \leq 2$ nor for $k \geq 4$, as shown in Figure 4.10.

On the one hand, a higher power of h and thus a smaller value of ε results in smoother, but at the same time in fewer basis functions, hence in a loss of approximation power. On the other hand, while choosing a larger value of ε increases the dimension of $\hat{G}_h^{1,\varepsilon}$, the resulting discretizations are not smooth enough

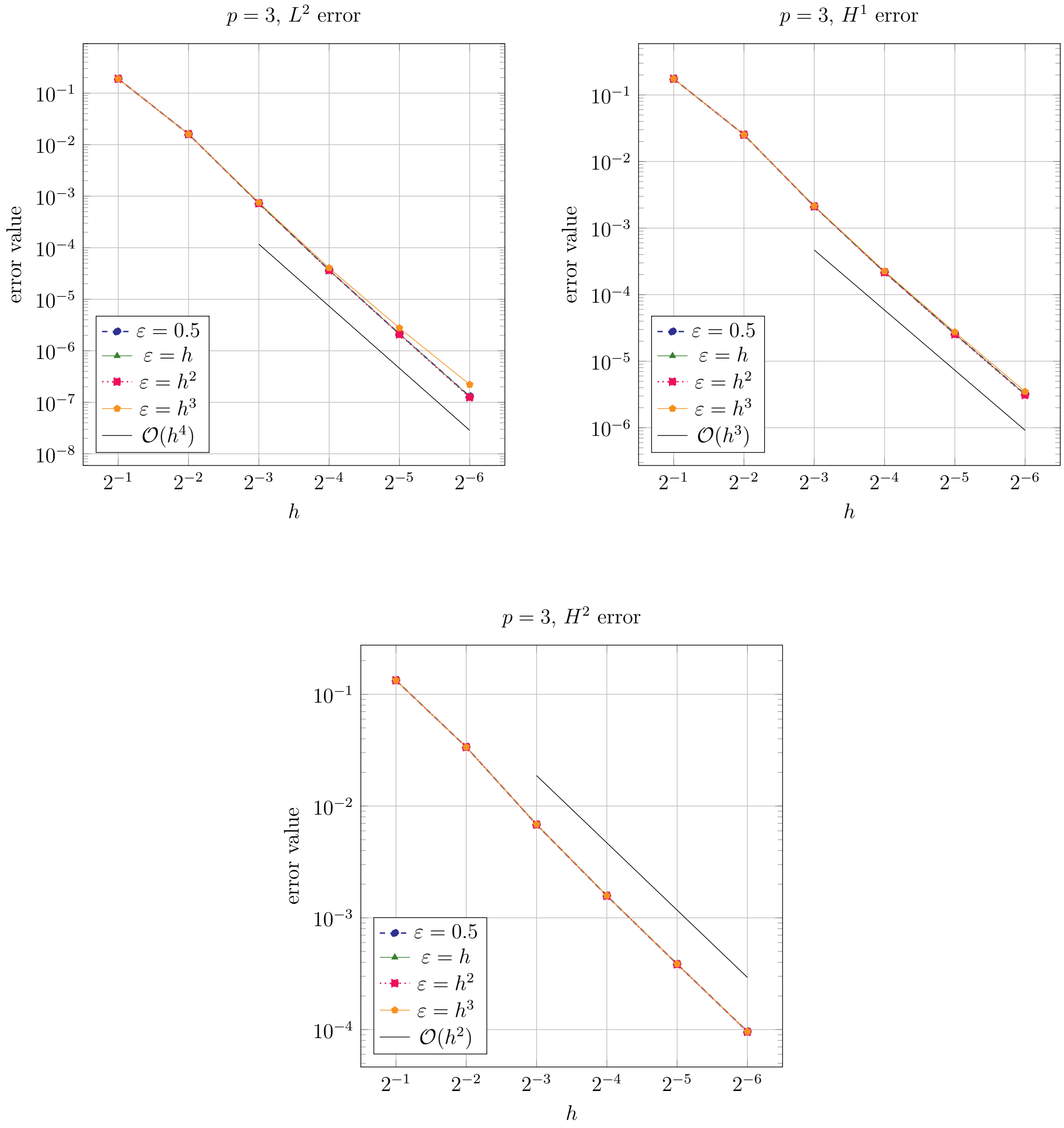


Figure 4.8: Biharmonic equation (4.26): Relative L^2 (top left), H^1 (top right) and H^2 (bottom) errors of the approximate solution for basis functions of degree three in $\hat{G}^{1,\varepsilon}_{h,\mathbf{0}}$ for four choices of ε.

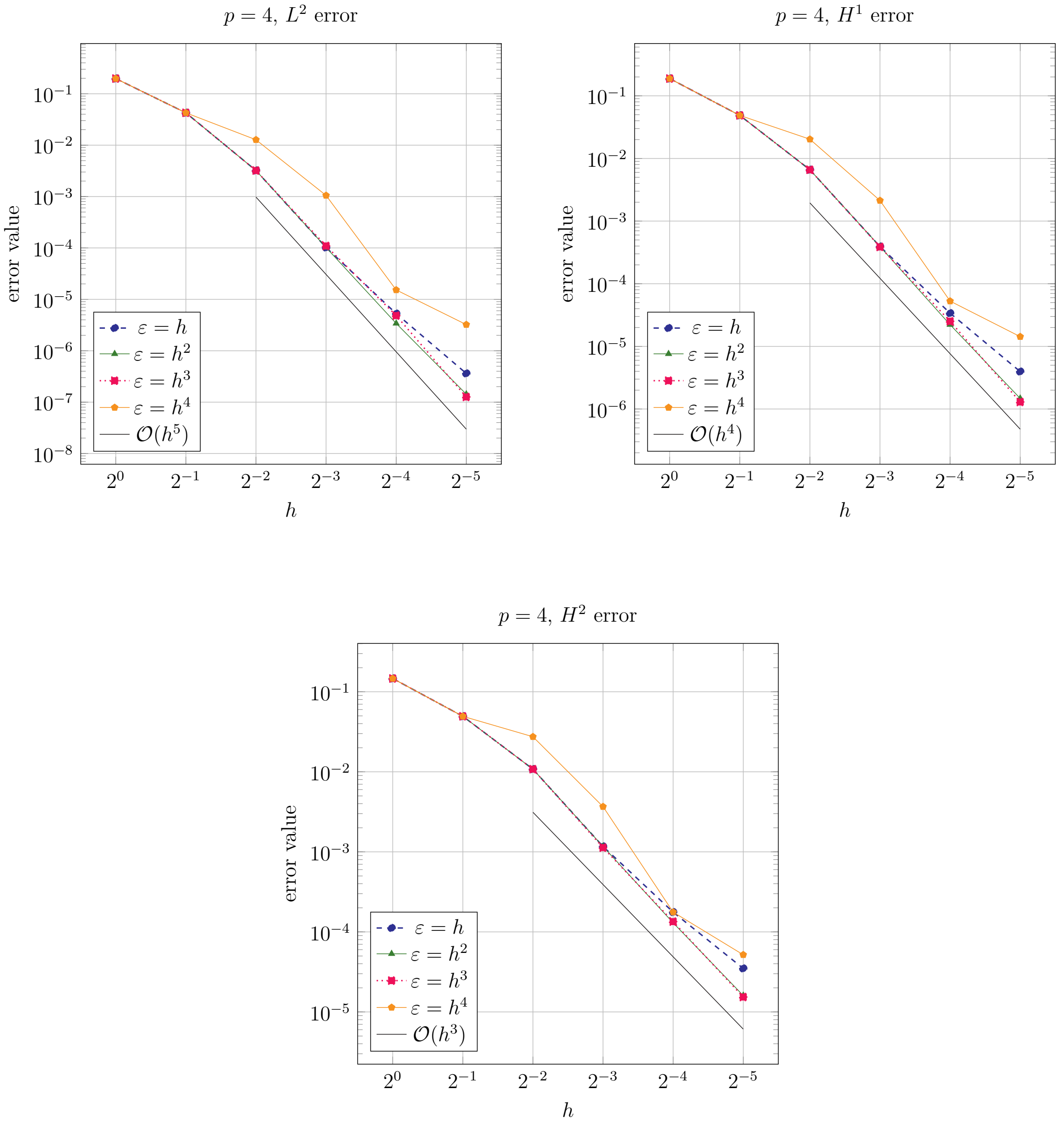


Figure 4.9: Biharmonic equation (4.26): Relative L^2 (top left), H^1 (top right) and H^2 (bottom) errors of the approximate solution for basis functions of degree four in $\hat{G}^{1,\varepsilon}_{h,\mathbf{0}}$ for four choices of ε.

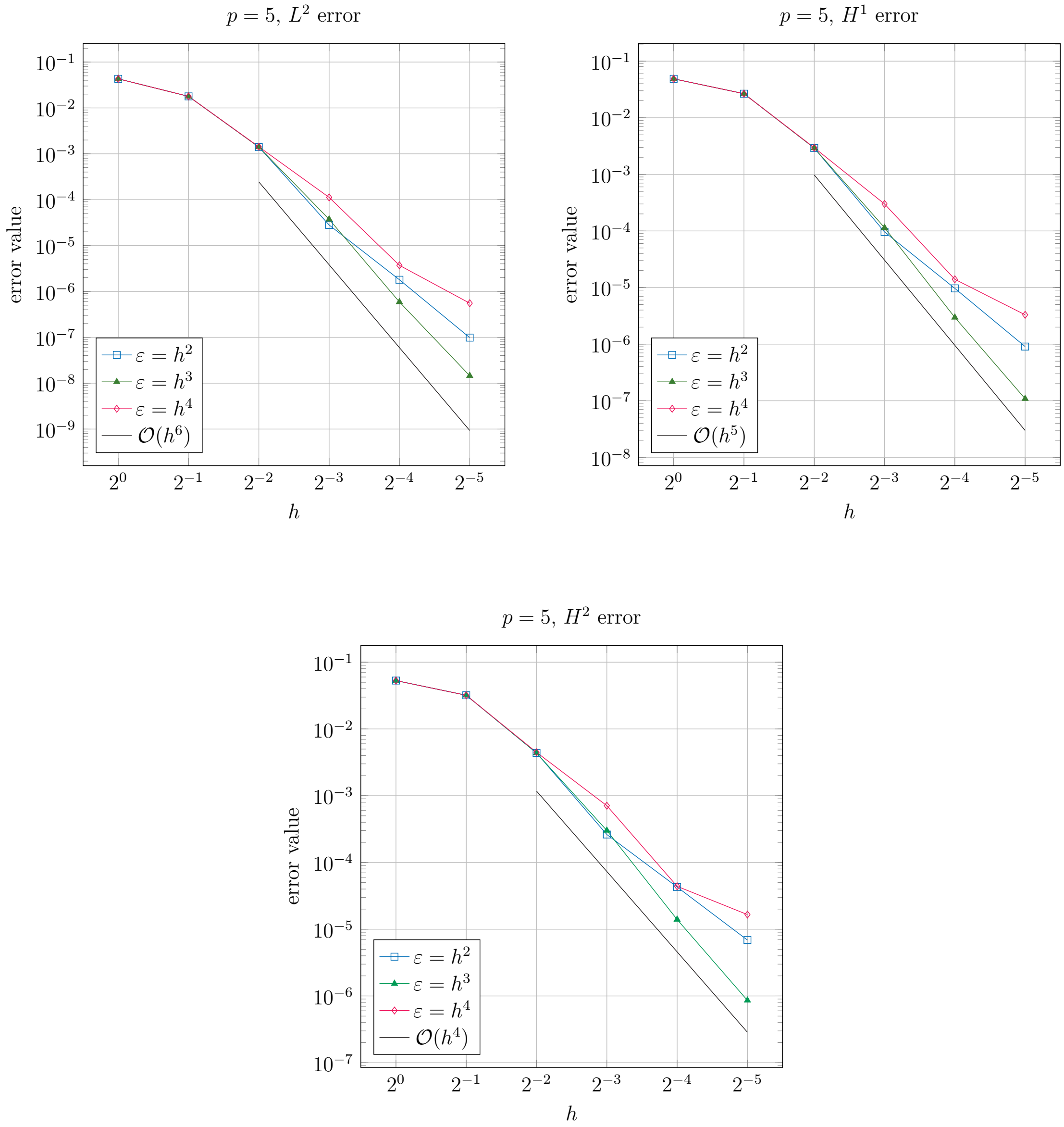


Figure 4.10: Biharmonic equation (4.26): Relative L^2 (top left), H^1 (top right) and H^2 (bottom) errors of the numeric solution for basis functions of degree five in $\hat{G}^{1,\varepsilon}_{h,\mathbf{0}}$ for three choices of ε.

for solving higher order problems. We conjecture that $\varepsilon = C \cdot h^{p-2}$ is the optimal choice.

4.4.2 Dimension of the space

We investigate the influence of ε on the number of interface basis functions, and thus on the dimension of the space $\hat{G}_h^{1,\varepsilon}$. Note that the number of trivially smooth basis functions is not affected by the choice of ε.

We cannot expect nested spaces, i.e., we cannot ensure that

$$\hat{G}_h^{1,\varepsilon} \subseteq \hat{G}_{\frac{h}{2}}^{1,\varepsilon}.$$

Nevertheless, the number of interface basis functions grows as h is decreased.

Figure 4.11 shows the number of interface basis functions for different degrees and different choices of ε. For $\varepsilon = C \cdot h^k$ with $k \leq p-2$, the number of interface basis functions grows linearly under h-refinement for all degrees. A larger choice of k, however, results in significantly fewer functions. This effect is clearly visible for $\varepsilon = h^4$ and the degrees three, four, five and six. The kink in the green curve, corresponding to $\varepsilon = h^3$, is a bit smaller. The results are consistent with the convergence rates for different degrees of test functions, which we discussed earlier.

4.5 Summary

We proposed a concept of constructing spaces $G_h^{1,\varepsilon}$ and $\hat{G}_h^{1,\varepsilon}$ of approximately $\mathcal{C}^1$-smooth isogeometric functions on planar multi-patch domains which is based on selecting eigenvalues and corresponding eigenvectors of the matrix representation of a suitable bilinear form. The resulting functions will have a non-zero gradient jump. We established the spaces in such a way that we can bound the gradient jump of the contained functions by construction.

We studied two different bilinear forms. The first bilinear form led to a bound on the gradient jump of the form

$$\|[\nabla f]\|_{L^2(e)}^2 \leq \varepsilon \|f\|_{L^2(\Omega)}^2,$$

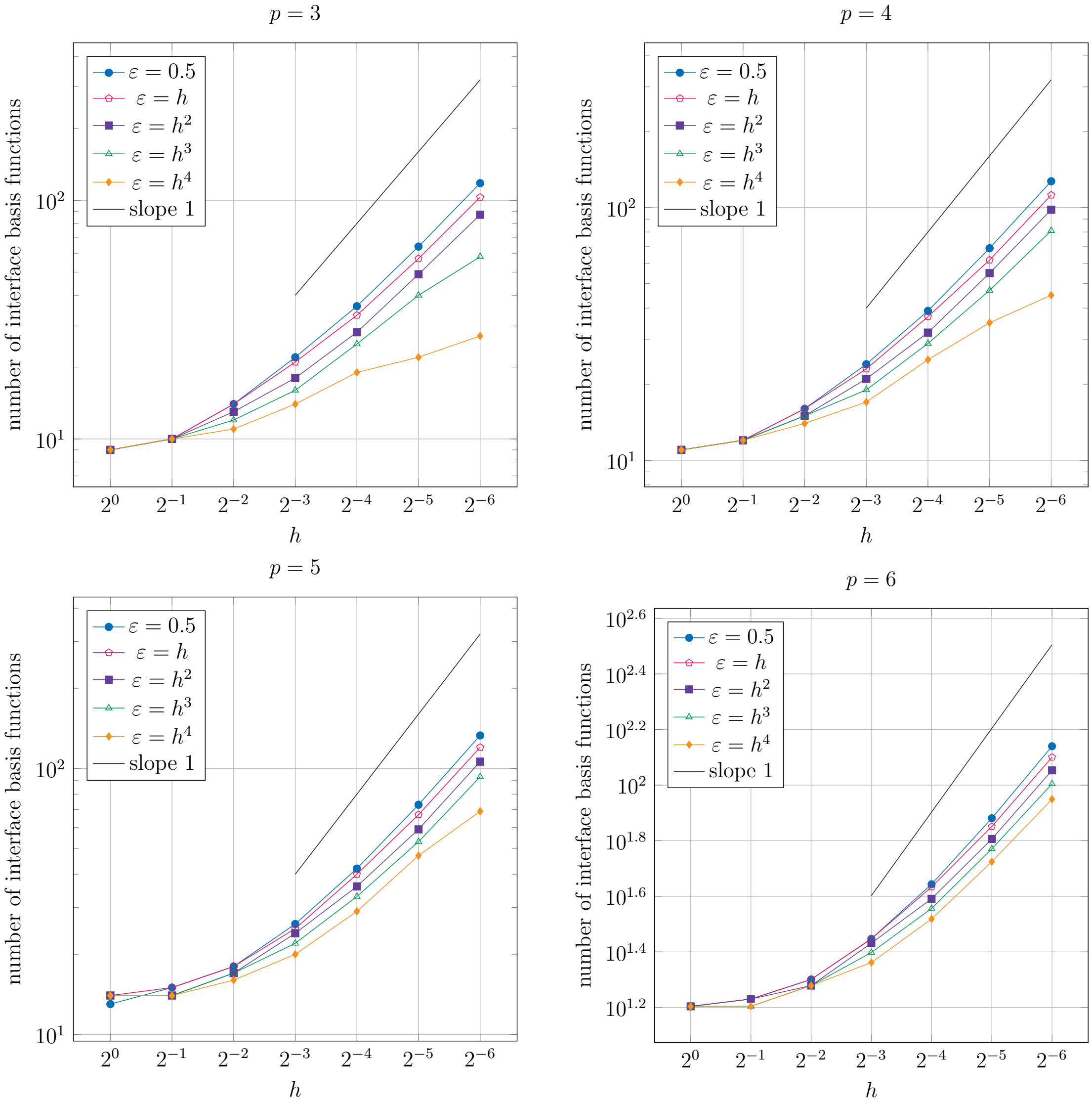


Figure 4.11: Number of interface basis functions under uniform h-refinement on the domain shown in Figure 4.1. Basis functions of degree three (top left), four (top right), five (bottom left) and six (bottom right) for various mesh-dependent choices of ε.

where ε is to be chosen in advance, in particular the bound is h-independent. On the other hand, the space constructed on basis of this bilinear form does not necessarily contain trivially smooth functions, which led to a decrease in the approximation order.

The function space $\hat{G}_h^{1,\varepsilon}$ based on the second bilinear form, which was designed to overcome the former drawbacks, contained all trivially smooth isogeometric functions and the gradient jump was bounded by

$$\|[\nabla f]\|_{L^2(e)}^2 \leq \varepsilon \frac{C(p,G)}{h^2} \|f\|_{L^2(\Omega)}^2.$$

Numerical experiments suggested that for second-order problems the latter approach maintains the optimal approximation order even for constant choices of ε and that the functions are sufficiently smooth to solve the biharmonic equation. The convergence of the approximate solution was influenced by the choice of ε. Depending on the degree of the basis functions, ε had to be chosen as a suitable power of the mesh size h in order achieve convergence of the solution to the biharmonic problem. On the other hand the dimension of $G_h^{1,\varepsilon}$ decreases as ε does. If too few functions remain in $G_h^{1,\varepsilon}$ we lose the optimal approximation properties. Thus, the choice of ε needs to balance the number of functions in $G_h^{1,\varepsilon}$ and their smoothness.

In future work one may focus on developing a theoretical background for the results concerning the approximation power of functions in $G_h^{1,\varepsilon}$. Also, one may look into the dimension of $G_h^{1,\varepsilon}$ and into bounding the number of non-trivial basis functions from below.

Chapter 5

Approximate Geometric Smoothness for Surface Reconstruction

In the last two chapters we covered topics related to approximate smoothness for numerical simulation. In this chapter, the focus of our attention will be shifted to surface reconstruction, more precisely to spline approximation. That means that we no longer deal with an isogeometric setting on a planar physical domain. Instead, from now on our considerations are restricted to tensor-product B-spline surfaces, defined on a parameter domain. To emphasize the changed setting, in this chapter we will again slightly modify the notation.

We aim at constructing a spline surface with minimal distance to given data points. If the spline basis is already known, the surface is determined by its control-points. Thus, our problem consists in finding (in our case) control points in $\mathbb{R}^3$ such that the distance between the resulting spline surface and the given data is minimal. We formulate the problem as an optimization problem, which combines minimizing point and normal errors with the use of norm-like functions, thereby generalizing the standard least squares fitting.

Using normal data helps us to connect the content of this chapter with the general topic of approximate smoothness on multi-patch domains. Assume the surface patch we want to construct is surrounded by neighboring patches. With the method we propose we can approximate point data on the whole patch. Simultaneously, we can measure normal data along the boundary edges of the neighboring patches and encourage the normals of the patch to be constructed to approximate them. This process enhances approximate geometric continuity between the neighboring surface patches.

The remainder of this chapter is organized as follows: In Section 5.1, we state the problem, introduce the used notation, which will differ slightly from the one presented in Chapters 3 and 4 and present our approach. Using uniform h-refinement, where h is the mesh size of the underlying knot span grid, we compute a series of solutions to our problem. In Section 5.2 we show that such a series of solutions exists and that it realizes the optimal approximation order for a suitable choice of the weight of the normal term, depending on h. In Section 5.3 we present numerical examples both with artificial and industrial data. Section 5.4 summarizes this chapter.

5.1 Simultaneous approximation of point and normal data

Consider the data depicted in Figure 5.1 (bottom right), which shows a part of a turbine and a turbine blade model. This point cloud is to be approximated by a spline surface. The parameter values shown on top are generated by a standard parameterization method [21] for meshes, applied to a triangulation of the data. In order to ensure smooth connections to the neighboring surfaces, we need to enforce approximate G^1-smoothness by additionally approximating the prescribed normal data along the red boundaries. We do not consider normals along the remaining two boundaries, since the associated neighboring surface patches (blends) are created in a subsequent step.

We realize the fillet surface construction by performing a minimization of the objective function

$$F(s) = \sum_{j=1}^{D} \nu(|x_s(t_j) - f_j|) + \gamma \sum_{k=1}^{K} \nu(|N x_s(\hat{t}_k) - n_k|) \rightarrow \min . \qquad (5.1)$$

In detail,

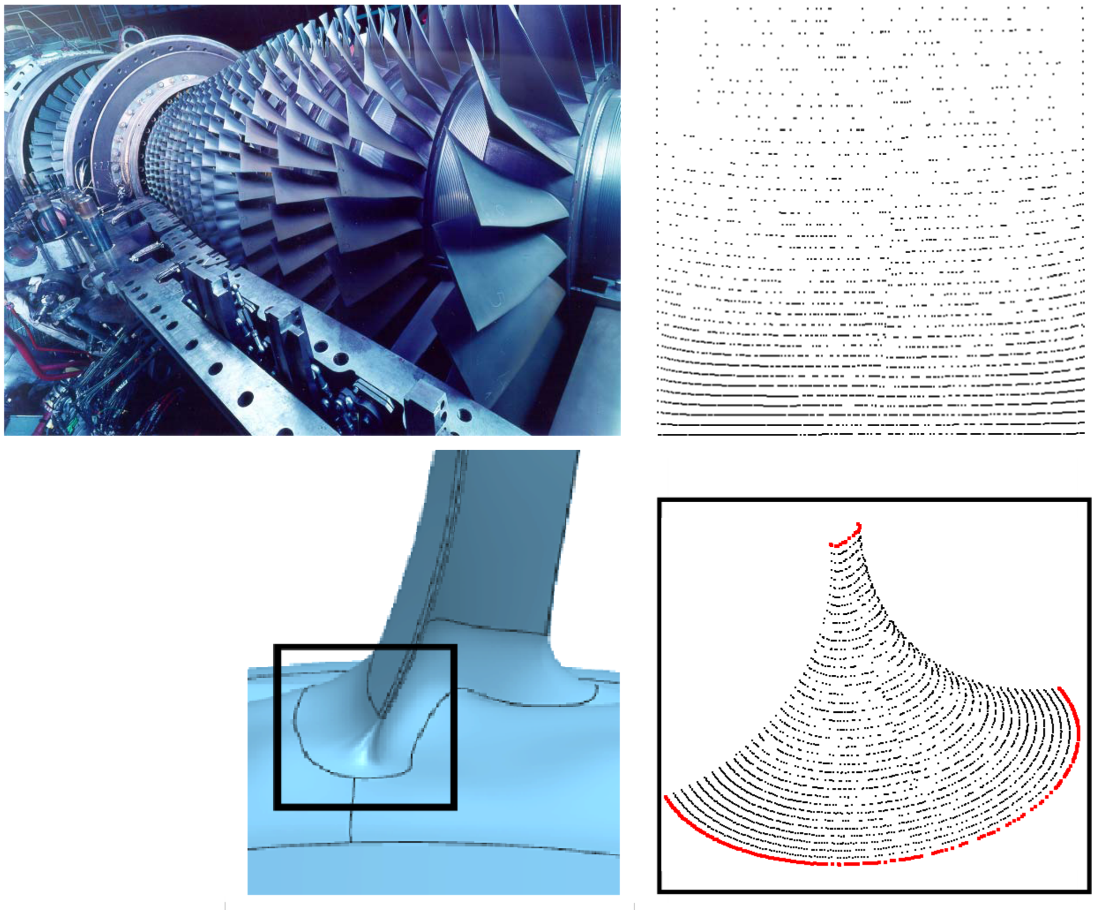


Figure 5.1: Turbine component (top left) and point cloud: Parametric (top right) and measured physical data (bottom right) of the fillet belonging the central part of the marked area of the turbine blade (bottom left).

- $|\cdot|$ denotes the standard Euclidean norm, $\nu : \mathbb{R}^+ \to \mathbb{R}^+ \in \mathcal{C}^2$ is a *norm-like function*[1] as described by [2],
- the approximating spline surface
$$x_s(t) = \sum_{i=1}^{n} c_i \beta_i(t), \quad t \in [0,1]^2,$$
which represents the fillet, depends on the vector
$$s = (c_1^T, \ldots, c_n^T) = (c_1^1, c_1^2, c_1^3, c_2^1, \ldots, c_n^1, c_n^2, c_n^3)$$
of control points, which are multiplied with either tensor-product B-splines or THB-splines [43] β_i,
- the surface approximates the given points f_j at parameter values $t_j \in [0,1]^2$, and the unit normal vectors n_k at parameter values $\hat{t}_k$,
- the operator N transforms the surface into the associated field of unit normal vectors, and
- the non-negative weight γ controls the relative influence of points and normals.

Note that in this chapter, like in Chapter 4, we do not use superscripts for basis functions or control points to refer to a specific single patch, since formally, we do not consider a multi-patch domain. Our problem formulation is given on a single patch. Furthermore, the former index set $\mathcal{I}$ and the patch-wise index sets $\mathcal{I}^1, \mathcal{I}^2$ are replaced by the set $\{1, \ldots, n\}$.

In the example shown in Figure 5.1, the distribution of the parameter values t_j is often quite non-uniform, and the values $\hat{t}_k$ are located on the patch boundary.

Due to the presence of the unit normals and for general norm-like functions, the minimization of (5.1) leads to a non-linear optimization problem. A necessary condition for s to be a minimizer of (5.1) is $\nabla F(s) = 0$. We solve this equation

---

[1] In particular, choosing $\nu(x) = x^2$ results in a standard least squares fitting problem. Please note that we consider a more restricted class of norm-like functions than in the original paper, see the assumptions (5.4) at the end of this section.

approximately by a Gauss-Newton-type method, i.e. a simplified Gauss-Newton method, where in the ℓ-th iteration we solve the linear system

$$H_F(s^\ell)\Delta s = -\nabla F(s^\ell) \tag{5.2}$$

and update the current solution s^ℓ via $s^{\ell+1} = s^\ell + (\Delta s)^T$. The solution of the standard least squares point fitting problem is used as start value. More precisely, we approximate the Hessian of the objective function by

$$H_F(s) = \sum_{j=1}^{D} \omega(|R_j|)\nabla R_j^T(s)\nabla R_j(s) + \sum_{k=1}^{K} \omega(|\hat{R}_k|)\nabla \hat{R}_k^T(s)\nabla \hat{R}_k(s),$$

where $R_j = x_s(t_j) - f_j$ and $\hat{R}_k = N x_s(\hat{t}_k) - n_k$ are the point and normal residuals, respectively, and

$$\omega(x) = \nu'(x)/x, \tag{5.3}$$

cf. [2]. The latter function is called the *weight function* associated with the norm-like function ν. The exact computation of the Hessian is costly, therefore the aforementioned approximation is chosen. In [2] it is shown that in the zero-residual case this expression converges to the true Hessian with respect to the spectral norm.

Besides the obvious choice $\nu(x) = x^2$, it is potentially useful to consider other types of norm-like functions, since they may enhance the performance of the method in the presence of outliers or improve the approximation result. However, for the analysis we restrict ourselves to norm-like functions that satisfy $\nu(0) = 0$ and whose weight functions possess the following properties:

- They have a global lower bound $\omega_{\min}$, i.e., $\omega(x) \geq \omega_{\min}$ holds for all $x \in \mathbb{R}^+$, and
- they possess an upper bound $\omega_{\max}(F)$ on each interval $[0, F]$, i.e., $\omega(x) \leq \omega_{\max}(F)$ holds for all $x \in [0, F]$.

We will refer to these as *norm-like functions with positive and partially bounded weights.* They satisfy

$$\frac{1}{2}\omega_{\min} x^2 \leq \nu(x) \leq \frac{1}{2}\omega_{\max}(F)\, x^2 \text{ on any interval } [0, F]. \tag{5.4}$$

It has been shown that in the zero-residual case, the method (5.2) achieves local quadratic convergence for this class of norm-like functions under certain technical assumptions, see [2, Theorem 5].

5.2 Existence of a solution and convergence rates

In order to gain some insight into the behavior of the solution to the minimization problem (5.1), we consider the continuous version

$$\hat{F}(s) = \|\nu(|x_s - f|)\|_{L^1} + \gamma\|\nu(|Nx_s - Nf|)\|_{L^1} \to \min, \tag{5.5}$$

where

$$f : [0,1]^2 \to \mathbb{R}^3 \tag{5.6}$$

is a given smooth surface that we want to approximate. We derived it by replacing the sums with integrals.

It should be noted that the presence of L^1 norms in (5.5) is caused by the use of the norm-like function ν. In the simplest possible case $\nu(x) = x^2$ one obtains

$$\hat{F}(s) = \|x_s - f\|_{L^2}^2 + \gamma\|Nx_s - Nf\|_{L^2}^2.$$

In this section we restrict the exposition to spline surfaces x_s defined by tensor-product B-splines B_i of degree (p,p), which are defined over two quasi-uniform open knot vectors $\Xi = (\Xi_1, \Xi_2)$ on $[0,1]^2$. As usual we use h to denote the mesh size, i.e., the maximum side length of the resulting elements.

We recall two basic facts from spline theory:

- Stability of a B-spline basis $\{B_i\}$ ([16], Theorem 4.1; [55], Theorem 12.5): There exists a constant $0 < D_p \leq 2(p+1)9^p$ such that

  $$\frac{1}{D_p^2}|s|_\infty \leq \|\sum_i c_i\beta_i\|_{L^\infty} \leq |s|_\infty. \tag{5.7}$$

  These inequalities are valid for splines with scalar coefficients. They can be generalized to spline surfaces by defining

  $$\|x_s\|_{L^\infty} = \max_{t\in[0,1]^2} |x_s(t)|$$

  and exploiting the equivalence

  $$|v|_\infty \leq |v| \leq \sqrt{3}|v|_\infty,$$

  of the Euclidean norm $|\cdot|$ and the maximum norm $|\cdot|_\infty$ in $\mathbb{R}^3$.

- Approximation power of splines with scalar coefficients [17]: There exist quasi-interpolation operators Π_Ξ that transform any given function $\phi \in H^{p+1}$ into a spline function $\Pi_\Xi\phi$. More precisely, the error satisfies
$$||\phi - \Pi_\Xi\phi||_{L^2} \leq C \cdot h^{p+1}||\phi||_{H^{p+1}}, \tag{5.8}$$
and
$$||\phi - \Pi_\Xi\phi||_{H^1} \leq C \cdot h^{p}||\phi||_{H^{p+1}}, \tag{5.9}$$
where the constant C does not depend on Ξ, h, or f, i.e., points are approximated with order $p+1$, while derivatives are approximated with order p. The result carries over to splines with values in $\mathbb{R}^3$ by applying Π_Ξ componentwise.

These observations will be used to derive a result about the existence of solutions and the convergence rate as h tends to zero. We consider a regular surface f, see (5.6), and a norm-like function ν satisfying $\nu(0) = 0$ with positive and bounded weight function, cf. (5.3) and (5.4).

Theorem 3. *The problem* (5.5) *has a solution for all pairs* Ξ *of knot vectors. The sequence of solutions realizes the optimal approximation order if* $\gamma = \gamma_0 h^2$ *for some positive constant* γ_0.

Before proving Theorem 3 we state an auxiliary result.

Lemma 4. *There exists constants* $C_N = C_N(f)$ *and* $h_0 = h_0(f)$*, which depend on the given surface* f*, such that*
$$\|N\Pi_\Xi f - Nf\|_{L^2} \leq C_N h^p$$
holds whenever $h < h_0$*, where* $\Pi_\Xi f$ *is applied to the elements of* f*, i.e.*
$$\Pi_\Xi f = \begin{pmatrix} \Pi_\Xi f^1 \\ \Pi_\Xi f^2 \\ \Pi_\Xi f^3 \end{pmatrix}.$$

The proof of this Lemma 4 follows from the approximation order of the derivatives and the fact that the normal depends continuously on the derivatives, taking the regularity of the parameterization into account. We present the details of this proof in the appendix of this chapter.

Proof of Theorem 3. First, we show that a solution exists for any pair of knot vectors. We restrict the $\|\cdot\|_{L^\infty}$- and the $\|\cdot\|_{L^2}$-norm to the spline space $\operatorname{span}\{B_i | i \in I_h\}$ which has finite dimension. Consequently, the two norms are equivalent, i.e., there exists a constant C_{norm} such that

$$\|\sum_{i\in I_h} c_i\beta_i\|_{L^\infty} \leq C_{\text{norm}}\|\sum_{i\in I_h} c_i\beta_i\|_{L^2}. \tag{5.10}$$

We combine this observation with (5.7) and obtain

$$\frac{1}{C_{\text{norm}}}\frac{1}{D_p^2}|s^*|_\infty \leq \|x_{s^*}\|_{L^2} = \|x_{s^*} - f + f\|_{L^2},$$

where s^* is the solution of (5.5). Now using the triangle inequality, the first inequality in (5.4) and the identity

$$\|\cdot\|_{L^2} = \sqrt{\|(\cdot)^2\|_{L^1}}$$

gives

$$\frac{1}{C_{\text{norm}}}\frac{1}{D_p^2}|s^*|_\infty \leq \sqrt{\frac{2}{\omega_{\min}}\|\nu(|x_{s^*} - f|)\|_{L^1}} + \|f\|_{L^2}. \tag{5.11}$$

The L^1 norm on the right-hand side is bounded by $\hat{F}(s)$ for any choice of s. In particular we may choose s as null vector and obtain

$$\frac{1}{C_{\text{norm}}}\frac{1}{D_p^2}|s^*|_\infty \leq \sqrt{\frac{2}{\omega_{\min}}\left(\|\nu(|f|)\|_{L^1} + \gamma\|\nu(|Nf|)\|_{L^1}\right)} + \|f\|_{L^2}. \tag{5.12}$$

Consequently, it suffices to consider the objective function (5.5) on the closed ball with radius

$$C_{\text{norm}}D_p^2\left(\sqrt{\frac{2}{\omega_{\min}}\left(\|\nu(|f|)\|_{L^1} + \gamma\|\nu(|Nf|)\|_{L^1}\right)} + \|f\|_{L^2}\right),$$

which is a compact domain. The continuity of the objective function thus ensures the existence of the minimum.

In order to establish the optimality of the approximation order, we consider a sequence of knot pairs $\Xi = \Xi_h$ with decreasing element size h and show that the point and normal errors of the solutions to (5.5) satisfy

$$\sqrt{\|\nu(|x_h - f|)\|_{L^1}} \leq C'h^{p+1} \text{ and } \sqrt{\|\nu(|Nx_h - Nf|)\|_{L^1}} \leq C'h^p \tag{5.13}$$

for some constant C', which is independent of h.

As

$$\|\Pi_\Xi f - f\|_{L^\infty([0,1]^2)}$$

is bounded and since

$$|(\Pi_\Xi f - f)(t)| \le \|\Pi_\Xi f - f\|_{L^\infty([0,1]^2)} \text{ for almost all } t \in [0,1]^2$$

we choose

$$F = \max\{2, \|\Pi_\Xi f - f\|_{L^\infty([0,1]^2)}\}$$

and obtain

$$\begin{aligned}&\|\nu(|\Pi_\Xi f - f|)\|_{L^1} + \gamma_0 h^2 \|\nu(|N\Pi_\Xi f - Nf|)\|_{L^1}\\ &\quad\le \left\| \sqrt{\frac{\omega_{\max}(F)}{2}} |\Pi_\Xi f - f| \right\|_{L^2}^2 + \gamma_0 h^2 \left\| \sqrt{\frac{\omega_{\max}(F)}{2}} |N\Pi_\Xi f - Nf| \right\|_{L^2}^2 .\end{aligned}$$

Thus, the value of objective function at the coefficients of the surfaces $\Pi_\Xi f$ has the upper bound

$$\frac{\omega_{\max}(F)}{2}(C^2 + \gamma_0 C_N^2) h^{2p+2},$$

where the constants C and C_N are taken from (5.8) and Lemma 4, respectively. This implies that the solution of (5.5) satisfies (5.13) with

$$C' = \max\left\{ \sqrt{\frac{\omega_{\max}(F)}{2}(C^2 + \gamma_0 C_N^2)}, \sqrt{\frac{\omega_{\max}(F)}{2} \frac{C^2 + \gamma_0 C_N^2}{\gamma_0}} \right\}.$$

□

This result is equivalent to the simple observation that – under a suitable regularity assumption – the unit normals possess the same approximation order as the derivatives. It carries over to the considered class of norm-like functions, which satisfy (5.4) and $\nu(0) = 0$.

It should be noted that the choice of γ, which is described in the above theorem, is similar to the weights that appear in the jump and average terms used in the discontinous Galerkin discretization of partial differential equations, cf. [57].

5.3 Numerical experiments

This section has three parts. First, we verify the theoretical results of the previous section by considering synthetic data sets. Second, we apply the fitting technique to the industrial data set representing the fillet surface. Finally, we explore the effects of using norm-like functions.

5.3.1 Least-squares fitting: synthetic data

We demonstrate the theoretical findings concerning the convergence rates by applying the fitting procedure with the trivial norm-like function $\nu(x) = x^2$ to two synthetic data sets. First, we generated point and normal data by uniformly sampling 10^4 points and normals from the graph of the trigonometric curve $\cos(20t)$, $t \in [0,1]$. Second we consider another data set obtained by evaluating 201^2 points and normals on a uniform grid in the domain of the ellipsoidal patch

$$(t_1, t_2) \mapsto \begin{pmatrix} \frac{3}{2} \cdot \cos(\frac{\pi}{3}t_1 - \frac{\pi}{6}) \cdot \cos(\frac{\pi}{2}t_2 + \frac{5\pi}{4}) \\ \frac{4}{5} \cdot \cos(\frac{\pi}{3}t_1 - \frac{\pi}{6}) \cdot \sin(\frac{\pi}{2}t_2 + \frac{5\pi}{4}) \\ \frac{6}{5} \cdot \sin(\frac{\pi}{3}t_1 - \frac{\pi}{6}) \end{pmatrix}, \quad t \in [0,1]^2, \tag{5.14}$$

see Figure 5.2, highlighted patch.

We computed the approximation result by using the Gauss-Newton method described in Section 5.1. The iterative procedure was terminated when the gradient of the objective function satisfied $|\nabla F(s^\ell)| \leq 10^{-8}$ or when the iteration count reached 500 (but this was never the case for the two synthetic data sets). The basis functions β_i were chosen as (bi-) cubic (tensor-product) B-splines. We consider uniform open knot vectors with mesh size h varying between 1 and 2^{-8} or 2^{-5} for curves and surfaces, respectively.

Figures 5.3 and 5.4 report the resulting root of the sum of squared errors with respect to the Euclidean norm for $\gamma = h^k$, $k = 0, 1, 2, 3$ and for $\gamma = 0$. Among the considered choices of the weight γ, the optimal rates of convergence (4 for the point data and simultaneously 3 for the normals) are achieved for $\gamma = h^2$, $\gamma = h^3$ and for $\gamma = 0$. The first case is covered by the theoretical results in the previous section, while the other ones confirm that standard L^2 approximation also provides

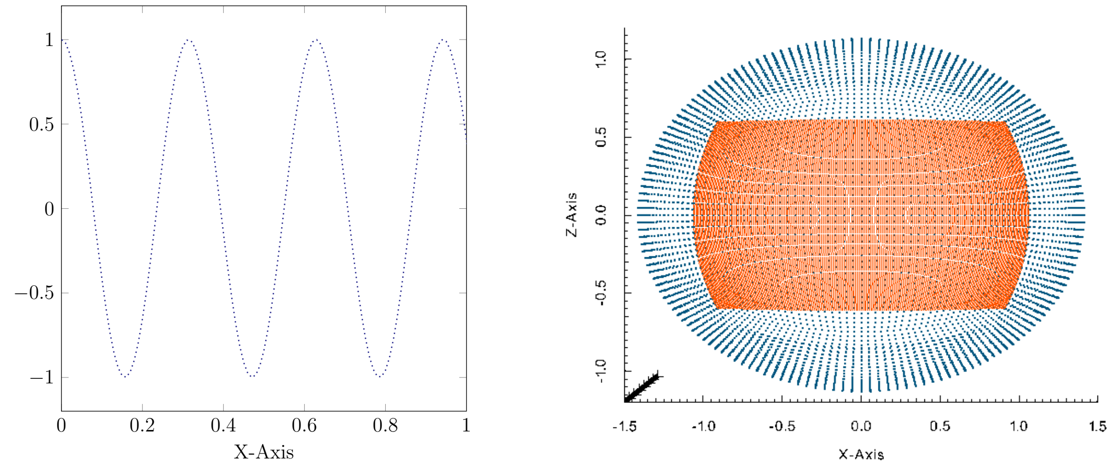


Figure 5.2: One- (left) and two-dimensional (right) synthetic data for the numerical tests.

the optimal rate of convergence of the derivatives. Consequently, using the normal vectors does not provide any advantages for synthetic data. However, the example in the next section will demonstrate the benefits for an industrial application.

Figures 5.5 and 5.6 demonstrate the influence of the constant γ_0. It can be seen that the fitting result is fairly robust with respect to variations of this weight. Note that the effects of changing γ_0 are not invariant under scaling. This issue can be resolved by performing a suitable scaling of the data.

5.3.2 Least-squares fitting: industrial data

Next we apply the fitting procedure to the fillet data shown in Figure 5.1, which consists of 3,280 point and 214 normal samples. The normal data is available only at the top and the bottom of the fillet, in order to ensure a smooth connection to the adjacent patches. First we consider tensor-product discretizations with varying mesh size h (up to 1,225 control points) obtained by dyadic refinement. We regularized the objective function by introducing a quadratic surface energy with weight $\lambda = 10^{-5}$, see [43] for details. This also has a fairing effect.

Figure 5.7, light and dark blue curves, visualizes the resulting sum of squared errors with respect to the Euclidean norm for $\gamma = 10^{-3}h^2$ and for $\gamma = 0$ (without normals). It can be seen that using the normal data is essential, since the nor-

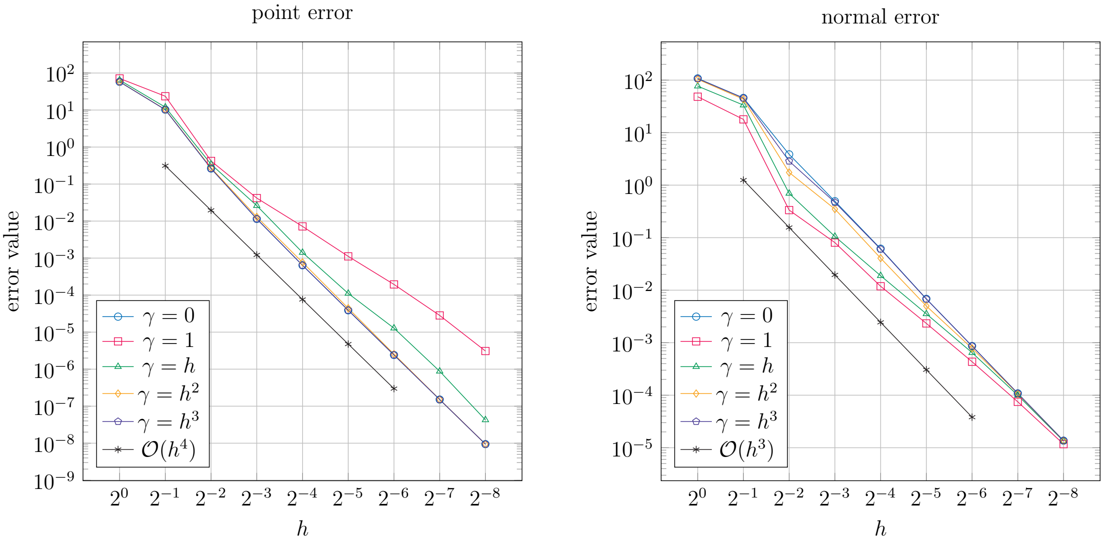


Figure 5.3: Synthetic curve data: Point (left) and normal (right) error for various choices of weight γ and mesh size h.

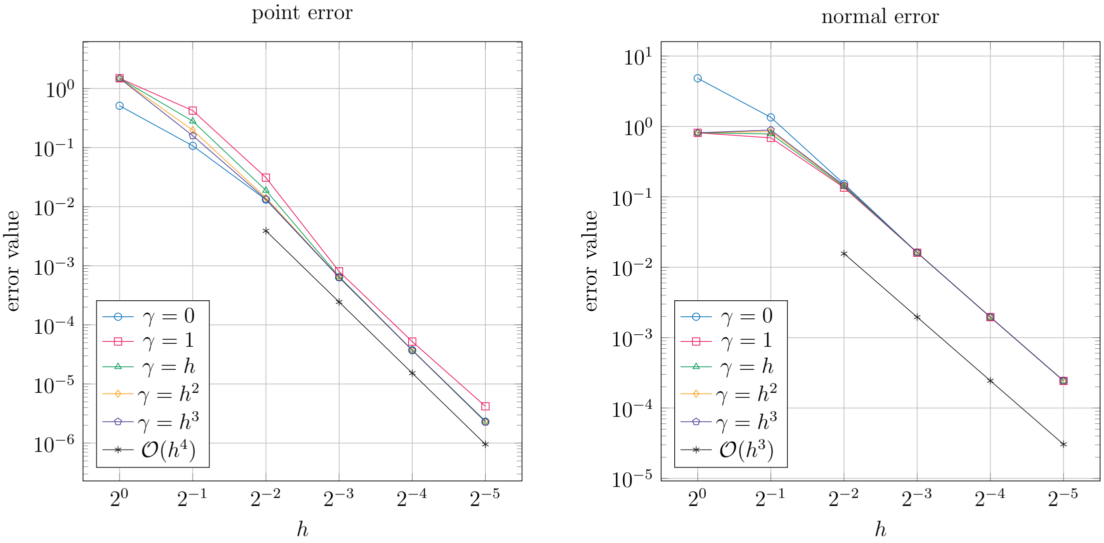


Figure 5.4: Synthetic surface data: Point (left) and normal (right) error for various choices of weight γ and mesh size h.

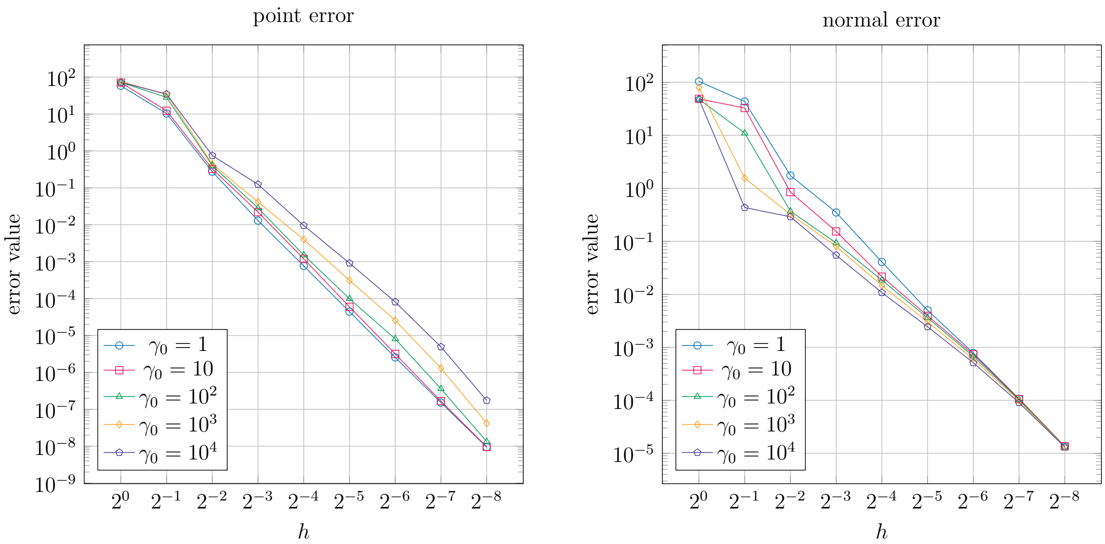


Figure 5.5: Synthetic curve data: Point (left) and normal (right) error for various choices of the constant γ_0 where $\gamma = \gamma_0 h^2$.

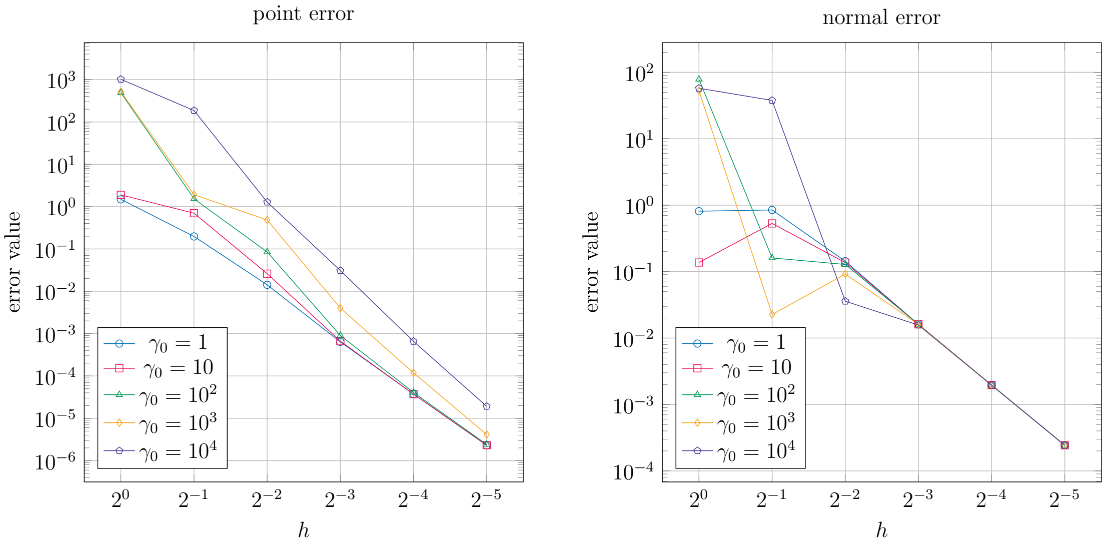


Figure 5.6: Synthetic surface data: Point (left) and normal (right) error for various choices of the constant γ_0 where $\gamma = \gamma_0 h^2$.

mals do not converge for $\gamma = 0$. Clearly, one cannot expect to achieve the same convergence rate as for the synthetic data, due to measurement errors. Also, the normals were sampled from neighboring patches, not from the fillet.

Figure 5.8 depicts the resulting fillet patches with (right) and without (left) approximating normal vectors for the finest tensor-product spline discretization. We added reflection lines to visualize the surface quality. On the one hand, the use of normal data (right) clearly improves the G^1 smoothness across the patch boundaries, since the reflection lines are continuous. This is especially visible in the marked areas, which are shown again in Figure 5.9, where there are discontinuities in the reflection lines of the left plot and smoother transitions in the reflection lines of the right plot. However, even at the finest discretization with 1,225 control points, the normal error was still relatively large (sum of squared errors $6.1 \cdot 10^{-7}$ and maximum angle $1.65 \cdot 10^{-2}$ degrees for $\gamma = 10^{-3}h^2$).

A further improvement – even when using a much smaller number of degrees of freedom – can be obtained by employing THB-spline discretizations instead of tensor-product splines, similar to the techniques reported by [25, 43]. We use the absolute threshold refinement strategy with $\varepsilon = 10^{-6}$ and compare the results obtained without (left) and with (right) using normal data in Figure 5.10. Both surfaces have acceptable quality, but the use of normal data again improves the G^1 smoothness (see close-up views). The leftmost reflection line is discontinuous without normal information.

Table 5.1 reports the resulting errors for the two THB-spline approximations. In particular, while the use of normal data does not compromise the point error, it significantly improves the approximate G^1 smoothness. Here, h refers to the mesh size of the finest discretization level.

The finest THB-spline discretization with 841 control points leads to a significantly smaller normal error than the finest uniform discretization (sum of squared errors norm $2.01 \cdot 10^{-8}$ and maximum angle $3.24 \cdot 10^{-3}$ degrees for $\gamma = 10^{-3}h^2$).

The sum of squared errors with respect to the Euclidean norm for $\gamma = 10^{-3}h^2$ and for $\gamma = 0$ (without normals) for different numbers of control points is shown in Figure 5.7, dark and light red curves. Here, again, the effect of reaching a smaller normal error with less control points is evident.

Clearly, we obtain different THB meshes for $\gamma = 0$ and $\gamma = 10^{-3}h^2$, which are visualized in Figure 5.11. The two surfaces have 556 and 841 control points. The use of the normal data led to additional refinement near the boundaries.

5.3.3 Norm-like functions

Now we consider the influence of different choices of ν while neglecting the normal data approximation term, using the data previously introduced (Figure 5.1 and 5.2, right). We artificially introduce outliers to the ellipsoidal patch values in order to illustrate the effects of different alternatives of ν. More precisely, we set the value of the third component of three of the 201^2 samples to 1.5 instead of 1.02083, 1.02901 and 0.979487, respectively. Thus, the outliers have a distance of 0.64977, 0.65063 and 0.67656 from the original data points. We do not have further information about the industrial data from Figure 5.1, but most likely there will be some noise in the data.

Following [2], we select the norm-like functions

- $\nu_1(x) = 1 - \exp(-\eta^2 x^2)$,
- $\nu_2(x) = x^2$, which corresponds to a standard least-squares fitting problem, and
- $\nu_3(x) = \exp(\eta^2 x^2) - 1$.

The constant η was chosen as described in [2]. Minimizing the first and the third function, which are motivated by the statistics literature, has been observed to have a beneficial effect on outliers and on the maximum error, respectively.

The latter two functions are norm-like functions with positive and partially bounded weights, since they fulfill the assumptions specified in Section 5.1. In contrast, the weight function associated with $\nu_1(x)$ is not globally bounded from below. One may replace the weight by a constant for all arguments exceeding some threshold, in order to satisfy the assumptions.

For the results in Table 5.2 we used a tensor-product B-spline discretization with B-splines of degree (3,3) and mesh size $h = 0.03125$, as in the previous experiments, and we set $\gamma = 0$. As expected, the ℓ_1-error is minimal when using

	without normals $\gamma = 0$, $3 \cdot 556$ dofs	with normals $\gamma = 10^{-3}h^2$, $3 \cdot 841$ dofs
$\sum_j \|x_{s,h}(t_j) - f_j\|_2^2$	$8.15 \cdot 10^{-10}$	$7.48 \cdot 10^{-10}$
$\sum_k \|Nx_{s,h}(\hat{t}_k) - n_k\|_2^2$	$1.32 \cdot 10^{-2}$	$2.01 \cdot 10^{-8}$
maximum angle between normals in degree	0.94	$3.24 \cdot 10^{-3}$

Table 5.1: Error values of the THB-spline approximations.

	Ellipsoid data with outliers			Fillet data		
	$\nu_1(x)$	$\nu_2(x)$	$\nu_3(x)$	$\nu_1(x)$	$\nu_2(x)$	$\nu_3(x)$
$\sum_j \|R_j\|_2$	6.1	7.17	7.86	0.001242	0.001242	0.001248
$\sqrt{\sum_j \|R_j\|_2^2}$	1.1	1.0942	1.095	$2.42 \cdot 10^{-5}$	$2.42 \cdot 10^{-5}$	$2.43 \cdot 10^{-5}$
$\max_j \|R_j\|_2$	0.66	0.65	0.64	$1.514 \cdot 10^{-6}$	$1.514 \cdot 10^{-6}$	$1.513 \cdot 10^{-6}$

Table 5.2: Error values in different ℓ_p norms for approximations of the ellipsoidal patch and fillet data.

ν_1 and an analogous outcome is visible for the other two norms. Note that the differences between the error values are more distinct for the ellipsoidal patch data than for the fillet data set which is due to the presence of very few strong outliers. The influence of the choice of ν is not visible for the fillet data where the error appears to have a more uniform distribution.

5.4 Summary

We considered the simultaneous approximation of point and normal data using norm-like functions by tensor-product B-spline surfaces. This leads to a non-linear optimization problem, which has been solved by a Gauss-Newton-type technique, based on the earlier results reported by [2]. We showed that for all pairs of knot vectors of the tensor-product B-splines, a solution to the minimization problem exists and that for a suitable choice of the weight of the normal data term the series of solution realizes the optimal approximation order.

We used an industrial data set to illustrate the advantages of fitting point and normal data simultaneously. In this context we combined our approach with the mathematical technology of truncated hierarchical B-splines (THB-splines), which provided again a significant improvement of the fitting results. As observed in our experiments, the use of normal information helps greatly to maintain approximate G^1 smoothness across patch boundaries.

5.5 Appendix: Proof of Lemma 4.

For $p > \frac{3}{2}$ and for sufficiently small element size, the derivatives of given surface f and of its spline approximation $\Pi_\Xi f$ satisfy

$$\|\partial_1 f\|_{L^\infty} \le M, \qquad \|\partial_2 f\|_{L^\infty} \le M, \tag{5.15}$$

$$\|\partial_1 \Pi_\Xi f\|_{L^\infty} \le M, \qquad \|\partial_2 \Pi_\Xi f\|_{L^\infty} \le M, \tag{5.16}$$

$$|\partial_1 f(t) \times \partial_2 f(t)| \ge m, \quad |\partial_1 \Pi_\Xi f(t) \times \partial_2 \Pi_\Xi f(t)| \ge m \, \forall t \in [0,1]^2 \tag{5.17}$$

for some constants $M, m > 0$, see [1]. Note that the latter two inequalities are induced by the regularity assumption for the surface f.

We consider the difference of the unit normals,

$$\left\| \frac{\partial_1 f \times \partial_2 f}{|\partial_1 f \times \partial_2 f|} - \frac{\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f}{|\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f|} \right\|_{L^2}. \tag{5.18}$$

The subtrahend of the difference can be rewritten as

$$\frac{\left(\frac{|\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f|^2}{|\partial_1 f \times \partial_2 f|} \partial_1 f + |\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f| \partial_1 \Pi_\Xi f - \frac{|\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f|^2}{|\partial_1 f \times \partial_2 f|} \partial_1 f \right) \times \partial_2 f}{|\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f|^2} +$$

$$\frac{\left(\frac{|\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f|^2}{|\partial_1 f \times \partial_2 f|} \partial_1 f + |\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f| \partial_1 \Pi_\Xi f - \frac{|\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f|^2}{|\partial_1 f \times \partial_2 f|} \partial_1 f \right) \times \partial_2 \Pi_\Xi f}{|\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f|^2} -$$

$$\frac{\left(\frac{|\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f|^2}{|\partial_1 f \times \partial_2 f|} \partial_1 f + |\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f| \partial_1 \Pi_\Xi f - \frac{|\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f|^2}{|\partial_1 f \times \partial_2 f|} \partial_1 f \right) \times \partial_2 f}{|\partial_1 \Pi_\Xi f \times \partial_2 \Pi_\Xi f|^2}$$

After substituting the rewritten subtrahend into (5.18), we expand the cross product and use the triangle inequality to expand the result into four terms.

The first term

$$\left\| \frac{\partial_1 f \times \partial_2 f}{|\partial_1 f \times \partial_2 f|} - \frac{\partial_1 f \times \partial_2 f}{|\partial_1 f \times \partial_2 f|} \right\|_{L^2}$$

is equal to zero. The second term satisfies

$$\left\| \frac{1}{|\partial_1 f \times \partial_2 f|} \left(\partial_1 f \times (\partial_2 \Pi_\Xi f - \partial_2 f) \right) \right\|_{L^2} \le \frac{1}{m} \cdot M \cdot C \cdot h^p \cdot \|f\|_{H^{p+1}}.$$

The third term takes the form

$$\left\| \frac{(|\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f|\,|\partial_1 f \times \partial_2 f|\partial_1\Pi_\Xi f - |\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f|^2\partial_1 f) \times \partial_2 f}{|\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f|^2|\partial_1 f \times \partial_2 f|} \right\|_{L^2}.$$

The first factor of the numerator can be rewritten as

$$\begin{aligned} &|\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f| \cdot |\partial_1 f \times \partial_2 f|\partial_1\Pi_\Xi f - |\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f|^2\partial_1 f \\ &\quad + |\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f| \cdot |\partial_1 f \times \partial_2 f|\partial_1 f - |\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f| \cdot |\partial_1 f \times \partial_2 f|\partial_1 f. \end{aligned}$$

This helps us to obtain an upper bound for the third term,

$$\frac{M}{m}\|\partial_1\Pi_\Xi f - \partial_1 f\|_{L^2} + \frac{M^2}{m^2}\|(\partial_1 f \times \partial_2 f) - (\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f)\|_{L^2}.$$

Using the properties (5.9) of the spline projector confirms that this can be bounded by

$$\frac{M}{m}C \cdot h^p \cdot \|f\|_{H^{p+1}} + 4\frac{M^2}{m}C \cdot h^p \cdot \|f\|_{H^{p+1}},$$

where we used the observation that

$$\begin{aligned} &\|(\partial_1 f \times \partial_2 f) - (\partial_1\Pi_\Xi f - \partial_2\Pi_\Xi f)\|_{L^2} \\ &= \|(\partial_1 f \times \partial_2 f) - ((\partial_1 f + \partial_1\Pi_\Xi f - \partial_1 f) \times (\partial_2 f + \partial_2\Pi_\Xi f - \partial_2 f))\,\|_{L^2} \\ &= \|(\partial_1 f \times \partial_2 f) - (\partial_1 f \times \partial_2 f) - ((\partial_1\Pi_\Xi f - \partial_1 f) \times \partial_2 f) \\ &\qquad - (\partial_1 f \times (\partial_2\Pi_\Xi f - \partial_2 f)) - ((\partial_1\Pi_\Xi f - \partial_1 f) \times (\partial_2\Pi_\Xi f - \partial_2 f))\,\|_{L^2} \\ &\leq M\|\partial_1\Pi_\Xi f - \partial_1 f\|_{L^2} + M\|\partial_2\Pi_\Xi f - \partial_2 f\|_{L^2} + 2M\|\partial_2\Pi_\Xi f - \partial_2 f\|_{L^2}. \end{aligned}$$

Finally, the expansion generates a fourth term, which is equal to

$$\left\| \frac{\left(|\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f|\partial_1\Pi_\Xi f - \frac{|\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f|^2}{|\partial_1 f \times \partial_2 f|}\partial_1 f\right) \times (\partial_2\Pi_\Xi f - \partial_2 f)}{|\partial_1\Pi_\Xi f \times \partial_2\Pi_\Xi f|^2} \right\|_{L^2}$$

and can be dealt with in the same way as for the third one.

Summing up, all four terms are bounded by terms of the form $\frac{1}{4}C_N h^p$ for a suitable choice of the constant C_N (which depends on f). This completes the proof.

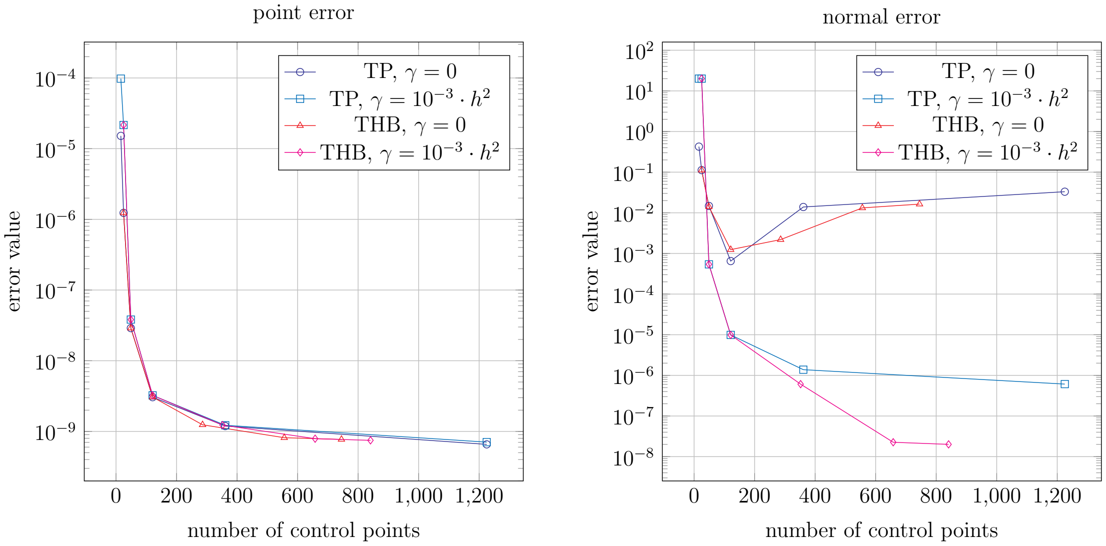


Figure 5.7: Point (left) and normal (right) error for tensor-product (blue) and THB-spline (red) approximations of the fillet data.

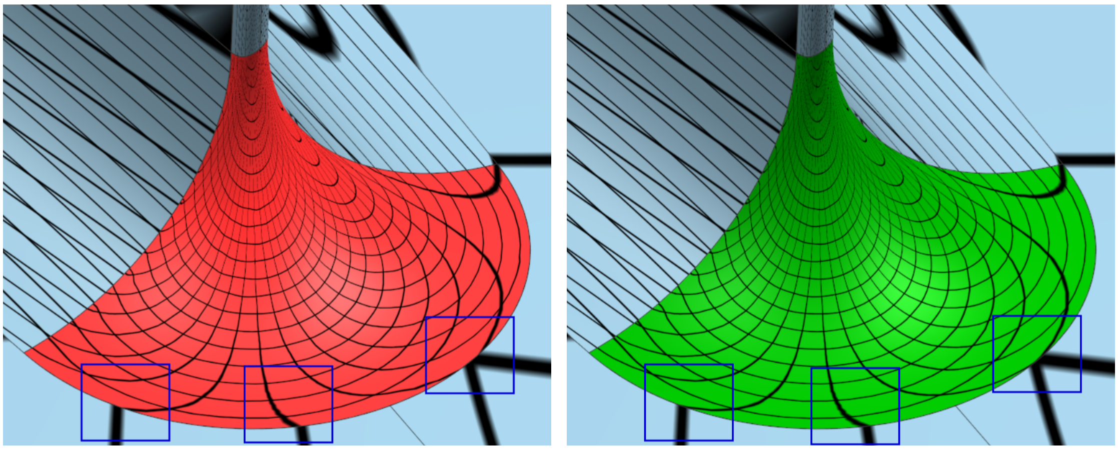


Figure 5.8: Tensor-product spline approximations of the fillet data without (left) and with (right) using normal information.

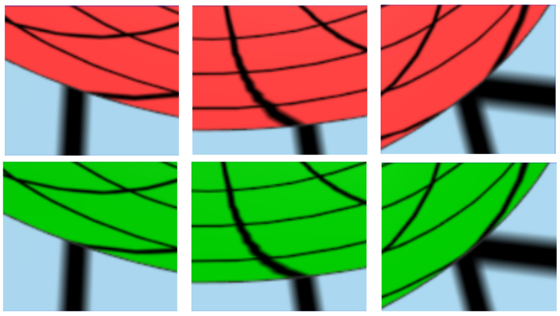


Figure 5.9: The details shown in the marked areas from left to right.

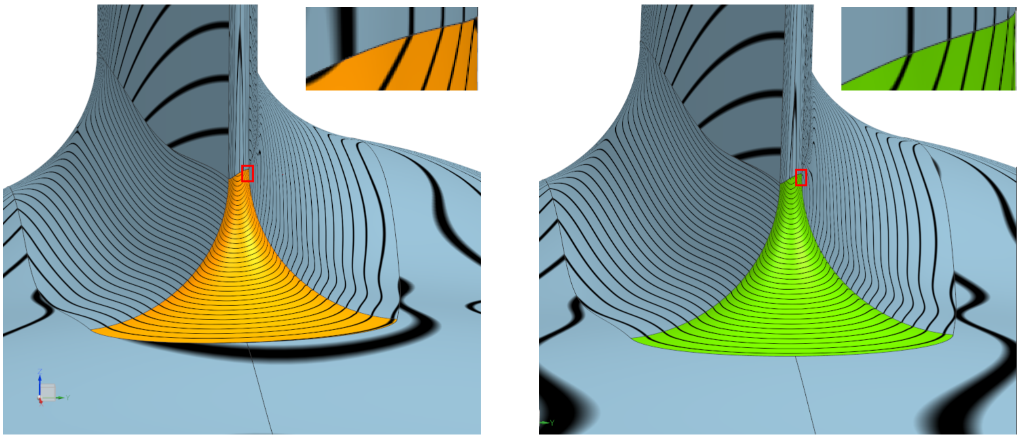


Figure 5.10: THB-spline approximations of the fillet data without (left) and with (right) using normal information.

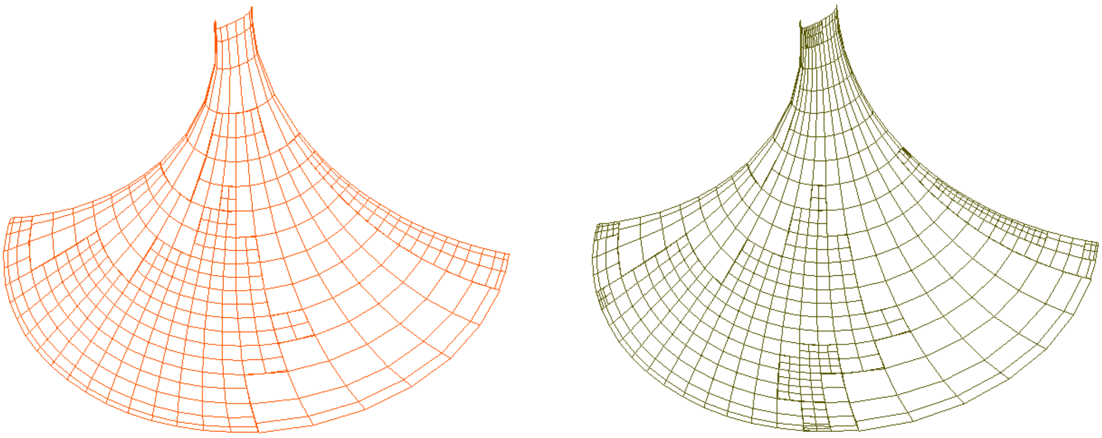


Figure 5.11: Control nets of the THB-spline surfaces approximating the fillet data without (left) and with (right) using normal information.

Chapter 6

Conclusion and Future Work

In recent years, Isogeometric Analysis has proven to be a powerful tool for the discretization of partial differential equations. The method relies on the geometry mapping which parameterizes the computational domain over a simple parametric domain by means of tensor-product spline functions. The test and ansatz functions are given by isogeometric functions, which are defined as the push-forward of the same B-splines that parameterized the geometry. As a consequence, no approximation of the physical domain is required, hence the numerical simulation is performed on the exact geometry at all times.

More complicated domains are represented as the union of several simpler single patches. In the context of numerical simulation, special attention has to be paid to the interfaces between neighboring patches. Smoothness of an isogeometric function on a single patch is given naturally, but global smoothness of such a function across patch interfaces does not come for free. This can be accomplished in different ways. This thesis studied two methods concerning the coupling of isogeometric functions on multi-patch domains.

Coupling techniques mainly can be categorized into two groups: The first group works with test function spaces on the individual patches and encourages smooth transitions by adapting the weak problem formulation, e.g. by adding jump penalty terms.

Among the methods of this group we particularly studied the isogeometric discontinuous Galerkin method in Chapter 3. The main contribution of this thesis to the context of dG-IgA methods was to provide a way to handle non-matching interface parameterizations. These give rise to difficulties when it comes to evaluating integrals of products of basis functions from different patches on the interface. We explained how to reparameterize the interface in order to identify pairs of corresponding points in the parameter domain with respect to function evaluation and how to choose suitable quadrature knots. Numerical experiments also included adaptive quadrature and the results suggested that in a two-dimensional setting both exact quadrature combined with a suitable splitting of the knot spans on the interface and adaptive quadrature realize optimal convergence rates.

Furthermore, in both approaches, the computational effort to evaluate integrals interior to single patches outweighted the effort of numerical integration on the interface. Future work may be devoted to the extension of the adaptive quadrature-based approach to the three-dimensional case. Here, we expect adaptive quadrature to computationally outperform exact quadrature combined with splitting of the two-dimensional knot spans.

The second group of coupling methods leaves the weak problem formulation unchanged and adapts the test function space instead. Constructing smooth isogeometric functions on a multi-patch domain is a challenging task, which often is accompanied by restrictive assumptions on the parameterization of the underlying geometry. In Chapter 4 we proposed an approach to constructing approximately $\mathcal{C}^1$-smooth isogeometric functions on general domains, i.e. our approach is independent of the geometry parameterization. In exchange, the constructed functions are not exactly smooth. The proposed approach was based on the choice of a suitable bilinear form and some of its eigenvalues and corresponding eigenvectors. This facilitated bounding the gradient jump of the constructed functions. Additionally, this magnitude of the jump was controlled by a parameter ε, which we chose in advance. Numerical experiments suggested that for second order problems, the space $G_h^{1,\varepsilon}$ maintains full approximation power even for constant, i.e., mesh-independent choices of ε. Our functions were smooth enough to serve as test functions for a fourth order problem. In order to restore the optimal convergence rates in this case, ε had to be chosen as a power of the mesh size. The suitable exponent had to be adapted to the degree of the basis functions.

In future work we would like to establish a theoretical background for the experimental results. This includes

- investigating a projector to the space of approximately $\mathcal{C}^1$-smooth isogeometric functions to proof optimal convergence rates,
- studying the eigenstructure of the matrix Q to develop a lower bound for the number of non-trivial basis functions and
- analyzing the influence of ε.

The last point affects the first two points: A smaller value of ε creates smoother but at the same time less functions. Figure 4.11 seemed to indicate that a linear growth rate of the number of interface basis functions can be restored if ε is chosen in $\mathcal{O}(h^{p-2})$. Hence, choosing ε is a trade-off between keeping enough functions and making them sufficiently smooth. At the same time, this has an impact on the approximation power, as seen e.g. in Figure 4.10. Developing a result that establishes a relation between the degree of the basis functions, the choice of ε and the resulting approximation power will be a prime focus in future work. Last but not least future work includes generalizing the approach to domains with more than two patches.

The last part of this thesis differed from the first two parts: Firstly, we considered splines instead of isogeometric functions and secondly we worked with geometrical instead of parametrical smoothness. Especially in industrial applications as presented in Chapter 5 approximate geometrical smoothness can be sufficient.

In this last part we presented a spline fitting technique, in which point and normal data were approximated simultaneously by one tensor-product B-spline patch. If the normals were sampled from a neighboring patch, this technique facilitates approximately G^1-smooth transitions across the patch interface. The problem was formulated as a minimization problem using norm-like functions, which made it non-linear. Standard least-squares fitting was contained in the setting by choosing a specific yet simple norm-like function. We applied a Gauss-Newton technique to find a minimizer, which we have shown to exist, of the problem.

In the context of coupling techniques, fitting the normals additionally to fitting point samples can be seen as a jump penalty term. Consequently, similar to choosing penalty parameters in non-linear optimization techniques, weighting the normal fitting term is a non-trivial task. We showed that the weight should be chosen in dependence on the mesh size in order to realize optimal convergence rates.

In experiments, we also computed truncated hierarchical B-splines (THB-splines) surfaces instead of tensor-product surfaces, which provided again a significant improvement of the fitted patch.

A further improvement of the fitting results could be achieved by performing an optimization of the parameters also, using methods such as parameter correction [31]. This is beyond the scope of this thesis. Last, but not least, it might be possible to generalize the theoretical observations made in Chapter 5 to hierarchical splines, based on the recent results of [60] on quasi-interpolation operators for THB-splines.

List of Figures

3.1 Multi-patch domain with two patches Ω^1, Ω^2, one interface e and geometry mappings G^1, G^2. 26

3.2 Multi-patch domain with two patches Ω^1, Ω^2 and geometry mappings G^1, G^2. The mappings L and R are the parameterizations of the interface e, i.e., the restriction of the geometry mappings to its preimage. 30

3.3 Multi-patch domain with geometry maps G^1 and G^2, their restrictions L and R to the preimages of the interface and reparameterizations λ and ϱ 31

3.4 Exact splitting of a knot span and application of a quadrature rule to each subsegment 33

3.5 Patch and its control net. Left: matching parameterizations at the interface. Right: non-matching parameterizations at the interface. . 36

3.6 Matching parameterizations at the interface, convergence behaviour of error in different norms: L^2 norm (blue curve), dG norm (red curve). 37

3.7 Influence of the quadrature rule. Left: Convergence behavior of the error in L^2 norm. Right: Convergence behavior of the error in dG norm. Blue and red curves: 10 and 30 uniform segments per t-knot span. Green curves: exact splitting of the knot spans. Yellow curves: adaptive quadrature. Note that the yellow curve coincides with the green one for smaller values of h. Exact representation of the reparameterizations λ and ϱ. 38

3.8 Influence of the reparameterization. Adaptive quadrature on interface integrals. Left: Convergence behaviour of the error in L^2 norm. Right: Convergence behaviour of the error in dG norm. Blue curves: Exact representation of λ and ϱ. Red curves: Approximation error of $\varrho \approx 0.0131167$. Green curves: Approximation error of $\varrho \approx 3.10616 \cdot 10^{-15}$ 40

3.9 Splitting points created by adaptive quadrature - see text for details. 41

3.10 Number of quadrature knots and root finding operations needed by exact and adaptive quadrature for increasingly finer discretizations. 42

4.1 Two patch domain Ω parameterized by a bicubic geometry map G. The knot vectors are given by $[-1,-1,-1,-1,0,0,0,1,1,1,1] \times [0,0,0,0,1,1,1,1]$. 45

4.2 Bicubically parameterized domain (see Fig. 4.1) and transparent plot of the exact solution $3xy\exp(-x)\sin(\pi y)$. 54

4.3 Least squares approximation with functions in $G_h^{1,\varepsilon}$ with $\varepsilon = 0.5$. Left: relative L^2 and H^1 error for the approximate solution of degree 3. Right: L^2 error values at the finest discretization step (scaled by factor 100). 55

4.4 Least squares approximation with functions in $\hat{G}_h^{1,\varepsilon}$ with $\varepsilon = 0.5$. Relative L^2 (top left) and H^1 (top right) error of the solution to the fitting problem on the bi-cubic domain, see Figure 4.1. Bottom: Patch-wise representation of the solution with 8,840 basis functions of degree three for $\varepsilon = 0.5$ with flat shading. 56

4.5 Poisson problem (4.25). Left: Domain Ω and its control net. Middle: Approximate solution with 2,048 basis functions of degree 2. Right: Patch-wise plot of the solution with flat shading. 57

4.6 Poisson problem (4.25). Relative L^2 (left) and H^1 error (right) of the approximate solution with basis functions of different degrees in $\hat{G}_{h,0}^{1,\varepsilon}$ for $\varepsilon = 0.5$. 58

4.7 Biharmonic equation (4.26): Domain with its control net (left), approximate solution with 2.101 basis functions of degree four in $\hat{G}_h^{1,\varepsilon}$ with $\varepsilon = h^2$ (middle) and patch-wise plot of the approximate solution with flat shading (right). 59

4.8 Biharmonic equation (4.26): Relative L^2 (top left), H^1 (top right) and H^2 (bottom) errors of the approximate solution for basis functions of degree three in $\hat{G}_{h,\mathbf{0}}^{1,\varepsilon}$ for four choices of ε. 60

4.9 Biharmonic equation (4.26): Relative L^2 (top left), H^1 (top right) and H^2 (bottom) errors of the approximate solution for basis functions of degree four in $\hat{G}_{h,\mathbf{0}}^{1,\varepsilon}$ for four choices of ε. 61

4.10 Biharmonic equation (4.26): Relative L^2 (top left), H^1 (top right) and H^2 (bottom) errors of the numeric solution for basis functions of degree five in $\hat{G}_{h,\mathbf{0}}^{1,\varepsilon}$ for three choices of ε. 62

4.11 Number of interface basis functions under uniform h-refinement on the domain shown in Figure 4.1. Basis functions of degree three (top left), four (top right), five (bottom left) and six (bottom right) for various mesh-dependent choices of ε. 64

5.1 Turbine component (top left) and point cloud: Parametric (top right) and measured physical data (bottom right) of the fillet belonging the central part of the marked area of the turbine blade (bottom left). 69

5.2 One- (left) and two-dimensional (right) synthetic data for the numerical tests. 77

5.3 Synthetic curve data: Point (left) and normal (right) error for various choices of weight γ and mesh size h. 78

5.4 Synthetic surface data: Point (left) and normal (right) error for various choices of weight γ and mesh size h. 78

5.5 Synthetic curve data: Point (left) and normal (right) error for various choices of the constant γ_0 where $\gamma = \gamma_0 h^2$. 79

5.6 Synthetic surface data: Point (left) and normal (right) error for various choices of the constant γ_0 where $\gamma = \gamma_0 h^2$. 79

5.7 Point (left) and normal (right) error for tensor-product (blue) and THB-spline (red) approximations of the fillet data. 85
5.8 Tensor-product spline approximations of the fillet data without (left) and with (right) using normal information. 85
5.9 The details shown in the marked areas from left to right. 85
5.10 THB-spline approximations of the fillet data without (left) and with (right) using normal information. 86
5.11 Control nets of the THB-spline surfaces approximating the fillet data without (left) and with (right) using normal information. . . . 86

List of Tables

5.1 Error values of the THB-spline approximations. 82
5.2 Error values in different ℓ_p norms for approximations of the ellipsoidal patch and fillet data. 82

Bibliography

[1] R. Adams and J. Fournier. *Sobolev Spaces. Second edition.* Pure and Applied Mathematics. Elsevier/Academic Press, Amsterdam, 2003.

[2] M. Aigner and B. Jüttler. Distance regression by Gauss-Newton-type methods and iteratively re-weighted least-squares. *Computing*, 86:73–87, 2009.

[3] Y. Bazilevs, L. B. de Veiga, J. A. Cottrell, T. J. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16:1031 – 1090, 2006.

[4] C. Bernardi, Y. Maday, and A. T. Patera. A new nonconforming approach to domain decomposition: The mortar element method. In H. Brezis and J.-L. Lions, editors, *Nonlinear partial differential equations and their applications. Collège de France Seminar, volume XI*, pages 13 – 51. Pitman, 1994.

[5] C. Bernardi, Y. Maday, and F. Rapetti. Basics and some applications of the mortar element method. *GAMM-Mitteilungen*, 28:97 – 123, 2005.

[6] K. Birner, B. Jüttler, and A. Mantzaflaris. Bases and dimension for $\mathcal{C}^1$-smooth isogeometric splines on volumetric two-patch domains. *Graphical Models*, 99:46 – 56, 2018.

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

[8] D. Brujic, I. Ainsworth, and M. Ristic. Fast and accurate NURBS fitting for reverse engineering. *International Journal of Advanced Manufacturing Technology*, 54(5-8):691–700, 2011.

[9] F. Brunero. Discontinuous Galerkin methods for isogeometric analysis. Master's thesis, Università degli Studi di Milano, 2012.

[10] 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.

[11] C. L. Chan, C. Anitescu, and T. Rabczuk. Isogeometric analysis with strong multipatch C^1-coupling. *Computer Aided Geometric Design*, 62:294 – 310, 2018.

[12] B. Cockburn. Discontinuous Galerkin methods. *Journal of Applied Mathematics and Mechanics*, 83:731–754, 2003.

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

[14] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[15] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis. Toward Integration of CAD and FEA.* John Wiley and Sons, Chichester, England, 2009.

[16] C. de Boor. On local linear functionals that vanish at all B-splines but one. In A. Law and B. Sahney, editors, *Theory of Approximation with Applications*, pages 120–145, New York, 1976. Academic Press.

[17] C. de Boor and G. Fix. Spline approximation by quasiinterpolants. *Journal of Approximation Theory*, 8:19–45, 1973.

[18] J. Delgado and J. M. Peña. A comparison of different progressive iteration approximation methods. In M. Dæhlen et al., editors, *Mathematical Methods for Curves and Surfaces*, pages 136–152, Berlin, 2010. Springer.

[19] C. Deng and H. Lin. Progressive and iterative approximation for least squares B-spline curve and surface fitting. *CAD Computer Aided Design*, 47:32–44, 2014.

[20] P. Dierckx. *Curve and Surface Fitting with Splines*. Monographs on Numerical Analysis. Oxford Science Publications, 1995.

[21] M. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 157–186, Berlin, 2005. Springer.

[22] S. Flöry. Fitting curves and surfaces to point clouds in the presence of obstacles. *Computer Aided Geometric Design*, 26(2):192–202, 2009.

[23] A. Gálvez, A. Iglesias, and J. Puig-Pey. Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction. *Information Sciences*, 182(1):56–76, 2012.

[24] W. Gander and W. Gautschi. Adaptive quadrature - revisited. *BIT Numerical Mathematics*, 40(1):84 – 101, 2000.

[25] C. Giannelli, B. Jüttler, S. Kleiss, A. Mantzaflaris, B. Simeon, and J. Špeh. THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Comp. Meth. Appl. Mech. Engrg.*, 299:337–365, 2016.

[26] D. Groisser and J. Peters. Matched G^k-constructions always yield C^k-continuous isogeometric elements. *Computer Aided Design*, 34:67–72, 2015.

[27] C. Grossmann. Penalties, Lagrange multipliers and Nitsche mortaring. *Differential Inclusions, Control and Optimization*, 20:205–220, 2010.

[28] D. Hansford and G. Farin. Curve and surface reconstruction. In G. Farin, J. Hoschek, and M.-S. Kim, editors, *Handbook of Computer Aided Geometric Design*, chapter 7, pages 165–192. Elsevier, 2002.

[29] C. Hofer, U. Langer, and I. Toulopoulos. Discontinuous Galerkin isogeometric analysis of elliptic diffusion problems on segmentations with gaps. *SIAM Journal on Scientific Computing*, 38(6):3430–3460, 2016.

[30] C. Hofer and I. Toulopoulos. Discontinuous Galerkin isogeometric analysis of elliptic problems on segmentations with non-matching interfaces. *Computers and Mathematics with Applications*, 72:1811 – 1827, 2016.

[31] J. Hoschek. Intrinsic parametrization for approximation. *Computer Aided Geometric Design*, 5(1):27–31, 1988.

[32] B. Jüttler and A. Felis. Least-squares fitting of algebraic spline surfaces. *Advances in Computational Mathematics*, 17(1-2):135–152, 2002.

[33] M. Kapl, F. Buchegger, M. Bercovier, and B. Jüttler. Isogeometric analysis with geometrically continuous functions on planar multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 316:209 – 234, 2017.

[34] M. Kapl, F. Buchegger, M. Bercovier, and B. Jüttler. Isogeometric analysis with geometrically continuous functions on planar multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 316:209 – 234, 2017.

[35] M. Kapl, G. Sangalli, and T. Takacs. Dimension and basis construction for analysis-suitable G^1 two-patch parameterizations. *Computer Aided Geometric Design*, 52:75 – 89, 2017.

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

[37] M. Kapl and V. Vitrih. Space of C^2-smooth geometrically continuous isogeometric functions on planar multi-patch geometries: Dimension and numerical

experiments. *Computers & Mathematics with Applications*, 73:2319 – 2338, 2017.

[38] M. Kapl and V. Vitrih. Space of C^2-smooth geometrically continuous isogeometric functions on two-patch geometries. *Computers & Mathematics with Applications*, 73(1):37 59, 2017.

[39] M. Kapl and V. Vitrih. Dimension and basis construction for C^2-smooth isogeometric spline spaces over bilinear-like G^2 two-patch parameterizations. *Journal of Computational and Applied Mathematics*, 335:289 – 311, 2018.

[40] M. Kapl, V. Vitrih, B. Jüttler, and K. Birner. Isogeometric analysis with geometrically continuous functions on two-patch geometries. *Computers & Mathematics with Applications*, 70(7):1518 – 1538, 2015.

[41] P. Kiciak. Geometric continuity of curves and surfaces. *Synthesis Lectures on Visual Computing*, 8(3):1–249, 2016.

[42] Y. Kineri, M. Wang, H. Lin, and T. Maekawa. B-spline surface fitting by iterative geometric interpolation/approximation algorithms. *CAD Computer Aided Design*, 44(7):697–708, 2012.

[43] G. Kiss et al. Adaptive CAD model (re-)construction with THB-splines. *Graphical Models*, 76:273–288, 2014.

[44] U. Langer, A. Mantzaflaris, S. E. Moore, and I. Toulopoulos. Multipatch discontinuous Galerkin isogeometric analysis. In B. Jüttler and B. Simeon, editors, *Isogeometric Analysis and Applications*, pages 1–32, Heidelberg, 2014. Springer. also available as NFN Technical Report No. 18 at www.gs.jku.at.

[45] U. Langer and S. E. Moore. Discontinuous Galerkin isogeometric analysis of elliptic PDEs on surfaces. In T. Dickopf, M. Gander, L. Halpern, R. Krause, and L. Pavarino, editors, *Domain Decomposition Methods in Science and Engineering*, volume 104 of *Lecture Notes in Computational Science and Engineering*, pages 319–326. Springer, 2016.

[46] U. Langer and I. Toulopoulos. Analysis of multipatch discontinuous Galerkin IgA approximations to elliptic boundary value problems. *Computing and Visualization in Science*, 17(5):217–233, 2016.

[47] H. Lin, G. Wang, and C. Dong. Constructing iterative non-uniform B-spline curve and surface to fit data points. *Science in China, Series F: Information Sciences*, 47(3):315–331, 2004.

[48] W. Ma. Subdivision surfaces for CAD - an overview. *CAD Computer Aided Design*, 37(7):693–709, 2005.

[49] A. Mantzaflaris and B. Jüttler. Integration by interpolation and look-up for Galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:373 – 400, 2015.

[50] M. Marinov and L. Kobbelt. Optimization methods for scattered data approximation with subdivision surfaces. *Graphical Models*, 67(5):452–473, 2005.

[51] M. Milroy, C. Bradley, G. Vickera, and D. Weir. G^1 continuity of B-spline surface patches in reverse engineering. *Computer-Aided Design*, 27:471–478, 1995.

[52] V. P. Nguyen, P. Kerfriden, M. Brino, S. P. Bordas, and E. Bonisoli. Nitsche's method for two and three dimensional NURBS patch coupling. *Compuational Mechanics*, 53(6):1163–1182, 2014.

[53] H. Pottmann, S. Leopoldseder, and M. Hofer. Approximation with active B-spline curves and surfaces. In *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, PG '02, pages 8–25, Washington, DC, USA, 2002. IEEE Computer Society.

[54] B. Rivière. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation.* SIAM, 2008.

[55] L. L. Schumaker. *Spline Functions: Basic Theory.* Wiley, New York, 1981.

[56] A. Seiler, D. Großmann, and B. Jüttler. Spline surface fitting using normal data and norm-like functions. *Computer Aided Geometric Design*, 64:37–49, 2018.

[57] A. Seiler and B. Jüttler. Reparameterization and adaptive quadrature for the isogeometric discontinuous Galerkin method. In M. Floater et al., editors, *Mathematical Methods for Curves and Surfaces*, pages 251–269. Springer, 2017.

[58] A. Seiler and B. Jüttler. Approximately $\mathcal{C}^1$-smooth isogeometric functions on two-patch domains. Technical Report 77, NFN Geometry + Simulation, Johannes Kepler University, Linz, Austria, 2019. www.gs.jku.at.

[59] X. Shi, T. Wang, P. Wu, and F. Liu. Reconstruction of convergent G^1 smooth B-spline surfaces. *Computer Aided Geometric Design*, 21(9):893–913, 2004.

[60] H. Speleers and C. Manni. Effortless quasi-interpolation in hierarchical spaces. *Numerische Mathematik*, 132(1):155–184, 2016.

[61] T. Varady. Reverse engineering of geometric models - an introduction. *Computer-aided design*, 29(4):255–268, 1997.

[62] M. F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM Journal on Numerical Analysis*, 15(1):152–161, 1978.

[63] W.-C. Xie, X.-F. Zou, J.-D. Yang, and J.-B. Yang. Iteration and optimization scheme for the reconstruction of 3D surfaces based on non-uniform rational B-splines. *CAD Computer Aided Design*, 44(11):1127–1140, 2012.

[64] F. Zhang, Y. Xu, and F. Chen. Discontinuous Galerkin methods for isogeometric analysis for elliptic equations on surfaces. *Communications in Mathematics and Statistics*, 2(3):431–461, 2014.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe. Die vorliegende Dissertation ist mit dem elektronisch übermittelten Textdokument identisch.

Linz, Jänner 2019

....

Agnes Seiler

Curriculum Vitae

Personal Details

Name	Agnes Seiler, MSc.
Date of birth	19.08.1991
Place of birth	Speyer, Germany
Nationality	German

Education

2015 - present	Doctoral Program "Computational Mathematics", Johannes Kepler University Linz, Austria
2013- 2015	Master's Degree Program of Mathematics at Technical University Kaiserslautern, Germany
2010-2013	Bachelor's Degree Program of Mathematics at Technical University Kaiserslautern, Germany
2010	High School Diploma (Abitur)

Research Stays

Jun - Dec 2016	exchange researcher in EXAMPLE Project at MTU Aero Engines AG, Munich, Germany

Conferences

June 28 - July 4, 2018	Curves and Surfaces 2018, Arcachon, France
September 17-21, 2017	Second Conference on Subdivision, Geometric and Algebraic Methods, Isogeometric Analysis and Refinability in Italy, Gaeta, Italy
June 23 - 28, 2016	9th International Conference of Mathematical Methods for Curves and Surfaces, Tonsberg, Norway

Publications

- A. Seiler and B. Jüttler. Reparameterization and adaptive quadrature for the isogeometric discontinuous Galerkin method. Mathematical Methods for Curves and Surfaces, 2017, M. Floater eds, Springer, LNCS 10521, pp 251 – 269.

- A. Seiler, D. Großmann, and B. Jüttler. Spline surface fitting using normal data and norm-like functions. Computer Aided Geometric Design, 64:37 – 49, 2018.

- A. Seiler and B. Jüttler. Approximately C^1-smooth isogeometric functions on two-patch domains. NFN Technical Report No. 77, 2019.