JOHANNES KEPLER
UNIVERSITÄT LINZ | JKU

Technisch-Naturwissenschaftliche
Fakultät

Enhancing Isogeometric Analysis by a Finite Element-Based Local Refinement Strategy

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplomingenieur

im Masterstudium

Industriemathematik

Eingereicht von:
Stefan K. Kleiss

Angefertigt am:
Institut für Numerische Mathematik

Beurteilung:
A. Univ.-Prof. Dipl.-Ing. Dr. Walter Zulehner

Mitwirkung:
Univ.-Prof. Dr. Bert Jüttler

Linz, Juni, 2010

Abstract

In isogeometric analysis, Non-Uniform Rational B-Spline (NURBS) basis functions are used for preserving an exact geometry representation, as well as for representing the discrete solution calculated by finite element methods. While NURBS basis functions are defined over a rectangular grid that does not allow locally restricted refinement, the introduction of T-splines has provided methods for adapting basis functions in such a way that local refinement is possible. In worst-case examples, however, even the use of T-splines can lead to an unwanted, almost global refinement.

We combine the exact geometry representation achieved by NURBS geometry mappings with classic finite element basis functions that can be refined locally more easily: We define subdivisions of the parameter domain that contain hanging nodes, and on these subdivisions, we define piecewise polynomial finite elements basis functions that are globally continuously differentiable. These basis functions are transformed to the physical domain by a global NURBS geometry mapping. Thereby, we preserve the exact representation of the geometry even through the course of refinement without the need for updating geometry information, while the use of hanging nodes allows the refinement to be locally restricted.

We investigate how the properties of the NURBS geometry mapping influence the properties of the finite element basis functions on the physical domain, and how a state-of-the-art a posteriori error estimator can be combined with these specific basis functions. The findings are illustrated in numerical examples.

Acknowledgements

First of all, I want to express my gratitude to Prof. Zulehner and Prof. Jüttler for the countless discussions and their valuable advice and critique. The great amount of time and energy spent in frequent meetings and their patience is highly appreciated and not taken for granted.

None of this would have been possible without the support and understanding of my parents Wolfgang and Midori Kleiss. Having laid the foundation for what I am today, they have always and still do let me find and go my own way, which I do with the assurance of my parents backing me up at all times.

Thanks are also in place for the colleagues I got to know in my studies and my extracurricular activities. To these people I owe that the years I have spent on campus were well spent and so much more than "just" an academic experience.

Last, but not least, I thank my closest friends, Andreas, Gerald, Lisa, Michael and Marion for being there all the way, and for providing the needed distraction at times, as well as the necessary pressure when my focus was not where it should have been. It is their friendship that I rely on and from which I get the confidence to take on whatever may be waiting for us in the future.

Contents

1 Introduction

1.1 Motivation

Many physical processes from fields such as, for example, heat transport, fluid dynamics, and elasticity can be modelled by partial differential equations. In general, these equations can not be solved analytically and various numerical methods for finding discrete solutions have been developed and researched, among them Finite Element Methods (FEM), which this thesis deals with.

The application of FEM to engineering and design problems obviously requires the interaction with engineers and design software. A typical situation could be imagined as follows: An engineer designs the domain on which a partial differential equation needs to be solved using Computer Aided Design (CAD) software. The geometry is then imported into a FEM-programme, which is applied to calculate a discrete solution. This solution is then exported back to the CAD- or some other programme for further processing by the engineer. These few steps already include several disadvantageous aspects:

Geometry representations in CAD and FEM software have been developed independently over the last decades, and every step of importing or exporting geometry data requires some sort of transformation. This also applies to exporting the solution found by the FEM programme for further processing. One issue is whether these transformations can be carried out automatically and efficiently, or whether manual input is necessary, and if so, to what extent. Another issue is that transforming geometry data is bound to require some approximations, which results in inaccuracies and geometry errors.

In classic FEM, the geometry is represented by polygonal, i.e. straight-sided elements. It is obvious that these elements fail to exactly represent curved geometries and that a good approximation of the geometry might require an undesireably high number of elements. Isoparametric FEM provides a solution for this problem by being able to represent certain piecewise curved geometries. Still, refining complicated parts of the boundary of the computational domain can require costly communication with the CAD programme in order to preserve the best possible geometry representation [8, 9].

A standard technique in CAD is to use Non-Uniform Rational B-Splines (NURBS) as basis functions for the geometry representation. These basis functions do not only allow easy editing in CAD software and an exact representation of circular and other curved geometries, but they also have certain properties that make them suitable as FEM basis functions. This aspect provides the foundation for the *isogeometric approach* presented in [8], in which

these NURBS basis functions are used to represent the discrete solution. In the isogeometric setting, the disadvantages mentioned above are overcome: The geometry is represented exactly by the first, i.e. the coarsest mesh, and the accurate representation is not lost in the course of mesh refinement. Expressing the discrete solution with the same basis functions that are used in CAD programmes also means that the transformation from one representation to the other and vice versa becomes unnecessary, hence no accuary is lost during the communication between CAD- and FEM-programmes.

Another property of NURBS basis functions, however, is that they are defined over knot vectors that form a grid-like structure. This structure does not allow adaptive, locally restricted refinement. To overcome this problem, *T-splines* and *splines over T-meshes* were introduced. As long as specific rule are fulfilled, these allow subdivisions of the rectangular cells of the grid. T-splines can be defined over "local knot vectors" and allow meshes with "T-shaped junctions", similar to hanging nodes in FEM, thus making local refinement possible. These specific rules for refinement with T-splines, however, can require the insertion of unwanted, additional knots. In worst-case examples, this even leads to a snowball-effect where inserted knots trigger the insertion of even more knots, resulting in almost global refinement and thereby jeopardizing the gained advantages [5, 6, 8].

While research in these field is still going on, the aim of this thesis is to combine the following four aspects from the isogeometric and the classic finite element approach:

- Exact representation of the physical domain by using NURBS geometry mappings.
- Piecewise polynomial finite element basis functions.
- A posteriori error estimation for selecting cells that need further refinement.
- The use of hanging nodes for locally restricted refinement.

1.2 Overview

We briefly introduce the kind of partial differential equations that are dealt with, and we repeat some basic aspects of the finite-element-technique in Section 2. There, we also give a raw outline of the differences between the "classic" and the "NURBS-FEM"-concepts.

In Section 3, we introduce the used finite element over subdivisions without hanging nodes, namely the Bogner-Fox-Schmit rectangle. The associated

finite element space provides the foundation on which we construct the finite elements over subdivisions with hanging nodes.

In Section 4, we introduce and define subdivisions with hanging nodes, the associated finite element space and its basis functions, as well as the used local refinement methods.

In Section 5, we briefly introduce NURBS basis functions and NURBS geometry mappings, and then discuss their interaction with the constructed finite element space.

The used a posteriori error estimator is defined in Section 6. The computational examples and their results are given and discussed in Section 7. Finally, a summary is given in Section 8.

2 Problem Formulation and Preliminary Remarks

2.1 Classical and Variational Formulation

We only give a brief overview of how to derive the finite element method from the classical formulation of a partial differential equation. Details can be found in e.g. [4, 9, 14, 15].

The *classical formulation* of a two-dimensional, scalar second-order partial differential equation with mixed or pure Neumann and Dirichlet-boundary conditions reads:

$$\left.\begin{array}{rcll} \text{Find } u, \text{ such that :} & & & \\ -\operatorname{div}(A \operatorname{grad} u) + b^T \operatorname{grad} u + cu & = & f & \text{in } \Omega \subset \mathbb{R}^2 \\ u & = & g_D & \text{on } \Gamma_D \\ (A \operatorname{grad} u) \cdot n & = & g_N & \text{on } \Gamma_N \end{array}\right\} \tag{1}$$

with the coefficients

$$A(x) = \begin{pmatrix} A_1(x) & 0 \\ 0 & A_2(x) \end{pmatrix}, \quad b(x) = \begin{pmatrix} b_1(x) \\ b_2(x) \end{pmatrix},$$

where $n = (n_1, n_2)^T$ is the outer normal unit vector of Ω and $\Gamma = \partial\Omega = \overline{\Gamma}_N \cup \overline{\Gamma}_D$ with $\Gamma_N \cup \Gamma_D = \emptyset$.

A function u (from a proper function space) that fulfills (1) is called *classial solution*. The *variational formulation*, or *weak formulation* is derived from the classic formulation by multiplying the partial differential equation with a test function v in a proper space V_0, integrating both sides over the computational domain Ω, and using partial integration.

Let $V = H^1(\Omega)$, $V_0 = \{v \in V : v|_{\Gamma_D} = 0\}$, $V_g = \{v \in V : v|_{\Gamma_D} = g_D\}$. The variational formulation of (1) reads:

$$\left.\begin{array}{rcll} \text{Find } u \in V_g, \text{ such that :} & & & \\ a(u, v) & = & \langle F, v \rangle & \forall\, v \in V_0 \end{array}\right\} \tag{2}$$

where the bilinear form $a(\cdot, \cdot)$ and the linear form F are defined as follows:

$$a(u, v) := \int_\Omega \Big((\nabla v)^T A \nabla u + (b^T \nabla u) v + cuv \Big)\, dx \tag{3}$$

$$\langle F, v \rangle := \int_\Omega fv\, dx + \int_{\Gamma_N} g_N v\, ds. \tag{4}$$

A function u that fulfills (2) is called *weak solution*. The following two properties of $a(\cdot, \cdot)$ are important for the existence and uniqueness of a weak solution:

A bilinear form $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ is called *coercive*, if there exists a constant $\mu_1 > 0$ such that

$$\mu_1 \|v\|_V^2 \ \leq \ a(v,v), \quad \forall\, v \in V.$$

It is called *bounded*, if there exists a constant $\mu_2 > 0$ such that

$$|a(u,v)| \ \leq \ \mu_2 \|u\|_V \|v\|_V, \quad \forall\, u,v \in V.$$

Conditions that imply the coercivity and boundedness of a bilinear form $a(\cdot,\cdot)$ as in (3) are briefly discussed in Appendix A.

Theorem 2.1. (Lax-Milgram) *Let V be a Hilbert space, let the bilinear form $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ be bounded and coercive, and let $F : V \to \mathbb{R}$ be a bounded linear form, i.e. $F \in V^*$. Then, the varational problem (2) has a unique solution $u \in V$ and*

$$\frac{1}{\mu_2}\|F\|_{V^*} \ \leq \ \|u\|_V \ \leq \ \frac{1}{\mu_1}\|F\|_{V^*}. \tag{5}$$

Proof. See [4]. □

For a bounded, coercive bilinear form $a(\cdot,\cdot)$, we can define the *energy norm*:

$$\|v\|_E \ := \ \sqrt{a(v,v)}.$$

In Appendix A, we verify that this definition fulfills the properties of a norm. Due to the boundedness and coercivity of $a(\cdot,\cdot)$, we have

$$\mu_1 \|v\|_V^2 \leq \|v\|_E^2 = a(v,v) \leq \mu_2 \|v\|_V^2,$$

therefore $\|\cdot\|_E$ is equivalent to $\|\cdot\|_V$. This also implies that $a(\cdot,\cdot)$ is bounded in the energy norm:

$$|a(u,v)| \ \leq \ \frac{\mu_2}{\mu_1}\|u\|_E \|v\|_E, \quad \forall\, u,v \in V.$$

2.2 Galerkin Discretization

We choose finite-dimensional subspaces $V_h \subset V$, $V_{0h} \subset V_h$ with $V_{0h} \subset V_0$, and $V_{gh} = \{u \in V_h : u = g_h + v_{0h},\ g_h \in V_h, v_{0h} \in V_{0h}\} \subset V_g$. Then, the following finite-dimensional problem can be derived from the continuous problem (2):

$$\left.\begin{array}{rcl} \text{Find } u_h \in V_{gh}, \text{ such that :} & & \\ a(u_h, v_h) & = & \langle F, v_h \rangle \quad \forall\, v_h \in V_{0h} \end{array}\right\} \tag{6}$$

A function u_h fulfilling (6) is called *discrete solution*. Since V_h is chosen such that $V_h \subset V$, V_h is called *conforming*. In this setting, if the standard assumptions of the theorem of Lax-Milgram are fulfilled in V for the continuous problem, they are also fulfilled in the finite-dimensional subspace V_h. Existence and uniqueness of a discrete solution again follow from the theorem of Lax-Milgram. Under the same standard assumptions, the Lemma of Cea gives the following estimate for the discretization error:

$$\|u - u_h\|_V \quad \leq \quad \frac{\mu_2}{\mu_1} \inf_{v_h \in V_h} \|u - v_h\|_V$$

Choosing a basis $\{\varphi^{(i)} : i = 0, \ldots, N_h\}$ of V_h, one can express $u_h \in V_h$ as a linear combination of the basis functions with coefficients $(u^{(i)})_{i=1,\ldots,N_h}$:

$$u_h(x) = \sum_{i=1}^{N_h} u^{(i)} \varphi^{(i)}(x). \tag{7}$$

By requiring the condition in (6) to be fulfilled for all basis functions $\varphi^{(i)}$, we arrive at N_h equations that can be written as:

$$K_h \underline{u}_h \quad = \quad \underline{f}_h \tag{8}$$

where

$$\begin{aligned} K_h &= (K_{ij})_{i,j=1,\ldots,N_h}, \quad K_{ij} = a(\varphi^{(j)}, \varphi^{(i)}), \\ \underline{u}_h &= (u^{(i)})_{i=1,\ldots,N_h}, \\ \underline{f}_h &= (f_i)_{i=1,\ldots,N_h}, \qquad f_i = \langle F, \varphi^{(i)} \rangle - a(g_h, \varphi^{(i)}). \end{aligned}$$

Thus, the problem of solving (6) has been transformed to the problem of solving the system of linear equations (8). The matrix K_h in (8) is called *stiffness matrix*.

It is standard to construct the basis functions such that they fulfill

$$\varphi^{(i)}(x^{(j)}) = \delta_{ij}. \tag{9}$$

This way, the coefficients of the solution vector $\underline{u}_h$ are also the function values at the nodes:

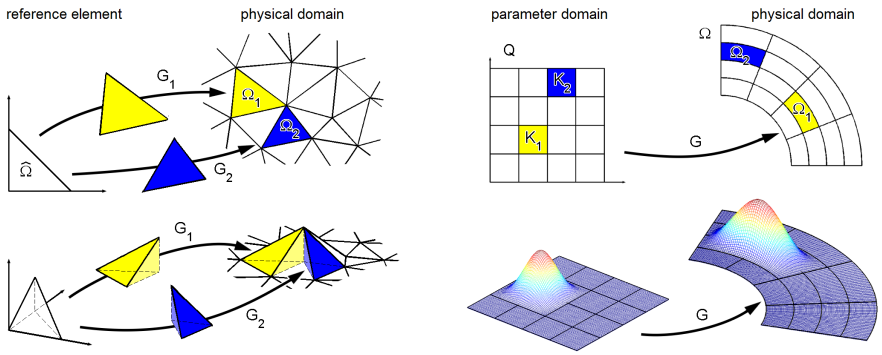
$$u_h(x^{(i)}) = u^{(i)}. \tag{10}$$

If any basis function $\varphi^{(i)}$ has common support with only a few other basis functions $\varphi^{(j)}$, then most entries of the stiffness matrix $K_{ij} = a(\varphi^{(j)}, \varphi^{(i)})$ are zero, i.e. K_h is a sparse matrix. If $a(\cdot, \cdot)$ is symmetric, K_h is symmetric as well.

2.3 Basis Functions of Finite-Element-Space

In classic FEM, the physical domain Ω is divided into subdomains Ω_k, which are obtained by element-wise defined mappings of a single reference element $\widehat{\Omega}$, i.e. $\Omega_k = G_k(\widehat{\Omega})$. The basis functions over the physical space are determined as geometric transformations of shape functions $\widehat{\varphi}^{(i)}$ which are defined over the reference element (see Fig. 1):

$$x \in \Omega_k : \ \varphi^{(i)}(x) = \widehat{\varphi}_{i_k}(G_k^{-1}(x)).$$



(a) Classic FEM (b) Using NURBS-mapping

Figure 1: Illustration of mapping of geometry and shape functions in classic FEM and when using NURBS.

When using NURBS for representing the geometry, the whole parameter domain Q and its cells are mapped by a single mapping G, i.e. $\Omega = G(Q)$. The same then applies to our basis functions: Basis functions are globally defined over the parameter space and transformed by G: $\varphi^{(i)}(x) = \widehat{\varphi}^{(i)}(G^{-1}(x))$. The equation equivalent to (9) in the parameter space is

$$\widehat{\varphi}^{(i)}(\zeta^{(k)}) \ = \ \delta_{ik}, \qquad \text{where } \zeta^{(k)} := G^{-1}(x^{(k)})$$

We will define our basis functions in the parameter space first, leaving the issue of transformation to the physical space for later.

To construct the basis functions $\varphi^{(i)}$, we will actually combine the "classic-FEM-" and the "NURBS-mapping"-approaches: Basis functions $\widehat{\varphi}^{(i)}$ in the parameter space will be defined piecewise by transforming functions from a reference element $\widehat{K}$, as it is done in classic FEM. These resulting functions, which are defined over the whole parameter domain Q, are then mapped to the physical space Ω by the one global geometry mapping.

In the following Sections 3 and 4, we will only discuss the construction and the properties of basis functions over the parameter domain. The issue of transforming them to the physical domain, i.e. the geometry function G and its influence on the shape functions, will be discussed in Section 5.

Note that, in our case, $\widehat{K}$ and Q actually describe the same domain, namely the unit square. We will, however, use these two different notations to emphasize the different context in which they are used, and also because one could also use other parameter domains Q for the NURBS-mapping than the unit square.

3 Finite-Element-Space without Hanging Nodes

3.1 Subdivision of Parameter Domain without Hanging Nodes

We subdivide the parameter domain $Q = [0,1]^2$ into open, axis-aligned rectangles as illustrated in Fig. 2.

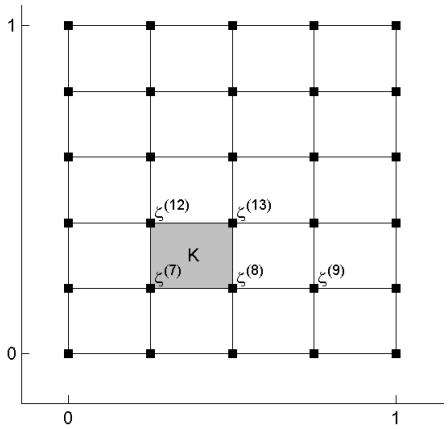


Figure 2: Example of a subdivision of $Q = [0,1]^2$ with rectangular cells and a highlighted cell K. Nodes marked with ■.
$N_E = 20$, $N_X = 30$, $\mathcal{I}_K = \{7, 8, 12, 13\}$.

These rectangles K of the form $K = (a_1, b_1) \times (a_2, b_2)$, where $a_1 < b_1$ and $a_2 < b_2$, will be referred to as *cells*. The total number of cells will be denoted by N_E.

The vertices of the cells (i.e. the corners of the cells) will be referred to as *nodes*. The total number of nodes will be denoted by N_X, the set of their indices by $\mathcal{I}_X = \{1, \ldots, N_X\}$. Nodes in the parameter domain will be denoted by $\zeta^{(i)}$, $i \in \mathcal{I}_X$.

Nodes are called *boundary nodes*, if they are contained in the boundary ∂Q of the parameter domain, and *inner nodes* otherwise.

We define an *edge* as the straight and axis-aligned line that connects two nodes. In a more formal way: If two nodes $\zeta^{(i)}$ and $\zeta^{(j)}$, $\zeta^{(i)} \neq \zeta^{(j)}$ have the same ξ_1- or the same ξ_2-coordinate, then we define the *edge* $E^{(i,j)}$ as follows:

$$E^{(i,j)} := \{\xi \in Q : \ \xi = (1-\alpha) \cdot \zeta^{(i)} + \alpha \cdot \zeta^{(j)}, \ \alpha \in [0,1]\}.$$

We call $\zeta^{(i)}$ and $\zeta^{(j)}$ the *vertices* of the edge $E^{(i,j)}$.

If the nodes $\zeta^{(i)}$ and $\zeta^{(j)}$ have different ξ_1- and ξ_2-coordinates, $E^{(i,j)}$ is defined as the empty set. If an edge contains no node in its interior, we call it an *elementary edge*.

For example, in Fig. 2, we have

$$\begin{aligned} E^{(7,9)} &= E^{(7,8)} \cup E^{(8,9)} \\ E^{(7,8)} &\dots \text{elementary edge} \\ E^{(7,12)} &\dots \text{elementary edge} \\ E^{(7,13)} &= \emptyset \end{aligned}$$

If two cells K and K' with $K \neq K'$ contain the same elementary edge in their closure, then K and K' are called *adjacent*.

To simplify descriptions, we call an edge *horizontal*, if it is parallel to the ξ_1-axis, and *vertical*, if it is parallel to the ξ_2-axis.

For convenience, we will sometimes use the informal terms “north”, “east”, “south” and “west” to indicate directions or positions. These terms refer to the directions $(0,1)$, $(1,0)$, $(0,-1)$ and $(-1,0)$, respectively.

The whole subdivision will be denoted by $\mathcal{K}$. For fixed cell K, the set of indices of the vertices of K will be denoted by $\mathcal{I}_K$ (see Fig. 2 for an example). Note that $|\mathcal{I}_K| = 4$ for all $K \in \mathcal{K}$.

3.2 Finite Element Space over the Parameter Domain

Beforehand, we give two definitions that will enable us to shorten some repeatedly used formulations:

Definition 3.1. Let $v : Q \to \mathbb{R}$ be a function with existing first and mixed derivatives. We write:

$$\partial_0 v := v \qquad \partial_1 v := \frac{\partial v}{\partial \xi_1} \qquad \partial_2 v := \frac{\partial v}{\partial \xi_2} \qquad \partial_3 v := \frac{\partial^2 v}{\partial \xi_1 \partial \xi_2}$$

Definition 3.2. A repeatedly used set of indices shall be denoted as follows:

$$I_\alpha \;:=\; \{0,1,2,3\}.$$

We also define the following sets of functions:

Definition 3.3. Let $\mathcal{K}$ be a subdivision of Q. We denote the set of bi-cubic polynomials over a cell $K \in \mathcal{K}$ by

$$\mathcal{Q}_3(K) := \Big\{ u : K \to \mathbb{R} \;\Big|\; u(\xi_1,\xi_2) = \sum_{s,t=0}^{3} c_{st} \xi_1^s \xi_2^t,\ c_{st} \in \mathbb{R}. \Big\},$$

the set of piecewise bi-cubic functions over a subdivision $\mathcal{K}$ by

$$\mathcal{Q}_3(\mathcal{K}) := \Big\{ u : \ u|_K \in \mathcal{Q}_3(K), \ \forall\, K \in \mathcal{K} \Big\},$$

and the set of piecewise bi-cubic and globally C^1-continuous functions by

$$\mathcal{Q}_3^1(\mathcal{K}) := C^1(\overline{Q}) \cap \mathcal{Q}_3(\mathcal{K}).$$

We choose the latter as the finite element space over the parameter domain:

$$X_h \quad := \quad \mathcal{Q}_3^1(\mathcal{K}). \tag{11}$$

A function $v \in \mathcal{Q}_3(K)$ is uniquely determined by prescribing function value, first and mixed derivatives of v at the four corners of K. In other words: Given 16 values $\lambda_\beta^{(j)} \in \mathbb{R}$, $(j, \beta) \in \mathcal{I}_K \times I_\alpha$, there exists exactly one bi-cubic polynomial $v \in \mathcal{Q}_3(K)$ that fulfills

$$\partial_\beta v(\zeta^{(j)}) = \lambda_\beta^{(j)}$$

for all $(j, \beta) \in \mathcal{I}_K \times I_\alpha$. In particular, if all $\lambda_\beta^{(j)}$ are zero, then $v \equiv 0$ on K.

A direct consequence is that a function $v \in \mathcal{Q}_3(\mathcal{K})$ is piecewise uniquely determined (and therefore uniquely determined over Q) by prescribing function value, first and mixed derivatives at all nodes $\zeta^{(i)} \in \mathcal{K}$. The following theorem shows that such a function is in X_h:

Theorem 3.4. *Let $u \in \mathcal{Q}_3(\mathcal{K})$ be piecewise determined by prescribing function value, first and mixed derivatives at all nodes $\zeta^{(i)}$, $i \in \mathcal{I}_X$. Then*

$$u \quad \in \quad C^1(\overline{Q})$$

holds [4].

Proof. (See Fig. 3 for reference.) Let K_1 and K_2 be two adjacent cells in $\mathcal{K}$, and let $E := \overline{K_1} \cap \overline{K_2}$ be their common elementary edge. Without loss of generality, we assume that E is vertical, i.e. parallel to the ξ_2-axis. The two vertices $\zeta^{(p)}$ and $\zeta^{(q)}$ of E can thus be written as

$$\zeta^{(p)} = \left(\bar{\xi}_1, \xi_2^p\right), \qquad \zeta^{(q)} = \left(\bar{\xi}_1, \xi_2^q\right).$$

Let u be a function in $\mathcal{Q}_3(\mathcal{K})$. This means that the restrictions of u to the cells $u_i := u|_{\overline{K_i}}$, $i \in \{1, 2\}$, are bi-cubic polynomials in ξ_1 and ξ_2. Furthermore, we uniquely determine u_i by prescribing function value, first and mixed derivatives at the vertices of K_i. The prescribed values at $\zeta^{(p)}$ and $\zeta^{(q)}$ shall be denoted by $\lambda_\alpha^{(k)}$, $(k, \alpha) \in \{p, q\} \times I_\alpha$, i.e. we have:

$$\partial_\alpha u_i(\zeta^{(k)}) \quad = \quad \lambda_\alpha^{(k)}, \quad i \in \{1, 2\}. \tag{12}$$

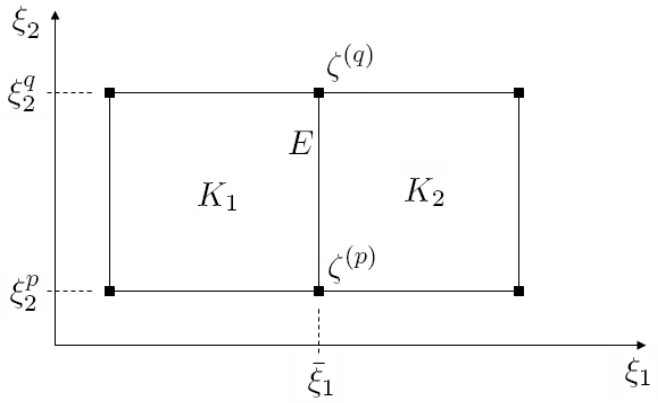


Figure 3: Adjacent cells with common vertical elementary edge.

In order to investigate u at the common edge of the two adjacent cells, we look at the difference between the functions u_i at E:

$$v \;:=\; u_1|_E - u_2|_E \;=\; u_1(\bar{\xi}_1,\cdot) - u_2(\bar{\xi}_1,\cdot).$$

v is a cubic polynomial in ξ_2 with the following properties that follow from (12):

$$\begin{aligned}
v(\zeta^{(p)}) &= u_1(\zeta^{(p)}) - u_2(\zeta^{(p)}) = \lambda_0^{(p)} - \lambda_0^{(p)} = 0\\
v'(\zeta^{(p)}) &= \frac{\partial v}{\partial \xi_2}(\zeta^{(p)}) = \frac{\partial u_1}{\partial \xi_2}(\zeta^{(p)}) - \frac{\partial u_2}{\partial \xi_2}(\zeta^{(p)}) = \lambda_2^{(p)} - \lambda_2^{(p)} = 0\\
v(\zeta^{(q)}) &= u_1(\zeta^{(q)}) - u_2(\zeta^{(q)}) = \lambda_0^{(q)} - \lambda_0^{(q)} = 0\\
v'(\zeta^{(q)}) &= \frac{\partial v}{\partial \xi_2}(\zeta^{(q)}) = \frac{\partial u_1}{\partial \xi_2}(\zeta^{(q)}) - \frac{\partial u_2}{\partial \xi_2}(\zeta^{(q)}) = \lambda_2^{(q)} - \lambda_2^{(q)} = 0
\end{aligned}$$

From $v(\zeta^{(p)}) = v'(\zeta^{(p)}) = v(\zeta^{(q)}) = v'(\zeta^{(q)}) = 0$, it follows that $v \equiv 0$, i.e. $u_1|_E \equiv u_2|_E$, which shows continuity of u at E.

$v \equiv 0$ also implies $\frac{\partial v}{\partial \xi_2} \equiv 0$. With

$$\frac{\partial v}{\partial \xi_2} \;=\; \frac{\partial}{\partial \xi_2}\Big(u_1(\bar{\xi}_1,\cdot) - u_2(\bar{\xi}_1,\cdot)\Big) \;=\; \frac{\partial u_1}{\partial \xi_2}(\bar{\xi}_1,\cdot) - \frac{\partial u_2}{\partial \xi_2}(\bar{\xi}_1,\cdot)$$

this shows the continuity of $\frac{\partial u}{\partial \xi_2}$ at E. To investigate $\frac{\partial u}{\partial \xi_1}$, we define the following function:

$$w \;:=\; \left.\frac{\partial u_1}{\partial \xi_1}\right|_E - \left.\frac{\partial u_2}{\partial \xi_1}\right|_E \;=\; \frac{\partial u_1}{\partial \xi_1}(\bar{\xi}_1,\cdot) - \frac{\partial u_2}{\partial \xi_1}(\bar{\xi}_1,\cdot)$$

Again, w is a cubic polynomial in ξ_2, and similar to above, we can show:

$$w(\zeta^{(p)}) \;=\; \frac{\partial u_1}{\partial \xi_1}(\zeta^{(p)}) - \frac{\partial u_2}{\partial \xi_1}(\zeta^{(p)}) = \lambda_1^{(p)} - \lambda_1^{(p)} = 0$$

$$
\begin{aligned}
w'(\zeta^{(p)}) &= \frac{\partial w}{\partial \xi_2}(\zeta^{(p)}) = \frac{\partial^2 u_1}{\partial \xi_1 \partial \xi_2}(\zeta^{(p)}) - \frac{\partial^2 u_2}{\partial \xi_1 \partial \xi_2}(\zeta^{(p)}) = \lambda_3^{(p)} - \lambda_3^{(p)} = 0 \\
w(\zeta^{(q)}) &= \frac{\partial u_1}{\partial \xi_1}(\zeta^{(q)}) - \frac{\partial u_2}{\partial \xi_1}(\zeta^{(q)}) = \lambda_1^{(q)} - \lambda_1^{(q)} = 0 \\
w'(\zeta^{(q)}) &= \frac{\partial w}{\partial \xi_2}(\zeta^{(q)}) = \frac{\partial^2 u_1}{\partial \xi_1 \partial \xi_2}(\zeta^{(q)}) - \frac{\partial^2 u_2}{\partial \xi_1 \partial \xi_2}(\zeta^{(q)}) = \lambda_3^{(q)} - \lambda_3^{(q)} = 0
\end{aligned}
$$

Using the same argument as above, it follows that $w \equiv 0$, showing the continuity of $\frac{\partial u}{\partial \xi_1}$ at E. We have thus shown the continuity of u and its first derivatives at E, i.e. $u \in C^1(\overline{K}_1 \cup \overline{K}_2)$. Since E was an arbitrary edge, it follows that $u \in C^1(\overline{Q})$. □

Remark 3.5. The finite element associated with X_h is called *Bogner-Fox-Schmit rectangle*. Since the piecewise bi-cubic functions are continuous over Q and piecewise in $H^1(K)$, we obtain $X_h \subseteq H^1(\Omega) = V$. We even have $X_h \subseteq H^2(\Omega)$, i.e. X_h is suitable for forth-order partial differential equations which, however, are not consider in this thesis. For details, see [4].

3.3 Basis Functions of Finite Element Space

For a fixed $(i, \alpha) \in \mathcal{I}_X \times I_\alpha$, we define the piecewise bi-cubic function $\widehat{\varphi}_\alpha^{(i)}$ through the following $4 \cdot N_X$ conditions:

$$
\left.
\begin{aligned}
\widehat{\varphi}_\alpha^{(i)}(\zeta^{(j)}) &= \delta_{ij}\delta_{\alpha 0} \\
\frac{\partial \widehat{\varphi}_\alpha^{(i)}}{\partial \xi_1}(\zeta^{(j)}) &= \delta_{ij}\delta_{\alpha 1} \\
\frac{\partial \widehat{\varphi}_\alpha^{(i)}}{\partial \xi_2}(\zeta^{(j)}) &= \delta_{ij}\delta_{\alpha 2} \\
\frac{\partial^2 \widehat{\varphi}_\alpha^{(i)}}{\partial \xi_1 \partial \xi_2}(\zeta^{(j)}) &= \delta_{ij}\delta_{\alpha 3}
\end{aligned}
\right\} \quad \forall\, j \in \mathcal{I}_X. \tag{13}
$$

Using Definitions 3.1 and 3.2, we can write (13) more compactly:

$$
\partial_\beta\, \widehat{\varphi}_\alpha^{(i)}(\zeta^{(j)}) = \delta_{(i,\alpha)(j,\beta)} \quad \forall\, (j,\beta) \in \mathcal{I}_X \times I_\alpha, \tag{14}
$$

where

$$
\delta_{(i,\alpha)(j,\beta)} := \begin{cases} 1, & \text{if } i = j \wedge \alpha = \beta \\ 0, & \text{else.} \end{cases}
$$

This means we require value, first and mixed derivatives of $\widehat{\varphi}_\alpha^{(i)}$ to be 0 at all nodes, with the only exception of the "α-derivative" at node $\zeta^{(i)}$, which

is set to 1 (see Fig. 4 for an example). Each basis function $\widehat{\varphi}_\alpha^{(i)}$ is associated both with a node and with a "type of derivative", which motivates the use of the double indices.

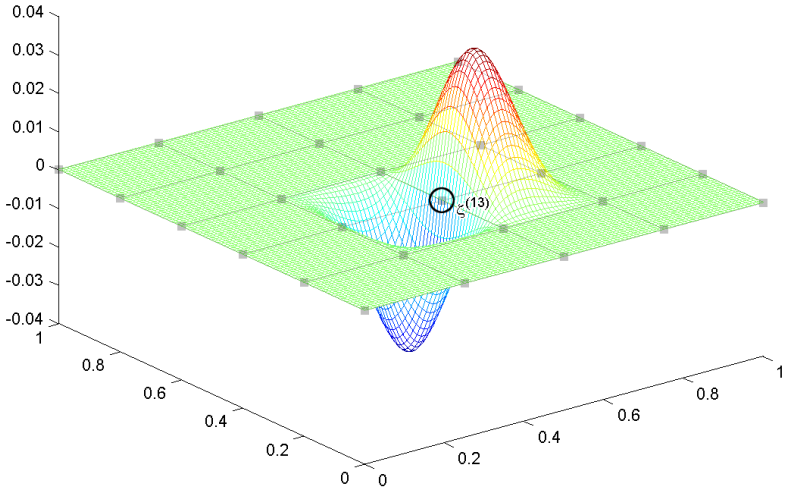


Figure 4: Example for a basis function: $\widehat{\varphi}_1^{(13)}$ over the mesh from Fig. 2: $\partial_1 \widehat{\varphi}_1^{(13)}(\zeta^{(13)}) = 1$, all other function values and derivatives at the subdivision's nodes are zero.

The conditions (14) imply that, if $\zeta^{(i)} \notin \mathcal{I}_K$, then $\widehat{\varphi}_\alpha^{(i)} \equiv 0$ on K. Thus, each function $\widehat{\varphi}_\alpha^{(i)}$ has support only in those four cells that have $\zeta^{(i)}$ as a vertex (if $\zeta^{(i)}$ is a boundary node, $\widehat{\varphi}_\alpha^{(i)}$ has support in less than four cells).

We will show that the functions defined by (14) form a basis of our finite element space.

Lemma 3.6. *The $4 \cdot N_X$ piecewise bi-cubic functions $\widehat{\varphi}_\alpha^{(i)}$, $(i, \alpha) \in \mathcal{I}_X \times I_\alpha$ are linearly independent.*

Proof. Let u be a linear combination of functions $\widehat{\varphi}_\alpha^{(i)}$:

$$u \;:=\; \sum_{i=1}^{N_X} \sum_{\alpha=0}^{3} \lambda_\alpha^{(i)} \widehat{\varphi}_\alpha^{(i)}, \quad \lambda_\alpha^{(i)} \in \mathbb{R}$$

We have to show:

$$\Big(\forall\, (i,\alpha) \in \mathcal{I}_X \times I_\alpha :\; \lambda_\alpha^{(i)} = 0\Big) \Leftrightarrow \Big(u \equiv 0\Big)$$

The implication is trivial, so we show the other direction: Assume $u \equiv 0$, which implies

$$0 \;\equiv\; \partial_0 u \;\equiv\; \partial_1 u \;\equiv\; \partial_2 u \;\equiv\; \partial_3 u. \tag{15}$$

Assume that there exist indices (i_0, α_0) such that there is a coefficient $\lambda_{\alpha_0}^{(i_0)} \neq 0$. Then, due to the property $\partial_\beta \widehat{\varphi}_\alpha^{(i)}(\zeta^{(j)}) = \delta_{(i,\alpha)(j,\beta)}$, it follows that

$$\partial_{\alpha_0} u(\zeta^{(i_0)}) \;=\; \lambda_{\alpha_0}^{(i_0)} \;\neq\; 0,$$

which contradicts (15). Thus, all coefficients $\lambda_\alpha^{(i)}$ have to be zero and the functions $\widehat{\varphi}_\alpha^{(i)}$ are linearly independent. □

Theorem 3.7. *The set of functions*

$$\Phi \;:=\; \left\{ \widehat{\varphi}_\alpha^{(i)},\ (i,\alpha) \in \mathcal{I}_X \times I_\alpha \right\} \tag{16}$$

defined by (14) is a basis of X_h.

Proof. From Theorem 3.4, it follows that $\widehat{\varphi}_\alpha^{(i)} \in X_h$ for all $(i,\alpha) \in \mathcal{I}_X \times I_\alpha$. Let u be an arbitrary function in X_h. Set $u_\alpha^{(i)} := \partial_\alpha u(\zeta^{(i)})$ for all $(i,\alpha) \in \mathcal{I}_X \times I_\alpha$, and define

$$v \;:=\; u - \sum_{i=1}^{N_X} \sum_{\alpha=0}^{3} u_\alpha^{(i)} \widehat{\varphi}_\alpha^{(i)}.$$

Because of (14) and the definition of $u_\alpha^{(i)}$, it follows that $\partial_\alpha v(\zeta^{(i)}) = 0$ for all (i,α), i.e. $v \equiv 0$. Thus, u can be expressed as a linear combination of functions $\widehat{\varphi}_\alpha^{(i)}$. Because of Lemma 3.6, the functions $\widehat{\varphi}_\alpha^{(i)}$ are linearly independent, therefore Φ is a basis of X_h. □

Remark 3.8. A direct conclusion is that the dimension of X_h is $4 \cdot N_X$.

Analogous to (7), we can express the discrete solution u_h as a linear combination of basis functions:

$$u_h(x) = \sum_{i=1}^{N_X} \sum_{\alpha=0}^{3} u_\alpha^{(i)} \varphi_\alpha^{(i)},$$

where $\varphi_\alpha^{(i)}$ is the transformation of $\widehat{\varphi}_\alpha^{(i)}$ under the global geometry mapping G. The property analogous to (10) now reads:

$$u_h(x^{(i)}) = u_0^{(i)}.$$

3.4 Representation of Basis Functions

We represent the basis functions $\widehat{\varphi}_\alpha^{(i)}$ as it is done in classic FEM: We define basis functions $\widehat{p}_\alpha^{(i)}$ on a reference element $\widehat{K}$ and transform them piecewise to the cells $K \in \mathcal{K}$.

3.4.1 Basis Functions on the Reference Element

Let $\zeta^{(i)}$, $i \in \{1,2,3,4\}$ benote the four corners of the unit square $\widehat{K} := [0,1]^2$:

$$\zeta^{(1)} := (0,0), \quad \zeta^{(2)} := (0,1), \quad \zeta^{(3)} := (1,0), \quad \zeta^{(4)} := (1,1). \tag{17}$$

For any fixed $(i,\alpha) \in \{1,2,3,4\} \times I_\alpha$, we prescribe sixteen conditions for the bi-cubic polynomial function $\widehat{p}_\alpha^{(i)}(\xi_1,\xi_2) = \sum_{s,t=0}^{3} c_{st}\xi_1^s\xi_2^t$, $c_{st} \in \mathbb{R}$:

$$\partial_\beta \widehat{p}_\alpha^{(i)}(\zeta^{(j)}) \quad = \quad \delta_{(i,\alpha)(j,\beta)} \quad \forall\,(j,\beta) \in \{1,2,3,4\} \times I_\alpha. \tag{18}$$

For each of the 16 combinations $(i,\alpha) \in \{1,2,3,4\} \times I_\alpha$, the system of linear equations (18) with the unknowns c_{st} can be solved uniquely. The resulting values for the coefficients c_{st} define explicit formulae for $\widehat{p}_\alpha^{(i)}$ (see Fig. 5 for an example).

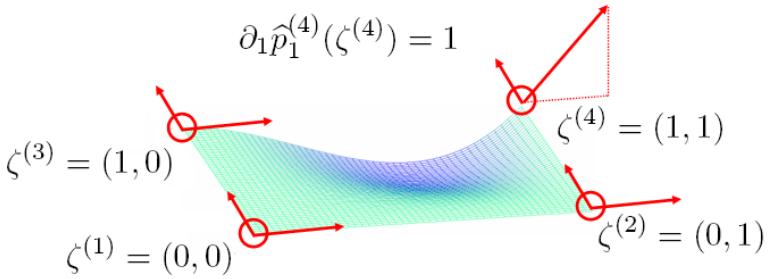


Figure 5: Illustration of $\widehat{p}_1^{(4)}$: Values and derivatives are zero, except for the derivative with respect to ξ_1 at the node $\zeta^{(4)}$.

Definition 3.9. From now on, whenever we refer to functions $\widehat{p}_\alpha^{(i)}$ with $(i,\alpha) \in \{1,2,3,4\} \times I_\alpha$, we will be referring to the sixteen explicitly known polynomials defined by (17) and (18) on the unit square.

These functions $\widehat{p}_\alpha^{(i)}$ are linearly independent, thus forming a basis of the 16-dimensional space $\mathcal{Q}_3(\widehat{K})$.

3.4.2 Local Representation of Basis Functions

Let $K = (a_1,b_1) \times (a_2,b_2)$ with $a_1 < b_1$, $a_2 < b_2$ be a cell of our subdivision $\mathcal{K}$, and let $\mathcal{I}_K$ denote the set of indices of the vertices of K. Let $\widehat{G}$ denote the linear mapping that maps the reference element $\widehat{K}$ to K:

$$\begin{array}{rccc} \widehat{G}: & \widehat{K} & \to & \overline{K} \\ & \begin{pmatrix} \widehat{\xi}_1 \\ \widehat{\xi}_2 \end{pmatrix} & \mapsto & \begin{pmatrix} (1-\widehat{\xi}_1)a_1 + \widehat{\xi}_1 b_1 \\ (1-\widehat{\xi}_2)a_2 + \widehat{\xi}_2 b_2 \end{pmatrix} \end{array}$$

and let $h_1 := b_1 - a_1 > 0$ and $h_2 := b_2 - a_2 > 0$.

We link the globally numbered vertices of K to the vertices of the reference element $\widehat{K}$ with the function $c_i : \mathcal{I}_K \to \{1, 2, 3, 4\}$. In our axis-aligned cells, we define c_i through the positions of the nodes in the cell K:

$$c_i \quad := \quad \begin{cases} 1, & \text{if } \zeta^{(i)} = (a_1, a_2) \quad \dots \text{“southwest”} \\ 2, & \text{if } \zeta^{(i)} = (b_1, a_2) \quad \dots \text{“southeast”} \\ 3, & \text{if } \zeta^{(i)} = (a_1, b_2) \quad \dots \text{“northwest”} \\ 4, & \text{if } \zeta^{(i)} = (b_1, b_2) \quad \dots \text{“northeast”} \end{cases}$$

Note that $\widehat{G}$, a_1, a_2, b_1, b_2, h_1, h_2, and c_i depend on the cell K, which should be indicated by adding K as a parameter. For the sake of readability, and assuming that the context sufficiently clarifies which cell is being referred to, we will not explicitly add K when using these variables.

For any fixed $(i, \alpha) \in \mathcal{I}_X \times I_\alpha$ we represent $\widehat{\varphi}_\alpha^{(i)} : Q \to \mathbb{R}$ piecewise as follows:

$$\left.\begin{array}{lrcl} \text{if } (\xi_1, \xi_2) \in \overline{K} \ \wedge \ i \in \mathcal{I}_K : & & & \\ & \widehat{\varphi}_0^{(i)}(\xi_1, \xi_2) & = & \widehat{p}_0^{(c_i)}\left(\widehat{G}^{-1}(\xi_1, \xi_2)\right) \\ & \widehat{\varphi}_1^{(i)}(\xi_1, \xi_2) & = & h_1 \cdot \widehat{p}_1^{(c_i)}\left(\widehat{G}^{-1}(\xi_1, \xi_2)\right) \\ & \widehat{\varphi}_2^{(i)}(\xi_1, \xi_2) & = & h_2 \cdot \widehat{p}_2^{(c_i)}\left(\widehat{G}^{-1}(\xi_1, \xi_2)\right) \\ & \widehat{\varphi}_3^{(i)}(\xi_1, \xi_2) & = & h_1 \cdot h_2 \cdot \widehat{p}_3^{(c_i)}\left(\widehat{G}^{-1}(\xi_1, \xi_2)\right) \\ \text{otherwise :} & & & \\ & \widehat{\varphi}_\alpha^{(i)}(\xi_1, \xi_2) & = & 0, \qquad \forall\, \alpha \in I_\alpha \end{array}\right\} \tag{19}$$

Note that, if a point is contained in the closure of two different cells, i.e. if $(\xi_1^*, \xi_2^*) \in \overline{K} \cap \overline{K'}$, $K \neq K'$, the definition does not specify in which cell $\widehat{\varphi}_\alpha^{(i)}(\xi_1^*, \xi_2^*)$ should be evaluated. From Theorem 3.4, we know that the functions $\widehat{\varphi}_\alpha^{(i)}$ are continuous at the edge of two adjacent cells, thus $\widehat{\varphi}_\alpha^{(i)}(\xi_1^*, \xi_2^*)$ can be evaluated either in $\overline{K}$ or in $\overline{K'}$.

Definition 3.10. The Jacobian of a function $F : Q \to \mathbb{R}^2$ will be denoted by $\nabla_\xi F$, i.e.

$$\nabla_\xi F \ := \ \begin{pmatrix} \partial F_1/\partial \xi_1 & \partial F_1/\partial \xi_2 \\ \partial F_2/\partial \xi_1 & \partial F_2/\partial \xi_2 \end{pmatrix}.$$

Using the properties of $\widehat{p}_\alpha^{(i)}$ and

$$\nabla_\xi \left(\widehat{G}^{-1}\right) \ = \ \begin{pmatrix} \frac{1}{h_1} & 0 \\ 0 & \frac{1}{h_2} \end{pmatrix},$$

one can show by direct calculation that the given representation of $\widehat{\varphi}_\alpha^{(i)}$ fulfills

$$\partial_\beta \widehat{\varphi}_\alpha^{(i)}(\zeta^{(j)}) \;=\; \delta_{(i,\alpha)(j,\beta)}, \qquad \forall\,(j,\beta) \in \mathcal{I}_X \times I_\alpha.$$

This local representation allows cell-wise assembling of K_h and the right-hand-side $\underline{f}_h$ in (8), as it is usually done in FEM.

3.5 Incorporation of Boundary Conditions

Neumann boundary conditions are incorporated the same way as it is done in standard FEM: For each elementary edge contained in the Neumann-boundary Γ_N, we calculate its contribution to the integral

$$\int_{\Gamma_N} g_N \varphi_\alpha^{(i)}\, ds \;=\; \sum_{E^{(p,q)} \subseteq \Gamma_N} \int_{E^{(p,q)}} g_N \varphi_\alpha^{(i)}\, ds.$$

We incorporate Dirichlet boundary counditions

$$u \;=\; g_D \quad \text{on } \Gamma_D$$

by applying them to the coefficients $u_\alpha^{(i)}$ corresponding to nodes $x^{(i)}$ on the Dirichlet boundary Γ_D [9]. We approximate the function g_D with a function that is in V_h, or, after transforming g_D to the parameter domain with G^{-1}, with a function in X_h. We interpolate $\widehat{g}_D := g_D \circ G$ by piecewise cubic polynomials that are determined by prescribing function value and first tangential derivative in the direction of the boundary at all nodes contained in the Dirichlet boundary.

To distinguish between the gradients in the parameter domain and the phyiscal domain, we write

$$\nabla_\xi \;:=\; \left(\frac{\partial}{\partial \xi_1}, \frac{\partial}{\partial \xi_2}\right)^T \quad \text{and} \quad \nabla_x \;:=\; \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}\right)^T.$$

Let $\widehat{\Gamma}_S$ be the "southern" boundary of Q and let Γ_S be its image in the physical domain:

$$\begin{aligned} \widehat{\Gamma}_S &:= \Big\{(\xi_1, 0),\ \xi_1 \in [0,1]\Big\} \\ \Gamma_S &:= G\Big(\widehat{\Gamma}_S\Big). \end{aligned}$$

We briefly discuss the incorporation of Dirichlet boundary conditions for the special case of $\Gamma_D \subseteq \Gamma_S$. Other cases can be treated analogously.

Γ_S can be interpretet as a two-dimensional curve with the parametrization $G(\cdot, 0)$ and parameter $\xi_1 \in [0, 1]$. The tangent vector of Γ_S at a point $G(\xi_1, 0)$ is given by

$$\frac{\partial}{\partial \xi_1} G(\xi_1, 0) \quad = \quad \begin{pmatrix} J_{11}(\xi_1, 0) \\ J_{21}(\xi_1, 0) \end{pmatrix}$$

where $J := \nabla_\xi G$ denotes the Jacobian of G. The derivative of g_D in the tangential direction of Γ_S is given by

$$\begin{pmatrix} J_{11} & J_{21} \end{pmatrix} \begin{pmatrix} \frac{\partial g_D}{\partial x_1} \\ \frac{\partial g_D}{\partial x_2} \end{pmatrix} \quad = \quad \Big(J^T (\nabla_x g_D) \Big)_1 .$$

Since $(J^T \nabla_x)_1 = \frac{\partial}{\partial \xi_1}$, we obtain

$$\Big(J^T (\nabla_x g_D(x^{(i)})) \Big)_1 \quad = \quad \frac{\partial}{\partial \xi_1} \widehat{g}_D(\xi(x^{(i)})),$$

where $\xi(x^{(i)}) := G^{-1}(x^{(i)})$ and where $\widehat{g}_D(\xi) = g_D(G(\xi))$. Let

$$\widehat{u}_h(\xi) \quad := \quad \sum_{i \in \mathcal{I}_P} \sum_{\alpha=0}^{3} u_\alpha^{(i)} \widehat{\varphi}_\alpha^{(i)}(\xi)$$

denote the approximate solution u_h transformed to the parameter domain. We want $\widehat{u}_h$ to approximate the Dirichlet boundary conditions, i.e.

$$\widehat{u}_h|_{\widehat{\Gamma}_S} \;\approx\; \widehat{g}_D,$$

which we realize by prescribing function value and the derivative in ξ_1-direction of $\widehat{g}_D$ at the nodes on $\widehat{\Gamma}_S$. Together with the definition of the basis functions $\widehat{\varphi}_\alpha^{(i)}$, this means we prescribe

$$\begin{aligned} \widehat{g}_D(\xi(x^{(i)})) \quad &= \quad \widehat{u}_h(\xi(x^{(i)})) \quad = u_0^{(i)} \\ \frac{\partial}{\partial \xi_1} \widehat{g}_D(\xi(x^{(i)})) &= \frac{\partial}{\partial \xi_1} \widehat{u}_h(\xi(x^{(i)})) = u_1^{(i)}. \end{aligned}$$

In more general settings, i.e. if not necessarily $\Gamma_D \subseteq \Gamma_S$, we can make analogous considerations. In summary, we conclude that, in order to incorporate Dirichlet boundary conditions, we have to prescribe

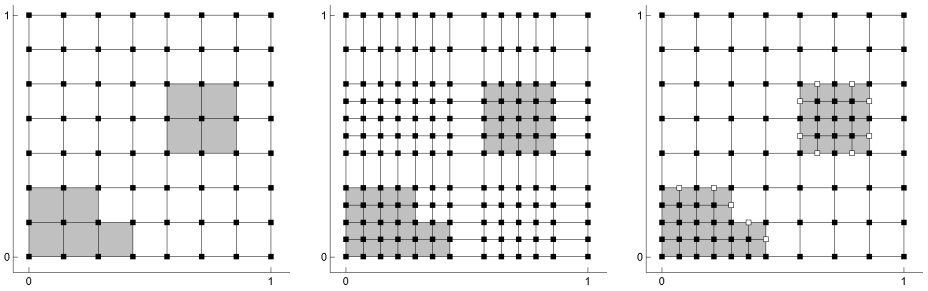
$$u_0^{(i)} \;=\; g_D(x^{(i)}) \quad \text{and} \quad u_\beta^{(i)} \;=\; \Big(J^T (\nabla_x g_D(x^{(i)})) \Big)_\beta$$

for all $x^{(i)} \in \Gamma_D$, where $\beta = 1$, if the edge containing $\xi(x^{(i)})$ is horizontal, and $\beta = 2$, if the edge is vertical.

4 Finite-Element-Space with Hanging Nodes

Decreasing the size of the cells in our subdivision leads to a better approximation of the exact solution, but also increases the computational effort needed for assembling and solving the discrete problem. Especially if large errors only appear in small areas of the domain, it would be preferable to refine only the cells in those areas.

The question of how to use an a posteriori error estimator to identify the cells with the highest contributions to the global error will be discussed in Section 6. For the time being, we assume that we have already found those areas of interest and marked them for refinement as illustrated in Fig. 6(a).



(a) Shaded cells marked for refinement.

(b) Unwanted refinement of unmarked ares.

(c) Local refinement with hanging nodes.

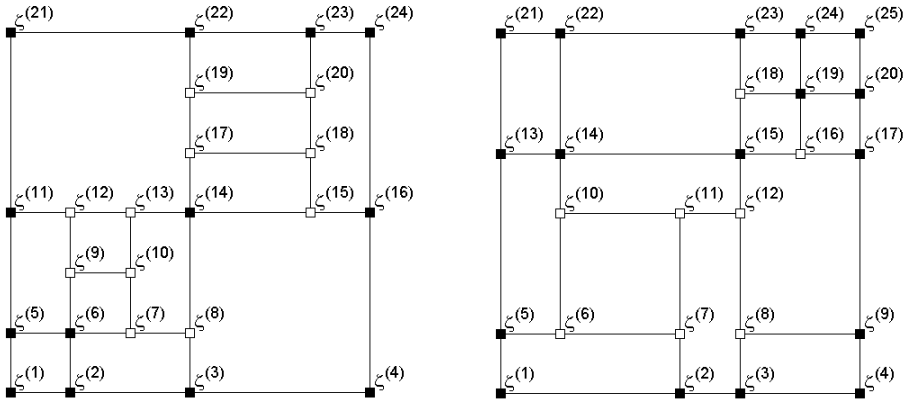
Figure 6: Example comparing the refinement of small areas of the domain without and with hanging nodes.

Simply using a subdivision with axis-aligned rectangles as discussed in Section 3, can lead to an unwanted refinement outside the areas of interest, as illustrated in Fig. 6(b). In order to prevent this, we introduce so-called "hanging nodes" that allow a locally restricted refinement as illustrated in Fig. 6(c).

4.1 Hierarchical Subdivision with Hanging Nodes

As before, we subdivide the parameter domain Q into axis-aligned, rectangular *cells* and require that they form a subdivision of Q (see Fig. 7). *Nodes* are again placed at the corners of each cell and the terms *edge* and *elementary edge* are the same as defined in Section 3.1. The whole subdivision is again referred to by $\mathcal{K}$.

We now distinguish between two kinds of nodes: A node is called *primary node*, if it is a boundary node, or if it is an inner node and the vertex of



(a) (b)

Figure 7: Two examples for subdivisions with hanging nodes:
■ . . . Primary node □ . . . Hanging node.

exactly four elementary edges. The set of indices of all primary nodes will be denoted by $\mathcal{I}_P$, their total number by $N_P := |\mathcal{I}_P|$. A node is called *hanging node*, if it is an inner node and the vertex of exactly three elementary edges. The set of indices of all hanging nodes will be denoted by $\mathcal{I}_H$. The set of indices of all nodes will again be denoted by $\mathcal{I}_X$. Note that all nodes of our subdivision are either primary nodes or hanging nodes, i.e. we have $\mathcal{I}_X = \mathcal{I}_P \cup \mathcal{I}_H$.

In addition to the definition given in Section 3.1, we now require an *edge* to be continuous in that sense that the whole edge must be represented as a line in the graphical representation of the subdivision. More formally speaking: Every point on an edge $E^{(i,j)}$ must either be contained in the boundary of Q or contained in the closure of at least two different cells. If there is no such continuous connection, $E^{(i,j)}$ is defined as the empty set.

As before, an edge is called *elementary edge*, if it contains no node in its interior. We introduce two additional types of edges that are defined as follows:

If the two vertices $\zeta^{(i)}$ and $\zeta^{(j)}$ of the edge $E^{(i,j)}$ both are corners of the same cell, we call $E^{(i,j)}$ a *border edge*.

If an edge $E^{(i,j)}$
- contains at least one hanging node in its interior, and
- contains only hanging nodes in its interior, and
- is maximal,

it is called *principal edge*. With "maximal", we mean that the edge can not be extended beyond $\zeta^{(i)}$ or $\zeta^{(j)}$, because those nodes are primary nodes, or

because there is no continuing edge. The nodes $\zeta^{(i)}$ and $\zeta^{(j)}$ are then called the *parent nodes* of all hanging nodes contained in the interior of $E^{(i,j)}$, and they are also called *parent nodes* of the edge itself. For illustrations of principal edges, see Fig. 8 and Fig. 9.

Some examples for the different types of edges appearing in the subdivision shown in Fig. 7(b) and Fig. 9 are:

$E^{(5,6)}$ ... elementary edge
$E^{(6,7)}$ ... elementary edge
$E^{(5,7)}$ $= E^{(5,6)} \cup E^{(6,7)}$
$E^{(4,7)}$ $= \emptyset$
$E^{(7,9)}$ $= \emptyset$
$E^{(10,12)}$ ... principal edge
$E^{(3,15)}$ ... principal edge
$E^{(8,9)}$ ... border edge
$E^{(8,15)}$ ... border edge
$E^{(15,16)}$ ... border edge and elementary edge
$E^{(6,14)}$ ... border edge and principal edge

Note that in a subdivision without hanging nodes, every node is a primary node, and every elementary edge is a border edge.

To structure the nodes of our subdivision, we assign a *level* to each node and require (L1) and (L2) to be valid:

(L1) The level of a primary node is always 1.
(L2) The level of a hanging node is always higher than the levels of its parent nodes.

We call $\mathcal{K}$ an *hierarchical subdivision*, if $\mathcal{K}$ is a partition of Q consisting of axis-aligned rectangles, and if (L1) and (L2) are fulfilled.

Hanging nodes, their parent nodes, and the levels of those nodes can be depicted in a directed *"node-levels-graph"*. In Fig. 8, this is done for the subdivision in Fig. 7(a): (L1) corresponds to all primary nodes appearing in the topmost level 1. (L2) means that each hanging node points to exactly two parent nodes which are of a lower level.

Fig. 9 shows that the subdivision in Fig. 7(b) is not hierarchical: The circular dependencies of hanging nodes and their parent nodes violate (L2).

Note that a subdivision without hanging nodes always is a hierarchical subdivision.

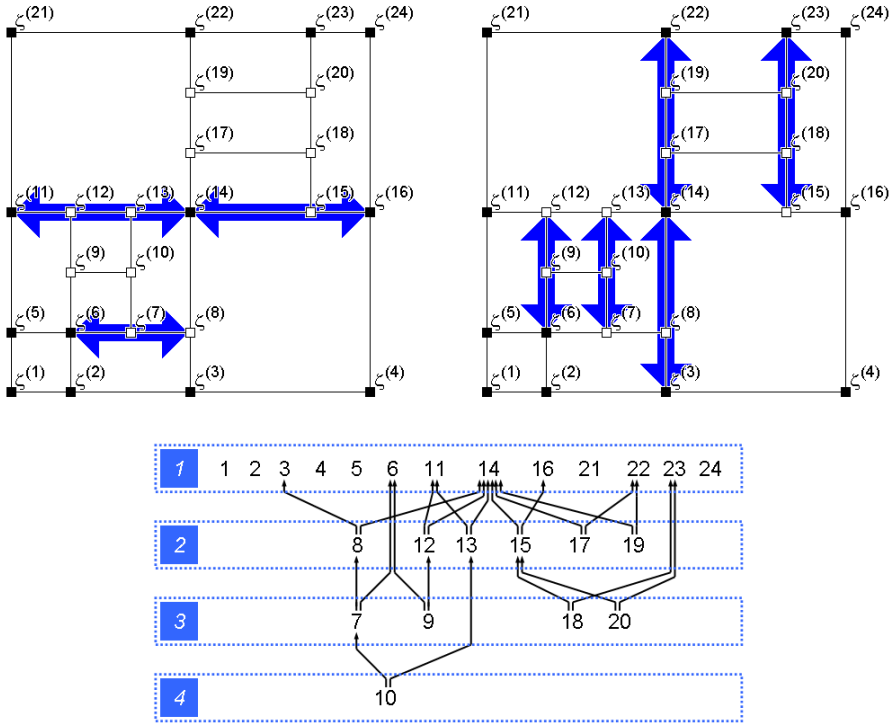


Figure 8: Top: Horizontal (left) and vertical (right) principal edges of the subdivision in Fig. 7(a). Arrowheads pointing to parent nodes of principal edges. Bottom: Node-levels-graph (levels indicated by blue boxes on the left).

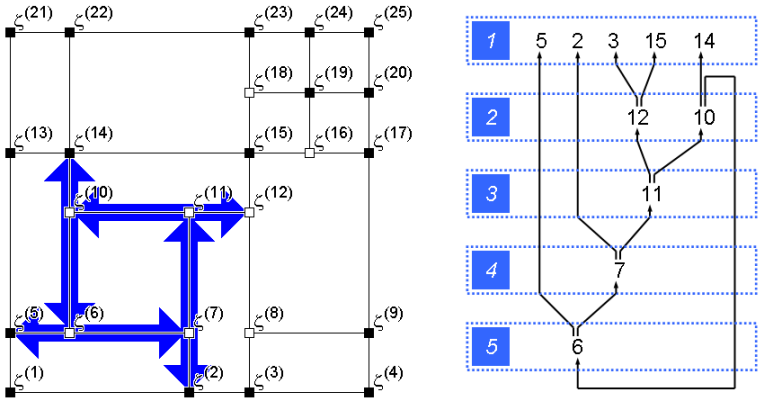


Figure 9: Example of a subdivision violating (L2). Principal edges and node-levels-graph shown only for the part in which the violation occurs.

4.2 Finite Element Space with Hanging Nodes

As in the case without hanging nodes, we choose the space of all piecewise bi-cubic, globally C^1-continuous functions as our finite element space:

$$X_h \quad := \quad \mathcal{Q}_3^1(\mathcal{K}). \tag{20}$$

Now, however, we require $\mathcal{K}$ to be a hierarchical subdivision.

A function $u \in X_h$ is again piecewise uniquely determined by prescribing function value, first and mixed derivatives at all nodes of the subdivision. Different to the situation in Section 3, however, we can not choose those values freely any more.

4.2.1 Illustrative Example regarding C^1-Continuity

An illustrative example is shown in Fig. 10. A subdivision without hanging nodes is depicted in Fig. 10(a). The basis function $\widehat{\varphi}_0^{(i)}$ associated with the node $\zeta^{(i)}$ (in the center, marked by a black circle), is plotted in Fig. 10(b). The function is piecewise defined by prescribing

$$\partial_\beta \widehat{\varphi}_\alpha^{(i)}(\zeta^{(j)}) \quad = \quad \delta_{(i,\alpha)(j,\beta)} \tag{21}$$

at all nodes. As discussed in Section 3, it is globally C^1-continuous.

A subdivision with hanging nodes is depicted in Fig. 10(c). If we piecewise define the basis function $\widehat{\varphi}_0^{(i)}$ by simply prescribing (21) at all nodes again, we obtain the piecewise bi-cubic function plotted in Fig. 10(d). Obviously, this choice of prescribed values leads to discontinuities at some of the edges containing hanging nodes.

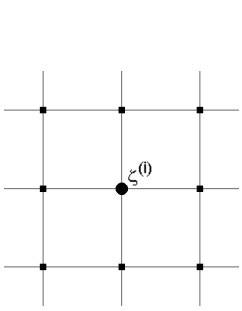
The desired, globally C^1-continuous basis function $\widehat{\varphi}_0^{(i)}$ is shown in Fig. 10(e). As we will discuss in the following subsections, prescribing values at primary nodes and requiring global C^1-continuity already determines the values that have to be prescribed at hanging nodes.

4.2.2 Condition for Global C^1-Continuity

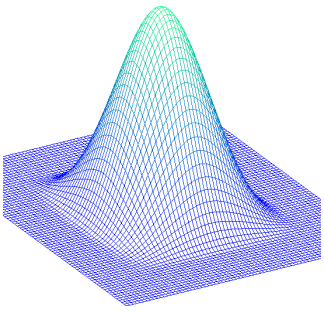
For $i \in \mathcal{I}_X$, let $\mathcal{K}^{(i)}$ be the set of cells that contain $\zeta^{(i)}$ in their closure:

$$\mathcal{K}^{(i)} \quad := \quad \Big\{ K \in \mathcal{K} : \ \zeta^{(i)} \in \overline{K} \Big\}. \tag{22}$$

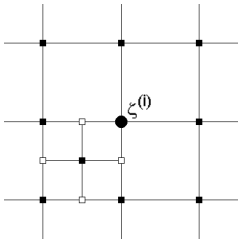
Note that $|\mathcal{K}^{(i)}| = 3$, if $i \in \mathcal{I}_H$. The following condition is essential for global C^1-continuity:



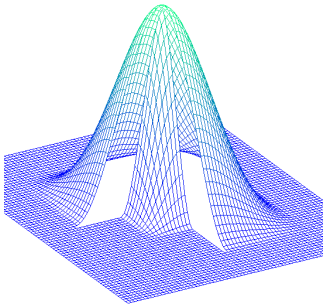
(a) Subdivision without hanging nodes.



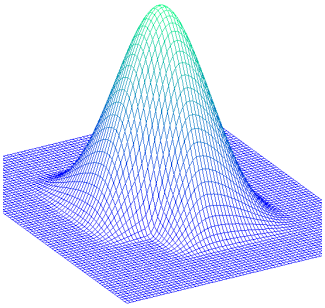
(b) "Delta-condition" at all nodes.



(c) Subdivision with hanging nodes.



(d) "Delta-condition" both at primary and at hanging nodes - discontinuities appear.



(e) "Delta-condition" only at primary nodes. Specially chosen values prescribed at hanging nodes.

Figure 10: Basis function $\widehat{\varphi}_0^{(i)}$ associated with the primary node marked by the black circle, defined over subdivisions with and without hanging nodes. Piecewise definition through "delta-condition" $\partial_\beta \widehat{\varphi}_0^{(i)}(\zeta^{(j)}) = \delta_{(i,0)(j,\beta)}$ and through a method described in the following sections.

Theorem 4.1. *Let $\mathcal{K}$ be a hierarchical subdivision of the parameter domain Q. Let $u \in \mathcal{Q}_3(\mathcal{K})$, and let $u_K := u|_{\overline{K}}$ be its restriction to the closure of a cell $K \in \mathcal{K}$. Then u is C^1-continuous over Q, if and only if the condition*

$$\partial_\alpha u_K(\zeta^{(i)}) \;=\; \partial_\alpha u_{K'}(\zeta^{(i)}), \quad \forall\, \alpha \in I_\alpha,\ \forall\, K, K' \in \mathcal{K}^{(i)}, \tag{23}$$

is fulfilled for all $i \in \mathcal{I}_X$.

Proof. Assume that (23) holds for all $i \in \mathcal{I}_X$. Let E be an elementary edge of the subdivision $\mathcal{K}$ and let $\zeta^{(p)}$ and $\zeta^{(q)}$ be its vertices as illustrated in Fig. 11. Let K and K' be the two cells that contain E in their closure. Since we assume that (23) holds for $i \in \{p, q\}$, we have

$$\left.\begin{aligned} \partial_\alpha u_K(\zeta^{(p)}) &= \partial_\alpha u_{K'}(\zeta^{(p)}) \\ \partial_\alpha u_K(\zeta^{(q)}) &= \partial_\alpha u_{K'}(\zeta^{(q)}) \end{aligned}\right\} \ \forall\, \alpha \in I_\alpha.$$

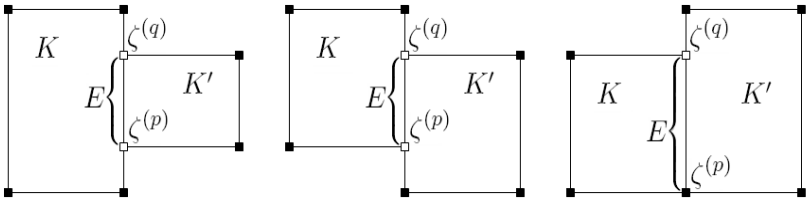


Figure 11: Examples for two neighbouring cells K and K' where $E := \overline{K} \cap \overline{K'}$ is not necessarily a whole border edge of K and K'.

Using the same argument as in the proof of Theorem 3.4, we can then show that u is C^1-continuous over the whole elementary edge E. Since E was an arbitrary edge of our subdivision, we can show that u is C^1-continuous over each elementary edge and thus globally over the whole parameter domain Q.

On the other hand, assume that (23) does not hold for all $i \in \mathcal{I}_X$, i.e. assume that there exist indices $(i_0, \alpha_0) \in \mathcal{I}_X \times I_\alpha$ such that, for two cells $K_1, K_2 \in \mathcal{K}^{(i_0)}$,

$$\partial_{\alpha_0} u_{K_1}(\zeta^{(i_0)}) \neq \partial_{\alpha_0} u_{K_2}(\zeta^{(i_0)}) \tag{24}$$

holds. If $\alpha_0 \in \{0, 1, 2\}$, it is obvious that u can not be C^1-continuous at $\zeta^{(i_0)}$. If $\alpha_0 = 3$, we proceed similar to the proof of Theorem 3.4: Let $E := \overline{K}_1 \cap \overline{K}_2$ be the common elementary edge of K_1 and K_2 containing $\zeta^{(i_0)}$. Without loss of generality, assume that E is a vertical edge and let $\bar{\xi}_1$ be its ξ_1-coordinate. Let $u_j := u|_{\overline{K_j}}$ be the restriction of u to the closure of K_j, $j \in \{1, 2\}$, and define the cubic polynomial $w : E \to \mathbb{R}$ of ξ_2:

$$w(\bar{\xi}_1, \cdot) \quad := \quad \frac{\partial u_1}{\partial \xi_1}\Big|_E - \frac{\partial u_2}{\partial \xi_1}\Big|_E.$$

If u was globally C^1-continuous, then $w \equiv 0$, which implies $w' \equiv 0$, i.e.:

$$\frac{\partial^2 u_1}{\partial \xi_1 \partial \xi_2}\Big|_E - \frac{\partial^2 u_2}{\partial \xi_1 \partial \xi_2}\Big|_E \equiv 0.$$

This contradicts (24), therefore u can not be globally C^1-continuous. □

4.2.3 Prescribing Values at the Nodes of the Subdivision

To determine a piecewise bi-cubic polynomial, we prescribe function value, first and mixed derivative at all nodes of our subdivision. We determine these values through the following inductive method:

Base Case: Values at Nodes of Level 1

We prescribe function value, first and mixed derivatives of our choice at all primary nodes, i.e. at all nodes of level 1.

Induction Assumption: Values at Nodes of Level n

Assume that we have already prescribed value, first and mixed derivatives at all nodes of levels less or equal to $n \in \mathbb{N}$:

$$\begin{aligned} \lambda_\alpha^{(i)} \quad &\dots \quad \text{value prescribed for } \partial_\alpha u \text{ at node } \zeta^{(i)} \\ &\qquad \text{for } i \in \{j \in \mathcal{I}_X : \text{ level of } \zeta^{(j)} \leq n\},\ \alpha \in I_\alpha. \end{aligned}$$

Induction Step: Prescribing Values at Nodes of Level $n+1$

We apply the following procedure to each node $\zeta^{(k)}$ of level $n+1$:

Procedure 4.2. (Prescribing values at hanging nodes)
Let E be the principal edge containing $\zeta^{(k)}$ and let $\zeta^{(p)}$ and $\zeta^{(q)}$ be the parent nodes of E. E can be either horizontal or vertical.

▶ If E is a vertical principal edge:

▷ Step 1: Let $\bar{\xi}_1$ be the ξ_1-coordinate of E. Define the cubic polynomials $v = v(\bar{\xi}_1, \cdot)$ and $w = w(\bar{\xi}_1, \cdot)$ by prescribing the following values:

$$\begin{aligned} &v(\zeta^{(p)}) = \lambda_0^{(p)},\ v'(\zeta^{(p)}) = \lambda_2^{(p)},\ v(\zeta^{(q)}) = \lambda_0^{(q)},\ v'(\zeta^{(q)}) = \lambda_2^{(q)} \\ &w(\zeta^{(p)}) = \lambda_1^{(p)},\ w'(\zeta^{(p)}) = \lambda_3^{(p)},\ w(\zeta^{(q)}) = \lambda_1^{(q)},\ w'(\zeta^{(q)}) = \lambda_3^{(q)} \end{aligned}$$

Since we have a hierarchical subdivision, it holds that the levels of $\zeta^{(p)}$ and $\zeta^{(q)}$ are less or equal n. Thus, by our induction assumption, we know the already prescribed values $\lambda_\alpha^{(p)}$ and $\lambda_\alpha^{(q)}$. The meaning of the polynomials v and w is similar to the meaning in the proof of Theorem 3.4. Now, we use these polynomials to prescribe values at the hanging node $\zeta^{(k)}$ contained in E:

▷ Step 2: Prescribe the following values $\lambda_\alpha^{(k)}$ for $\partial_\alpha u(\zeta^{(k)})$:

$$\begin{aligned} \lambda_0^{(k)} &:= v(\zeta^{(k)}) \\ \lambda_1^{(k)} &:= w(\zeta^{(k)}) \\ \lambda_2^{(k)} &:= v'(\zeta^{(k)}) \\ \lambda_3^{(k)} &:= w'(\zeta^{(k)}) \end{aligned}$$

▶ If E is a horizontal principal edge:

This case is treated analogously:

▷ Step 1: Let $\bar{\xi}_2$ be the ξ_2-coordinate of E. Define the cubic polynomials $v = v(\cdot, \bar{\xi}_2)$ and $w = w(\cdot, \bar{\xi}_2)$ by prescribing the following values:

$$v(\zeta^{(p)}) = \lambda_0^{(p)},\ v'(\zeta^{(p)}) = \lambda_1^{(p)},\ v(\zeta^{(q)}) = \lambda_0^{(q)},\ v'(\zeta^{(q)}) = \lambda_1^{(q)}$$
$$w(\zeta^{(p)}) = \lambda_2^{(p)},\ w'(\zeta^{(p)}) = \lambda_3^{(p)},\ w(\zeta^{(q)}) = \lambda_2^{(q)},\ w'(\zeta^{(q)}) = \lambda_3^{(q)}$$

▷ Step 2: Prescribe the following values $\lambda_\alpha^{(k)}$ for $\partial_\alpha u(\zeta^{(k)})$:

$$\begin{aligned} \lambda_0^{(k)} &:= v(\zeta^{(k)}) \\ \lambda_1^{(k)} &:= v'(\zeta^{(k)}) \\ \lambda_2^{(k)} &:= w(\zeta^{(k)}) \\ \lambda_3^{(k)} &:= w'(\zeta^{(k)}) \end{aligned}$$

Thus, we have prescribe values for function, first and mixed derivatives at all nodes of level $n+1$ and the procedure can be repeated inductively until we have reached the highest level, i.e. until we have prescribed values at all hanging nodes.

4.2.4 Preliminary Considerations regarding C^1-Continuity

Before showing that a piecewise bi-cubic function determined by the method given in Section 4.2.3 is C^1-continuous over Q, we make some preliminary considerations.

For a cell $K \in \mathcal{K}$, we define a "local basis" of $\mathcal{Q}_3(\overline{K})$ analogously to (19):

Definition 4.3. Let $K \in \mathcal{K}$, and let $\widehat{G}$, h_1, h_2, and c_i be as defined in Section 3.4.2. For $(i,\alpha) \in \mathcal{I}_K \times I_\alpha$, define $\widehat{\psi}_\alpha^{(i)} : \overline{K} \to \mathbb{R}$ as follows:

$$\left.\begin{aligned} \widehat{\psi}_0^{(i)}(\xi_1,\xi_2) &:= \widehat{p}_0^{(c_i)}\left(\widehat{G}^{-1}(\xi_1,\xi_2)\right) \\ \widehat{\psi}_1^{(i)}(\xi_1,\xi_2) &:= h_1 \cdot \widehat{p}_1^{(c_i)}\left(\widehat{G}^{-1}(\xi_1,\xi_2)\right) \\ \widehat{\psi}_2^{(i)}(\xi_1,\xi_2) &:= h_2 \cdot \widehat{p}_2^{(c_i)}\left(\widehat{G}^{-1}(\xi_1,\xi_2)\right) \\ \widehat{\psi}_3^{(i)}(\xi_1,\xi_2) &:= h_1 \cdot h_2 \cdot \widehat{p}_3^{(c_i)}\left(\widehat{G}^{-1}(\xi_1,\xi_2)\right) \end{aligned}\right\} \tag{25}$$

and denote the set of these 16 functions by Ψ_K:

$$\Psi_K := \left\{\widehat{\psi}_\alpha^{(i)},\ (i,\alpha) \in \mathcal{I}_K \times I_\alpha\right\}. \tag{26}$$

Note that, like $\widehat{G}$, h_1, h_2 and c_i, the basis functions $\widehat{\psi}_\alpha^{(i)}$ depend on the cell K on which they are defined. For the sake of readability, however, we do not explicitly add K as a parameter, because the context should sufficiently clarify which cell is being referred to.

Certain properties of these functions are summarized in the following Lemma:

Lemma 4.4. ***(a)*** *Ψ_K is a basis of $\mathcal{Q}_3(\overline{K})$.*

(b) *The functions $\widehat{\psi}_\alpha^{(i)}$, $(i,\alpha) \in \mathcal{I}_K \times I_\alpha$ satisfy*

$$\partial_\beta \widehat{\psi}_\alpha^{(i)}(\zeta^{(j)}) \;=\; \delta_{(i,\alpha)(j,\beta)}, \qquad \forall\, (j,\beta) \in \mathcal{I}_K \times I_\alpha.$$

(c) *Let $\zeta^{(k)}$ be a corner of a cell $K \in \mathcal{K}$ and let E_1 and E_2 be the two border edges of K that have $\zeta^{(k)}$ as a vertex. Then*

$$\xi \in \partial K \backslash (E_1 \cup E_2) \;\;\Rightarrow\;\; \partial_\beta \widehat{\psi}_\alpha^{(k)}(\xi) = 0, \quad \forall\, \alpha, \beta \in I_\alpha.$$

Proof. By direct calculation. □

With the basis Ψ_K at hand, we can easily construct or represent a bi-cubic polynomial over K such that the function and its first and mixed derivatives at the corners of K take values of our choice:

Corollary 4.5. *Given 16 values $\lambda_\alpha^{(i)} \in \mathbb{R}$, $(i,\alpha) \in \mathcal{I}_K \times I_\alpha$, the polynomial*

$$v(\xi_1,\xi_2) \;:=\; \sum_{i \in \mathcal{I}_K} \sum_{\alpha \in I_\alpha} \lambda_\alpha^{(i)} \widehat{\psi}_\alpha^{(i)}(\xi_1,\xi_2)$$

is the uniquely defined polynomial in $\mathcal{Q}_3(\overline{K})$ fulfilling

$$\partial_\alpha v(\zeta^{(k)}) \;=\; \lambda_\alpha^{(k)} \quad \forall\, (k,\alpha) \in \mathcal{I}_K \times I_\alpha.$$

Proof. Follows from Lemma 4.4(a) and (b). □

Let $K \in \mathcal{K}$ be an arbitrary, but fixed cell. Let $E := E^{(p,q)}$ be a border edge of K and let $u_K := u|_{\overline{K}}$ be the restriction of a function $u \in X_h$ to the closure of K. As before, we denote the value prescribed for $\partial_\alpha u$ at the node $\zeta^{(i)}$ by $\lambda_\alpha^{(i)}$. We know from Corollary 4.5 and Lemma 4.4(c) that we can write

$$\partial_\beta (u_K|_E) \;=\; \sum_{k \in \mathcal{I}_K} \sum_{\alpha \in I_\alpha} \lambda_\alpha^{(k)} \cdot \partial_\beta \widehat{\psi}_\alpha^{(k)} \;=\; \sum_{k \in \{p,q\}} \sum_{\alpha \in I_\alpha} \lambda_\alpha^{(k)} \cdot \partial_\beta \widehat{\psi}_\alpha^{(k)} \qquad (27)$$

for any $\beta \in I_\alpha$. This formalizes that function value, first and mixed derivatives of $u_K|_E$ only depend on the values, first and mixed derivatives at the vertices of E, but *not* at the other nodes. Even though the size of the cell

is implicitly included in (27) due to the definition of the basis functions $\widehat{\psi}_\alpha^{(i)}$ in (25), one can show by direct calculation that $\partial_\beta(u_K|_E)$ depends on the length of E, but *not* on the size of K in the other coordinate direction. Note that these aspects were already used implicitly in the proof of Theorem 3.4. A consequence that can be easily verified by direct calculation is formulated as a remark for later reference:

Remark 4.6. Let K, $E = E^{(p,q)}$, u_K, and $\lambda_\alpha^{(i)}$ be as defined above. Let $v^{(p,q)}$ and $w^{(p,q)}$ be the univariate cubic polynomials that are determined over E by the values $\lambda_\alpha^{(i)}$, $(i,\alpha) \in \{p,q\} \times I_\alpha$ as in Step 1 of Procedure 4.2. Then $v^{(p,q)}$ and $w^{(p,q)}$ uniquely determine function value, first and mixed derivatives of u_K over the whole border edge E.

The following observation is also formulated as a remark for later reference:

Remark 4.7. Two univariate cubic polynomials f and g are globally equal, if they have the same function value and first derivatives at two different points, i.e. if

$$f(s) = g(s), \quad f'(s) = g'(s), \quad f(t) = g(t), \quad f'(t) = g'(t), \quad \text{for } s \neq t,$$

then $f \equiv g$ globally.

4.2.5 C^1-Continuity over a Hierarchical Subdivision

The following Lemma shows that the piecewise bi-cubic function determined as described in in Section 4.2.3 is globally C^1-continuous:

Lemma 4.8. *Let $u \in \mathcal{Q}_3(\mathcal{K})$ be piecewise defined by prescribing values for function, first and mixed derivatives the corners of each cell $K \in \mathcal{K}$. If those values are determined as described in Section 4.2.3, then u is globally C^1-continuous.*

Proof. From Theorem 4.1, we know that it is sufficient to show that (23) is fulfilled at all nodes of our subdivision. Let $\zeta^{(i)}$ be an arbitrary node of our subdivision and let $\lambda_\alpha^{(i)}$ denote the values prescribed through the described procedure. As before, let $u_K := u|_{\overline{K}}$ denote the restriction of u to the closure of a cell $K \in \mathcal{K}$.

If $\zeta^{(i)}$ is a primary node, the values $\lambda_\alpha^{(i)}$ were chosen freely. A primary node automatically is a corner of all cells $K \in \mathcal{K}^{(i)}$. Hence, if $i \in \mathcal{I}_P$, (23) holds due to the piecewise definition of u.

If $\zeta^{(i)}$ is a hanging node, then $\zeta^{(i)}$ is the corner of two cells and contained in the border edge of a third cell (see Fig. 12 for reference). Let $K_1, K_2 \in \mathcal{K}^{(i)}$

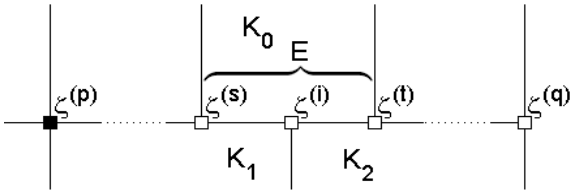


Figure 12: Illustration of three cells containing a hanging node $\zeta^{(i)}$ in their closures.

denote the two cells that have $\zeta^{(i)}$ as a corner. From the definition of u, it directly follows that

$$\partial_\alpha u_{K_1}(\zeta^{(i)}) \;=\; \lambda_\alpha^{(i)} \qquad \text{and} \qquad \partial_\alpha u_{K_2}(\zeta^{(i)}) \;=\; \lambda_\alpha^{(i)} \tag{28}$$

for all $\alpha \in I_\alpha$.

Let $K_0 \in \mathcal{K}^{(i)}$ be the cell that contains $\zeta^{(i)}$ in a border edge E, but does not have $\zeta^{(i)}$ as a corner. Let $\zeta^{(s)}$ and $\zeta^{(t)}$ denote the vertices of E. Let E^* be the principal edge containing $\zeta^{(i)}$ and let $\zeta^{(p)}$ and $\zeta^{(q)}$ be the parent nodes of E^*. E^* also contains $\zeta^{(s)}$ and $\zeta^{(t)}$.

The values $\lambda_\alpha^{(s)}$ and $\lambda_\alpha^{(t)}$ are determined in Step 2 of Procedure 4.2 by the two univariate cubic polynomials v and w that, in turn, are determined over E^* by the values $\lambda_\alpha^{(p)}$ and $\lambda_\alpha^{(q)}$. The values $\lambda_\alpha^{(s)}$ and $\lambda_\alpha^{(t)}$ analogously define two univariate polynomials on E that we denote by $v^{(s,t)}$ and $w^{(s,t)}$. As mentioned in Remark 4.7, we have $v^{(s,t)} \equiv v|_E$ and $w^{(s,t)} \equiv w|_E$. From this, it follows with Remark 4.6 and the fact that the values $\lambda_\alpha^{(i)}$ are also determined by v and w that

$$\partial_\alpha u_{K_0}(\zeta^{(i)}) \;=\; \lambda_\alpha^{(i)}, \quad \forall\, \alpha \in I_\alpha.$$

If $\zeta^{(s)} = \zeta^{(p)}$ or $\zeta^{(t)} = \zeta^{(q)}$ or both, the same argument holds. With (28), this means that, (23) holds for all $K, K' \in \mathcal{K}^{(i)}$, if $i \in \mathcal{I}_H$. Thus, (23) holds for all $i \in \mathcal{I}_P \cup \mathcal{I}_H = \mathcal{I}_X$ which prooves the global C^1-continuity of u. □

4.3 Basis Functions of Finite Element Space with Hanging Nodes

We choose the following modification of condition (14) to define our basis functions over a hierarchical subdivision with hanging nodes:

Theorem 4.9. *For a fixed $(i,\alpha) \in \mathcal{I}_P \times I_\alpha$, let the piecewise bi-cubic function $\widehat{\varphi}_\alpha^{(i)}$ be determined through the $4 \cdot N_P$ conditions*

$$\partial_\beta \widehat{\varphi}_\alpha^{(i)}(\zeta^{(j)}) \;=\; \delta_{(i,\alpha)(j,\beta)}, \quad \forall\, (j,\beta) \in \mathcal{I}_P \times I_\alpha \tag{29}$$

at primary nodes, and by prescribing values at hanging nodes as described in Section 4.2.3. Then the set of functions

$$\Phi \quad := \quad \left\{ \widehat{\varphi}_\alpha^{(i)},\ (i,\alpha) \in \mathcal{I}_P \times I_\alpha \right\} \tag{30}$$

is a basis of X_h.

Proof. From Lemma 4.8, we know that the given definition results in globally C^1-continuous functions $\widehat{\varphi}_\alpha^{(i)}$, thus $\Phi \subseteq X_h$. Linear independence can be shown with the same argument as in Lemma 3.6.

Let f be an arbitrary but fixed function in X_h and let g be defined by

$$g \quad := \quad f - \sum_{i \in \mathcal{I}_P} \sum_{\alpha \in I_\alpha} \mu_\alpha^{(i)} \widehat{\varphi}_\alpha^{(i)}$$

where $\mu_\alpha^{(i)} := \partial_\alpha f(\zeta^{(i)})$ for $i \in \mathcal{I}_P$. Obviously, we have

$$\partial_\alpha g(\zeta^{(i)}) \quad = \quad 0, \quad \forall\, (i,\alpha) \in \mathcal{I}_P \times I_\alpha. \tag{31}$$

Because $g \in X_h$, condition (23) holds for g and all $i \in \mathcal{I}_X$.

Let $E^{(p,q)}$ be a princial edge of the subdivision. As mentioned in Remark 4.6, the values $\partial_\alpha g$, $\alpha \in I_\alpha$ piecewise define univariate polynomials $v^{(i,k)}$ and $w^{(i,k)}$ over the each border edge $E^{(i,k)}$ contained in $E^{(p,q)}$.

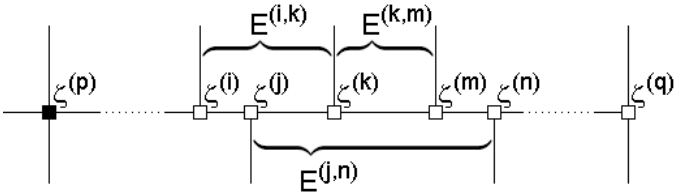


Figure 13: Illustration of overlapping border edges on a principal edge.

An important property of a principal edge is that the border edges contained in it always overlap (as illustrated in Fig. 13). Hence, with Remark 4.7 and (23), it follows that the piecewise cubic polynomials $v^{(i,k)}$ and $w^{(i,k)}$ are segments of cubic polynomials v and w defined over the whole princial edge. These cubic polynomials v and w, in turn, are determined by function value, first and mixed derivatives of g at the parent nodes. If a parent node is a primary node, then v and w and their first derivatives are zero due to (31). If a parent node is a hanging node, we can repeat the argument recursively until we reach primary nodes, because $\mathcal{K}$ is a hierarchical subdivision. Hence, with (31), we obtain

$$\partial_\alpha g(\zeta^{(i)}) = 0, \quad \forall\, (i,\alpha) \in \mathcal{I}_X \times I_\alpha.$$

Therefore, $g \equiv 0$ and f can be represented as a linear combination of functions in Φ. Since $f \in X_h$ was arbitrary, Φ is a basis of X_h. □

Remark 4.10. A direct consequence of Theorem 4.9 is that the dimension of X_h is $4 \cdot N_P$, i.e. the number of degrees of freedom of the discrete solution only depends on the number of primary nodes. Note that the definition of the basis functions given in Section 3 is a special case of the definition given here, because $\mathcal{I}_P = \mathcal{I}_X$ and $N_P = N_X$ in a subdivision without hanging nodes.

4.4 Local Representation of Basis Functions

As usual in FEM, the stiffness matrix K_h and the right-hand side $\underline{f}_h$ in equation (8) are assembled cell-wise. For each cell, its contributions to the integrals $a(\widehat{\varphi}_\beta^{(j)}, \widehat{\varphi}_\alpha^{(i)})$ and $\langle F, \widehat{\varphi}_\alpha^{(i)} \rangle$ are calculated. To do this, we need to know which basis functions have support in a cell K, and we need to find their local representation on K.

We define the set of *ancestors of a node* $\zeta^{(j)}$:

$$\mathsf{Anc}(\zeta^{(j)}) := \begin{cases} \{\zeta^{(j)}\}, & \text{if } j \in \mathcal{I}_P \\ \mathsf{Anc}(\zeta^{(p)}) \cup \mathsf{Anc}(\zeta^{(q)}), & \text{else, i.e. if } j \in \mathcal{I}_H, \end{cases}$$

where $\zeta^{(p)}$ and $\zeta^{(q)}$ are the parent nodes of $\zeta^{(j)}$ in the case $j \in \mathcal{I}_H$.

We also define the set of *ancestors of a cell* K:

$$\mathsf{Anc}(K) := \bigcup_{i \in \mathcal{I}_K} \mathsf{Anc}(\zeta^{(i)}).$$

For example, consider the subdivision in Fig. 8:

$$\begin{aligned} \mathsf{Anc}(\zeta^{(7)}) &= \mathsf{Anc}(\zeta^{(6)}) \cup \mathsf{Anc}(\zeta^{(8)}) \\ &= \{\zeta^{(6)}\} \cup \mathsf{Anc}(\zeta^{(3)}) \cup \mathsf{Anc}(\zeta^{(14)}) \;=\; \{\zeta^{(3)}, \zeta^{(6)}, \zeta^{(14)}\}. \end{aligned}$$

Let K refer to the cell in Fig. 8 that has $\zeta^{(6)}$, $\zeta^{(7)}$, $\zeta^{(9)}$, and $\zeta^{(10)}$ as its vertices. Then

$$\mathsf{Anc}(K) = \{\zeta^{(3)}, \zeta^{(6)}, \zeta^{(11)}, \zeta^{(14)}\}.$$

Let $(i, \alpha) \in \mathcal{I}_P \times I_\alpha$ and $K \in \mathcal{K}$ be arbitrary but fixed. Naturally, the restriction of the basis function $\widehat{\varphi}_\alpha^{(i)} \in X_h$ to $\overline{K}$ is bi-cubic, therefore it can be represented locally as a linear combination of local basis functions $\widehat{\psi}_\beta^{(j)} \in \Psi_K$:

$$\widehat{\varphi}_\alpha^{(i)}|_{\overline{K}} \;=\; \sum_{j \in \mathcal{I}_K} \sum_{\beta \in I_\alpha} \lambda_\beta^{(j)} \widehat{\psi}_\beta^{(j)}. \tag{32}$$

The coefficients $\lambda_\beta^{(k)}$ are the values defined by (29) at primary nodes and determined at hanging nodes through the procedure given above in Section 4.2.3. Note that the coefficients $\lambda_\beta^{(j)}$ depend on i and α, so actually $\lambda_\beta^{(j)} = \lambda_\beta^{(j)}(i,\alpha)$. We determine these coefficients $\lambda_\beta^{(j)}$ by the following method that can be interpreted as a reversal of Procedure 4.2:

Procedure 4.11. (Determination of coefficients for local representation)

- If $\zeta^{(j)}$ is a primary node:
  - $\lambda_\beta^{(j)} := \delta_{(i,\alpha),(j,\beta)}$.
- If $\zeta^{(j)}$ is a hanging node:
  - Find the parent nodes $\zeta^{(p)}$ and $\zeta^{(q)}$ of $\zeta^{(j)}$.
  - Determine $\lambda_\beta^{(p)} = \lambda_\beta^{(p)}(i,\alpha)$ and $\lambda_\beta^{(q)} = \lambda_\beta^{(q)}(i,\alpha)$, $\beta \in I_\alpha$ by applying this Procedure 4.11 recursively to $\zeta^{(p)}$ and $\zeta^{(q)}$.
  - Apply Procedure 4.2 to derive the four values $\lambda_\beta^{(j)}$, $\beta \in I_\alpha$ from the values $\lambda_\beta^{(p)}$ and $\lambda_\beta^{(q)}$.

The recursion terminates, because the procedure is applied to nodes with decreasing levels until level 1, i.e. until a primary node is reached. The set of primary nodes that are reached by this method is $\mathsf{Anc}(\zeta^{(j)})$. The consequence is formulated in the following lemma:

Lemma 4.12. *Let $K \in \mathcal{K}$ and let $\widehat{\varphi}_\alpha^{(i)}$ be an arbitrary, but fixed basis function of X_h. Then*

$$\zeta^{(i)} \notin \mathsf{Anc}(K) \quad \Rightarrow \quad \operatorname{supp}(\widehat{\varphi}_\alpha^{(i)}) \cap \overline{K} \;=\; \emptyset.$$

Proof. Follows from the definition of $\widehat{\varphi}_\alpha^{(i)}$: If $\zeta^{(i)} \notin \mathsf{Anc}(K)$, we obtain $\lambda_\beta^{(j)} = 0$, for all $(j,\beta) \in \mathcal{I}_K \times I_\alpha$, and therefore $\widehat{\varphi}_\alpha^{(i)}|_{\overline{K}} \equiv 0$. □

4.5 Edge Insertion into a Hierarchical Subdivision

We define the term *edge insertion*, as dividing one cell $K \in \mathcal{K}$ into two cells by inserting an axis-aligned edge as illustrated in Fig. 14.

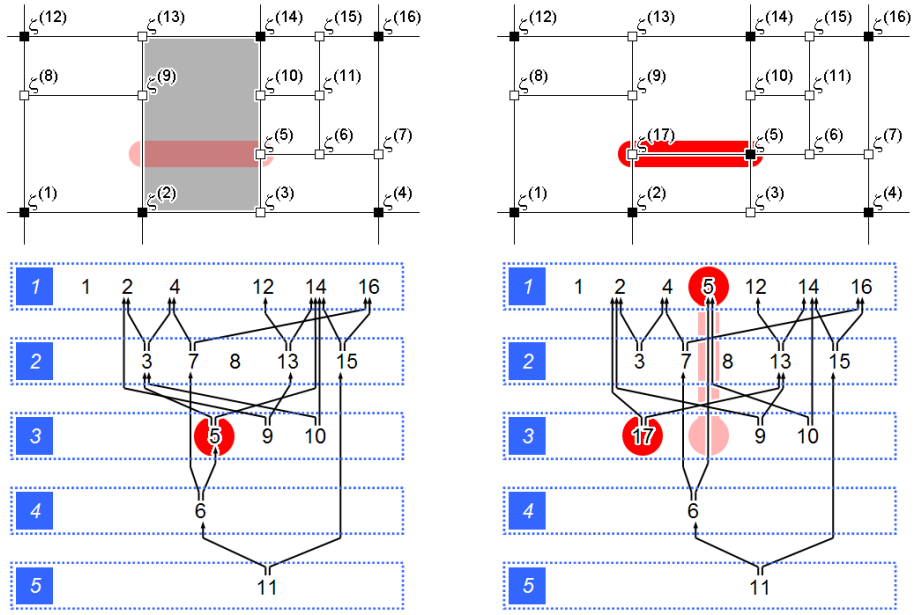
Since we have to insert edges in order to refine all or some selected cells, we need to discuss the effects of such an edge insertion on the structure of the subdivision $\mathcal{K}$ and on the function space X_h.

Lemma 4.13. *Let $\mathcal{K}$ be a hierarchical subdivision and let $\mathcal{K}^\ddagger$ be the subdivision resulting from $\mathcal{K}$ and an edge insertion. Then $\mathcal{K}^\ddagger$ is a hierarchical subdivison.*

Proof. When a new edge is inserted into a hierarchical subdivision, two cases can appear at its vertices:

Case 1: A new "T-shaped junction" is created, thus creating a new hanging node (such as node $\zeta^{(17)}$ in Fig. 14).

Case 2: An already existing "T-shaped junction" is changed into an "X-shaped junction", i.e. an already existing hanging node is changed into a primary node (such as node $\zeta^{(5)}$ in Fig. 14).



(a) Before inserting a horizontal edge into the shaded cell.

(b) After edge insertion: $\zeta^{(17)}$ is a new hanging node. Former hanging node $\zeta^{(5)}$ has become a primary node.

Figure 14: Example of a hierarchical subdivision before and after an edge insertion with corresponding node-levels-graphs.

Case 1: Creation of a New Hanging Node $\zeta^{(p)}$

The node is created on a principal edge and has two parent nodes. On the new, inserted edge, there are no nodes which have $\zeta^{(p)}$ as a parent node. The new node $\zeta^{(p)}$ only needs to get assigned a level that is higher than the levels of its parent nodes to make sure that the subdivision remains hierarchical.

Case 2: Changing a Hanging Node $\zeta^{(q)}$ into a Primary Node

The new primary node splits a principal edge E containing $\zeta^{(q)}$ into two principal edges E_1 and E_2, both of which have $\zeta^{(q)}$ as a parent node (in Fig. 14, principal edge $E^{(3,14)}$ is split into the principal edges $E^{(3,5)}$ and $E^{(5,14)}$).

Let $\zeta^{(i)}$ be a hanging node that was on E (such as $\zeta^{(10)}$ in Fig. 14). Depending on its position on E, one of the parent nodes of $\zeta^{(i)}$ is replaced by $\zeta^{(q)}$. Since $\mathcal{K}$ was a hierarchical subdivision before the edge insertion, $\zeta^{(i)}$ has a higher level than the other parent node. Because the level of a primary node is always set to 1, $\zeta^{(i)}$ has a higher level than $\zeta^{(q)}$ as well.

A node $\zeta^{(j)}$ that had $\zeta^{(q)}$ as a parent node before the edge insertion (such as $\zeta^{(6)}$ in Fig. 14), still has $\zeta^{(q)}$ as parent node. Before the insertion, the level of $\zeta^{(j)}$ was higher than the level of $\zeta^{(q)}$. This is not changed by setting the level of $\zeta^{(q)}$ to 1. □

Note that in Fig. 14(b), the level of node $\zeta^{(6)}$ is 4, while the levels of its parent nodes are 1 and 2. One could shift node $\zeta^{(6)}$ to level 3 and node $\zeta^{(11)}$ to level 4. However, this is not necessary for showing that the new subdivision is hierarchical, because (L2) is fulfilled either way.

Theorem 4.14. *Let $\mathcal{K}$ be a hierarchial subdivision and let $\mathcal{K}^\ddagger$ be the subdivision resulting from $\mathcal{K}$ and an edge insertion. Let $X_h := \mathfrak{Q}_3^1(\mathcal{K})$ and $X_h^\ddagger := \mathfrak{Q}_3^1(\mathcal{K}^\ddagger)$ be the finite element spaces defined over those subdivisions, then the inclusion*

$$X_h \quad \subseteq \quad X_h^\ddagger \tag{33}$$

holds.

Proof. Let K be the cell of the subdivision $\mathcal{K}$ that is split into the two cells K' and K'' of $\mathcal{K}^\ddagger$. Obviously, a bi-cubic function over K can be represented exactly by two bi-cubic functions defined over K' and K'', thus $\mathfrak{Q}_3^1(\mathcal{K}) \subseteq \mathfrak{Q}_3^1(\mathcal{K}^\ddagger)$. □

Because of Lemma 4.13, we know that $\mathcal{K}^\ddagger$ is a hierarchical subdivision, therefore we can define basis functions of $X_h^\ddagger$ analogous to (29) and (30).

4.6 Implemented Cell Refinement Procedure

As discussed in Section 4.3, we do not gain any degrees of freedom by inserting hanging nodes. Therefore, at least one new primary node should be created when refining a cell. In our case, we define a *cell refinement* as inserting one new primary node inside the cell, thereby splitting the cell into four smaller axis-aligned cells as illustrated in Fig. 15. This way, at least four new degrees of freedom are gained. Such a cell refinement can be interpreted as repeating the discussed edge insertions three times, therefore the statements of Lemma 4.13 and Theorem 4.14 hold.

The more general considerations in Sections 4.3 and 4.5 were made so that other refinement procedures are also covered. Fig. 16 shows examples of alterative procedures that can be realized by consecutive edge insertions.

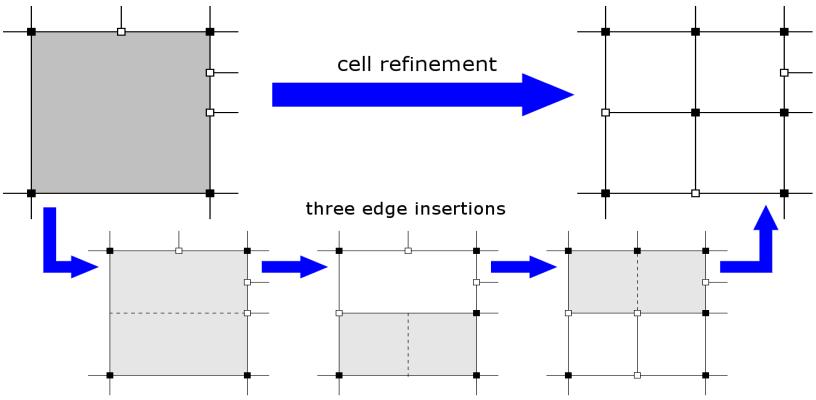


Figure 15: Cell refinement through three consecutive edge insertions.

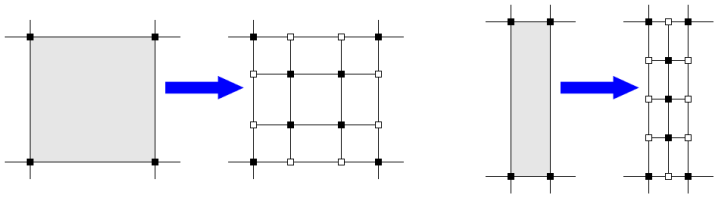


Figure 16: Examples of possible alternative cell refinement procedures.

4.6.1 Cell Refinement on a Structured Grid

In the actual implementation, a very specific way of refining cells was chosen. If two neighbouring cells are both refined, we want a primary node to be created at their common edge. Even stronger, we make the following requirement: Let K and K' be two cells that share a whole border edge E as illustrated in Fig. 17, i.e. E is an elementary edge. Assume that cell K is refined, creating a new hanging node $\zeta^{(p)}$ on E. The smaller cells that K is split into may or may not be refined in the further process. Either way, as soon as cell K' is refined, it must be done so that $\zeta^{(p)}$ is changed into a primary node.

Now, assume that the cell K'' to the east of cell K' in Fig. 17 is also refined before cell K', creating a hanging node $\zeta^{(q)}$ on their common edge E'. When refining cell K, the stated requirement has to be fulfilled at $\zeta^{(p)}$ and $\zeta^{(q)}$, i.e. by inserting one horizontal edge, both nodes have to be changed into primary nodes. This implies that the nodes which are inserted in the course

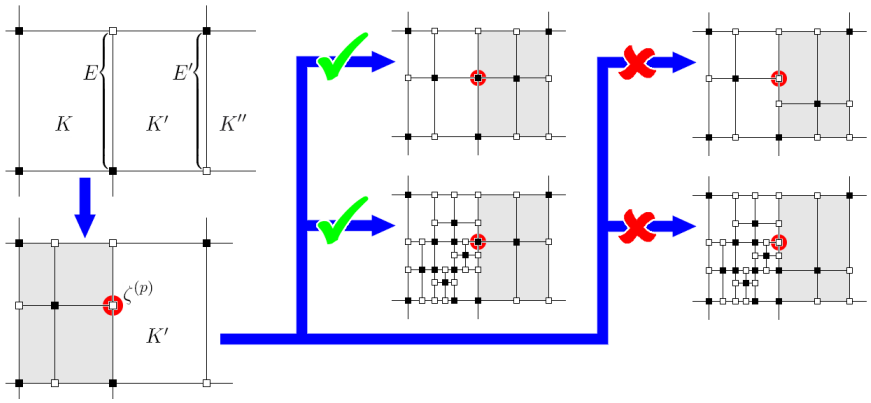


Figure 17: Illustration of the refinement of two neighbouring cells K and K'. The node $\zeta^{(p)}$ marked by the red circle must be the first to be changed into a primary node, when cell K' is refined.

of cell refinement have to be aligned on a grid. Additionally, we require that $\zeta^{(p)}$ and $\zeta^{(q)}$ are changed into primary nodes when refining cell K', regardless of how many other hanging nodes have been created on E and E'. This implies that the grid needs to be structured somehow.

We define such a structured grid via two coordinate vectors T_1 and T_2. In order not to confuse them with nodes of the subdivision or knots of the geometry mapping (see Section 5), the entries $\xi_k^{(i)}$, $k \in \{1, 2\}$ of these vectors T_k are referred to as *ticks*. As illustrated in Fig. 18(a), one can think of T_1 and T_2 as ticks on the two coordinate axes, which motivates the use of this term.

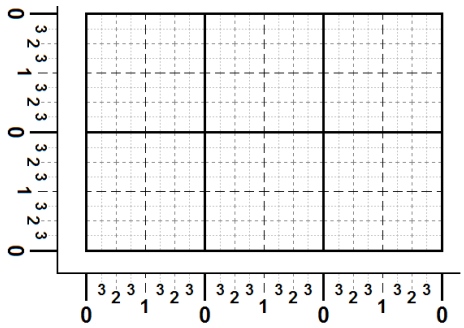
To each tick $\xi_k^{(i)}$, we assign a *ticks-layer* that we denote with $\mathsf{TicksL}(\xi_k^{(i)})$ (we choose the term "layer" instead of "level" in order to avoid confusion with the levels of nodes introduced in Section 4.1).

We construct T_1 and T_2 iteratively as follows:

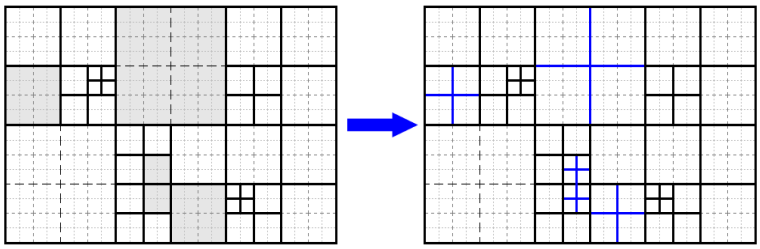
Procedure 4.15. (Construction of ticks for T_k, $k \in \{1, 2\}$)

- Initialization:
  Define
  $$T_k^0 := (\xi_k^{(1)}, \xi_k^{(2)}, \ldots, \xi_k^{(n_k)})$$
  as the sorted sequence of ξ_k-coordinates of nodes in the initial subdivision, i.e. $\xi_k^{(1)} = 0$, $\xi_k^{(n_k)} = 1$, and $\xi_k^{(i)} < \xi_k^{(i+1)}, \forall\, i \in \{1, \ldots, n_k - 1\}$. Assign ticks-layer 0 to each tick in T_k^0.

- Ticks-insertion in the coordinate vector T_k^L, $L \geq 0$:
  For each $i \in \{1, \ldots, |T_k^L| - 1\}$, define a new tick $\bar{\xi}_k^{(i)}$ in the interior of



(a) Example of a grid and ticks-layers with highest ticks-layer $L_{\max} = 3$.



(b) Refinement of the cells shaded gray, based on the grid in (a).

Figure 18: Illustration of a structured grid with ticks in both coordinate-directions (a) and cell refinements based on that grid (b).

the interval $(\xi_k^{(i)}, \xi_k^{(i+1)})$ and assign the following ticks-layer:

$$\mathsf{TicksL}(\bar{\xi}_k^{(i)}) \;:=\; \max\left\{\mathsf{TicksL}(\xi_k^{(i)}), \mathsf{TicksL}(\xi_k^{(i+1)})\right\} + 1$$

- Iteration:
  Define
  $$T_k^{L+1} \;:=\; (\xi_k^{(1)}, \bar{\xi}_k^{(1)}, \xi_k^{(2)}, \bar{\xi}_k^{(2)} \ldots, \bar{\xi}_k^{(n_k-1)}, \xi_k^{(n_k)})$$
  and repeat step ticks-insertion with T_k^{L+1} iteratively until the desired ticks-layer $L_{\max}$ is reached. Then, set $T_k := T_k^{L_{\max}}$.

The resulting two coordinate vectors T_1 and T_2 define a grid such as illustrated in Fig. 18(a). Note that the new tick $\bar{\xi}_k^{(i)}$ in step ticks-insertion is not necessarily the midpoint $(\xi_k^{(i)} + \xi_k^{(i+1)})/2$ of the interval. In Section 5.2.2, we will discuss an alternative procedure for determining $\bar{\xi}_k^{(i)}$.

Once we have defined T_1 and T_2, we refine a cell K as follows:

Procedure 4.16. (Cell-refinement on a structured grid)

- Find the ticks $\xi_1^{(w)}$, $\xi_1^{(e)}$, $\xi_2^{(s)}$, and $\xi_2^{(n)}$, such that $K = [\xi_1^{(w)}, \xi_1^{(e)}] \times [\xi_2^{(s)}, \xi_2^{(n)}]$ (i.e. the ticks define the "western", "eastern", "southern", and "northern" boundaries of K).
- Let L_w, L_e, L_s, and L_n be the ticks-layers of $\xi_1^{(w)}$, $\xi_1^{(e)}$, $\xi_2^{(s)}$, and $\xi_2^{(n)}$, respectively.
- Find the unique tick $\xi_1^{(m)}$ between $\xi_1^{(w)}$ and $\xi_1^{(e)}$ and the unique tick $\xi_2^{(m)}$ between $\xi_2^{(s)}$ and $\xi_2^{(n)}$, such that

$$\begin{aligned} \mathsf{TicksL}(\xi_1^{(m)}) &= \max\{L_w, L_e\} + 1. \\ \mathsf{TicksL}(\xi_2^{(m)}) &= \max\{L_s, L_n\} + 1. \end{aligned}$$

- Insert a new node $\zeta^{(N_X+1)}$ at the coordinates $(\xi_1^{(m)}, \xi_2^{(m)})$ and split the cell K into four cells by inserting edges at the coordinates of $\zeta^{(N_X+1)}$.

Illustrations for this procedure are given in Fig. 18(b) and Fig. 19.

4.6.2 Advantages/Disadvantages of using a Structured Grid

Procedure 4.16 described above allows a new definition of parent nodes: A new primary node is inserted at the coordinates $(\xi_1^{(m)}, \xi_2^{(m)})$, which triggers the insertion of up to four new hanging nodes. Each one of these hanging nodes is inserted on a border edge of K, and we can define the vertices of that border edge to be the parent nodes of the new hanging node. For example, assume that a new hanging node is inserted in the "southern" border edge of K, i.e. at the coordinates $(\xi_1^{(m)}, \xi_2^{(s)})$. Then, the parent nodes of this node would be the "southwestern" corner of K (at the coordinates $(\xi_1^{(w)}, \xi_2^{(s)})$) and the "southeastern" corner of K (at the coordinates $(\xi_1^{(e)}, \xi_2^{(s)})$).

With this definition, it is possible to assign levels to each node such that (L1) and (L2) are fulfilled. In Fig. 19, some examples for node-levels-graphs in this new setting are shown. The difference to the previous version is that now two nodes can only have one "child node" in common. For example, see Fig. 19(a): With the previous definition, the nodes $\zeta^{(2)}$, $\zeta^{(3)}$, $\zeta^{(4)}$, and all hanging nodes east of node $\zeta^{(4)}$ would have the same parent nodes $\zeta^{(1)}$ and $\zeta^{(5)}$ (the latter is outside the plotted region of the subdivision). In the new setting, $\zeta^{(4)}$ is the only node that has $\zeta^{(1)}$ and $\zeta^{(5)}$ as parent nodes. The parent nodes of $\zeta^{(3)}$ are $\zeta^{(1)}$ and $\zeta^{(4)}$.

A big advantage of this new definition combined with Procedure 4.16 is that the parent nodes do not change in the course of refinement, which is also illustrated in Fig. 19: The node $\zeta^{(4)}$ changes from a hanging node in Fig. 19(a) to a primary node in Fig. 19(b). With the previous definition of parent nodes, we would have to re-assign one parent node to all other hanging nodes on the edge $E^{(1,5)}$. With the new definition, however, this is not the case, as illustrated by the node-levels-graph of Fig. 19(b).

Since (L1) and (L2) are still fulfilled, all statements that were made for hierarchical subdivisions still hold.

With this procedure, we limit ourselves to inserting one node per cell at a time and neglect other refinement strategies such as those in Fig. 16, which might be perceived as a disadvantage.

Another disadvantage is that the depth of the recursion in Procedure 4.11 for determining the coefficients $\lambda_\alpha^{(i)}$ at a hanging node is larger. For example, in Fig. 19(a), instead of directly calculating $\lambda_\alpha^{(2)}$ from the values at $\zeta^{(1)}$ and $\zeta^{(5)}$, we have to recursively carry out similar calculations for $\zeta^{(3)}$ and $\zeta^{(4)}$.

This disadvantage, however, can be countered in the following setting: Assume that, in step ticks-insertion of Procedure 4.15, the new ticks $\bar{\xi}_k^{(i)}$ are inserted at the midpoints $(\xi_k^{(i)} + \xi_k^{(i+1)})/2$. In this case, with the new definition of parent nodes, every hanging node is exactly the midpoint of its two parent nodes.

Let v be a cubic polynomial over the interval $[p, q]$ with size $h := |q - p|$. Then v and v' at the midpoint of the interval are given by

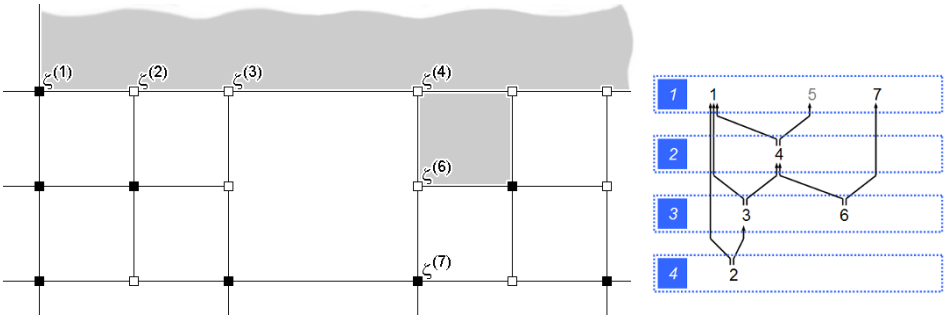
$$v\left(\frac{p+q}{2}\right) = \frac{1}{2}\Big(v(p) + v(q)\Big) + \frac{h}{8}\Big(v'(p) - v'(q)\Big), \tag{34}$$

$$v'\left(\frac{p+q}{2}\right) = -\frac{3}{2h}\Big(v(p) - v(q)\Big) - \frac{1}{4}\Big(v'(p) + v'(q)\Big). \tag{35}$$

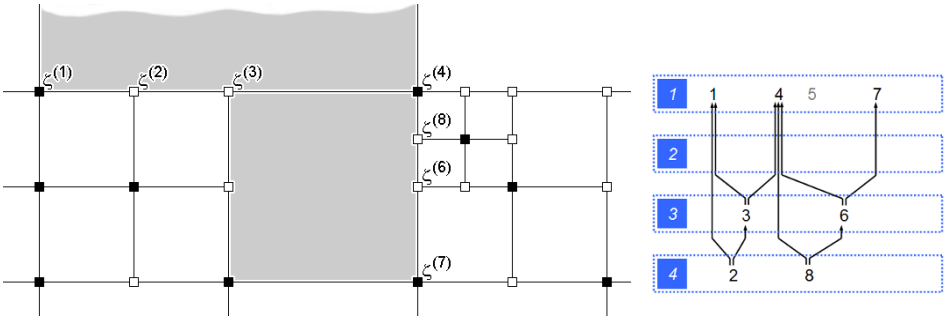
In Procedure 4.2, the values of v and v' at p and q correspond to the coefficients $\lambda_\alpha^{(i)}$, $i \in \{p, q\}$. Thus, Step 2 of Procedure 4.2 is reduced to evaluating (34) and (35) from the known coefficients $\lambda_\alpha^{(i)}$ at the parent nodes and the easily calculated distance h between the parent nodes, without actually having to construct and evaluate v.

4.7 Incorporation of Boundary Conditions

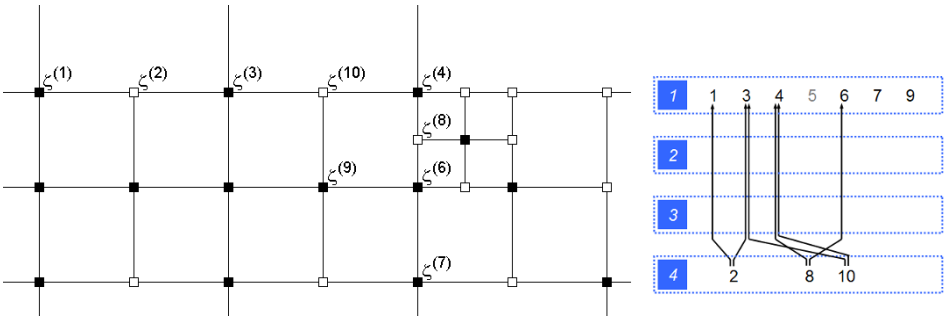
Boundary nodes are always primary nodes, therefore the incorporation of boundary conditions is done exactly the same way as in a subdivision without hanging nodes and as described in Section 3.5.



(a) Cells marked for refinement shaded gray. Any two parent nodes have only one "child node" in common.



(b) Parent nodes of hanging nodes remain unchanged after a refinement step. Cells marked for the next refinement step shaded gray.



(c) Parent nodes remain unchanged. Shifting hanging nodes $\zeta^{(2)}$, $\zeta^{(8)}$, and $\zeta^{(10)}$ to level 2 is possible, but not necessary.

Figure 19: Subdivisions and corresponding node-levels-graphs illustrating the new definition of parent nodes. (L1) and (L2) are fulfilled. Node $\zeta^{(5)}$ is outside the plotted area.

5 NURBS Geometry Mapping

5.1 Introduction to B-Splines and NURBS

At first, we give an overview of the construction of B-splines and Non-Uniform Rational B-Spline (NURBS) basis functions [8, 10, 11].

5.1.1 B-Spline Basis Functions

Definition 5.1. Let $p \in \mathbb{N}_0$ and let $\eta = (\eta_1, \dots, \eta_m)$ be a sequence of knots in the parameter space with $\eta_i \leq \eta_{i+1}$ for all i. The $n = m-p-1$ univariate *B-spline basis functions* $B_{i,p} : [\eta_1, \eta_m] \to \mathbb{R}$ are defined recursively as follows:

$$B_{i,0}(\xi) := \begin{cases} 1 & \text{for} \quad \eta_i \leq \xi < \eta_{i+1} \\ 0 & \text{else} \end{cases} \tag{36}$$

$$B_{i,p}(\xi) := \frac{\xi - \eta_i}{\eta_{i+p} - \eta_i} B_{i,p-1}(\xi) + \frac{\eta_{i+p+1} - \xi}{\eta_{i+p+1} - \eta_{i+1}} B_{i+1,p-1}(\xi). \tag{37}$$

p is called the *degree* of the B-spline basis functions and η the *knot vector*. A knot vector is called *uniform*, if the knots are equally-spaced in the parameter space, and *non-uniform* otherwise.

Whenever a zero denominator appears in (37), the corresponding function $B_{i,p}$ is zero, so the whole fraction can be considered zero. Examples for B-splines of degrees 0 to 3 over a uniform knot vector are depicted in Fig. 20; quadratic B-splines over a non-uniform knot vector are shown in Fig. 21(a).

The following properties of B-spline basis functions are illustrated in Fig. 20 and Fig. 21(a): The support of the basis function $B_{i,p}$ is contained in the interval $[\eta_i, \eta_{i+p+1}]$. In general, a B-spline is continuous, if its degree is greater or equal to 1, and B-splines of degree p have $p-1$ continuous derivatives. If an interior knot is repeated k times, the number of continuous derivatives decreases by k. If an interior knot appears exactly p times, the function is interpolatory (compare $\eta_6 = \eta_7 = 3$ in Fig. 21(a)).

On the interval $[\eta_{p+1}, \eta_{n-p}]$, the B-spline basis functions form a partition of 1, i.e.:

$$\forall\, \xi \in [\eta_{p+1}, \eta_{n-p}] : \quad \sum_{i=1}^{n} B_{i,p}(\xi) = 1. \tag{38}$$

If the first and the last knot appear $p+1$-times, the functions are interpolatory at the beginning and the end of the interval, and (38) holds for all $\xi \in [\eta_1, \eta_n]$ (see Fig. 21(a)).

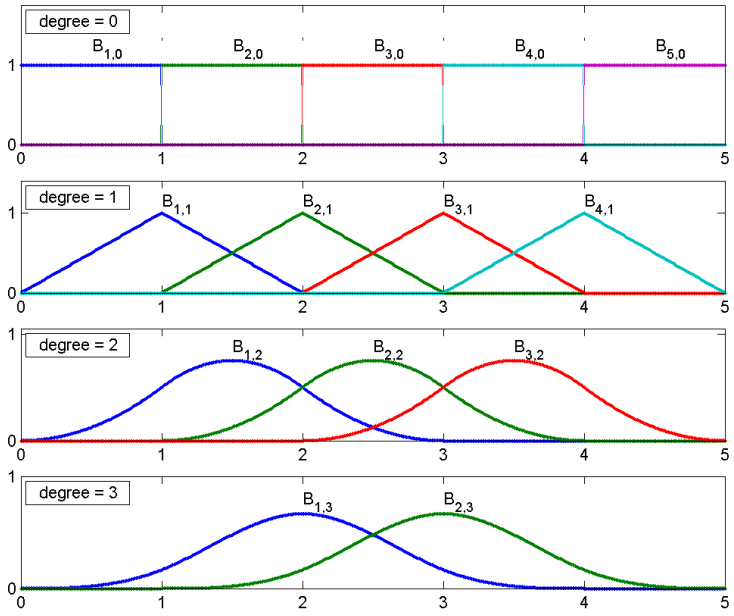


Figure 20: B-spline basis functions of degrees 0,1,2, and 3 with the uniform knot vector $\eta = (0, 1, 2, 3, 4, 5)$. No repeated knots, the functions are in C^{p-1}.

A knot vector in which the first and the last knot appear $p+1$ times is called an *open knot vector*. This is the standard in CAD literature and from now on, we will only consider open knot vectors.

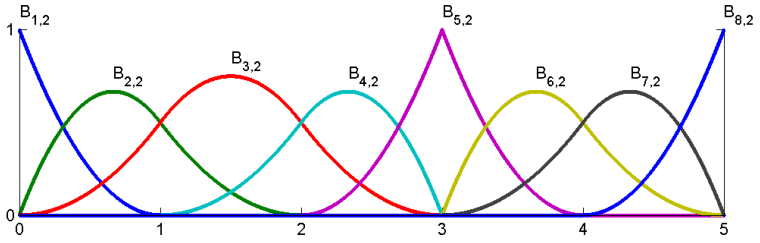
5.1.2 B-Spline Curve and NURBS Curve

A *B-spline curve* in $\mathbb{R}^d$ is constructed as a linear combination of B-spline basis functions with coefficients $P_i \in \mathbb{R}^d$. These points P_i are referred to as *control points*; the piecewise linear interpolation of the control points is called *control polygon*. For n basis functions $B_{i,p}$, $i \in \{1, \ldots, n\}$ and n control points $P_i \in \mathbb{R}^d$, the piecewise polynomial *B-spline curve* is given by:

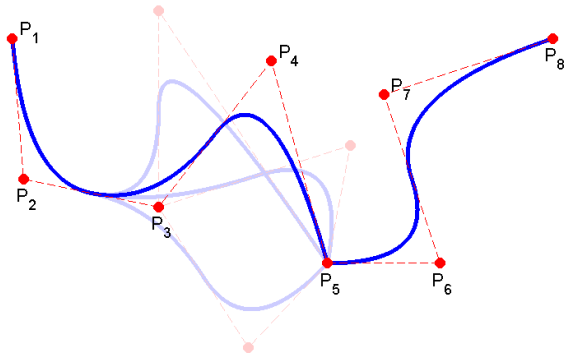
$$C(\xi) \quad := \quad \sum_{i=1}^{n} B_{i,p}(\xi) \cdot P_i. \tag{39}$$

We will also refer to p as the degree of the B-spline curve C.

As an example, Fig. 21(b) depicts the B-spline curve that results from the basis functions in Fig. 21(a) and the illustrated, arbitrarily chosen control polygon. Each control point P_i of the B-spline curve is associated with one basis function $B_{i,p}$. The curve in Fig. 21(b) is interpolatory at P_1, P_5, and P_8, which corresponds to $B_{1,2}(0) = 1$, $B_{5,2}(3) = 1$, and $B_{8,2}(5) = 1$, respectively.



(a) $p = 2$, i.e. generally C^1-continuity. Non-uniform open knot vector $\eta = (0, 0, 0, 1, 2, 3, 3, 4, 5, 5, 5)$. Knot at $\xi = 3$ appears p-times: Interpolatory function and only C^0-continuity. Interpolatory functions at the beginning and the end of the interval.



(b) B-spline curve resulting from basis functions in Fig. 21(a) and the shown set of control points. Curve is interpolatory at the control points P_1, P_5, and P_8. Effects of changing the position of P_4 indicated by semi-transparent curves.

Figure 21: Example of quadratic B-spline basis functions and an associated B-spline curve defined by a random set of control points.

As mentioned, repeating an interior knot reduces the differentiability of the B-spline basis functions. Hence, if an interior knot η_j appears twice, a B-spline curve with degree $p = 2$ is not continuously differentiable at η_j.

Another possibility of constructing a B-spline curve that is interpolatory at an "interior" control point is the following: Let C be a B-spline curve of degree $p = 2$, defined over an open knot vector η with no repeated interior knots. For $2 \leq j \leq n-1$, let η_{j+2} be an interior knot of η and let $P_j = P_{j+1}$ be a double control point. Then, due to the properties of the B-spline basis functions, direct calculation yields

$$\begin{aligned} C(\eta_{j+2}) &= \sum_{i=1}^{n} B_{i,2}(\eta_{j+2}) \cdot P_i \\ &= B_{j,2}(\eta_{j+2}) \cdot P_j + B_{j+1,2}(\eta_{j+2}) \cdot P_{j+1} \;=\; P_j, \end{aligned}$$

hence, C is interpolatory at P_j. Furthermore, one can calculate directly that

$$C'(\eta_{j+2}) = 0,$$

i.e. the tangent vector of C vanishes at P_j. Note that, even though a cusp, i.e. a "visual discontinuity" of the tangent vector may appear at P_j, the curve is still continuously differentiable at P_j.

If $d = 2$ and a control point appears three times, i.e. if we have $P_j = P_{j+1} = P_{j+2}$, then C' vanishes over the whole interval $[\eta_{j+2}, \eta_{j+3}]$ and $C(\eta) = P_j$ for all $\eta \in [\eta_{j+2}, \eta_{j+3}]$. In this case, the curve C is not injective.

Since each basis function has only small support, the position of a single control point effects the shape of the curve only over a small part. Hence, B-spline curves can be easily edited locally by moving a control point. This is also illustrated in Fig. 21(b) by the semi-transparently indicated positions of control point P_4 and the resulting changed curves.

Some important geometric entities such as circles can not be exactly represented using only B-splines. For this, it is necessary to introduce *Rational B-Splines*. We associate a *weight* w_i with each control point P_i and define a *NURBS curve* as follows:

$$C_R(\xi) := \sum_{i=1}^{n} R_i^p(\xi) \cdot P_i,$$

with the NURBS basis functions

$$R_i^p(\xi) := \frac{B_{i,p}(\xi) \cdot w_i}{\sum_{j=1}^{n} B_{j,p}(\xi) \cdot w_j}.$$

If all weights w_i are equal, we have $R_i^p = B_{i,p}$ due to (38).

In Fig. 22, the unit circle is represented exactly by a NURBS curve. For comparison, the B-spline curve using the same control points is plotted as well. The geometry data is given in Appendix B.1.

For the sake of completeness, we mention that NURBS in $\mathbb{R}^d$ can be interpreted as images of B-splines in $\mathbb{R}^{d+1}$ under projective transformation. This aspect, however, has no influence on the topic of this theses; see [8] for details.

5.1.3 B-Spline- and NURBS-Surface, Geometry Mapping G

Definition 5.2. Given a *control net* of points $P_{i,j} \in \mathbb{R}^2$, $(i,j) \in \{1,\dots,n\} \times \{1,\dots,n^*\}$ and corresponding weights $w_{i,j} \in \mathbb{R}$, the two-dimensional *NURBS*

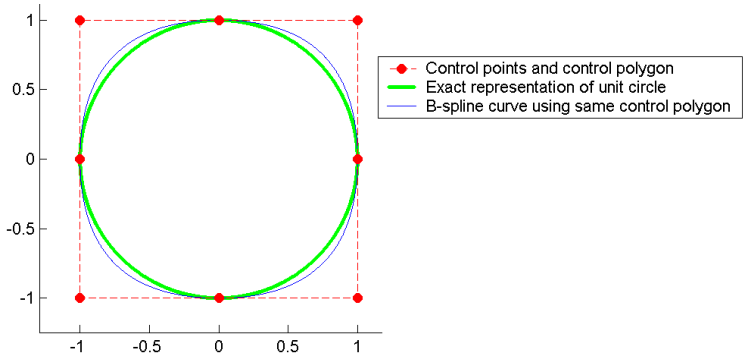


Figure 22: Exactly represented unit circle as example for a quadratic NURBS-curve with control polygon (plotted in red) and non-uniform knot vector (for complete geometry data see Appendix B.1).

surface is defined by

$$G(\xi_1, \xi_2) \quad := \quad \sum_{i=1}^{n} \sum_{j=1}^{n^*} \mathbf{R}_{(i,j)}^{(p,p^*)}(\xi_1, \xi_2) \cdot P_{i,j} \tag{40}$$

with the bivariate NURBS basis functions

$$\mathbf{R}_{(i,j)}^{(p,p^*)}(\xi_1, \xi_2) \quad := \quad \frac{B_{i,p}(\xi_1) \cdot B^*_{j,p^*}(\xi_2) \cdot w_{i,j}}{\sum_{k=1}^{n} \sum_{l=1}^{n^*} B_{k,p}(\xi_1) \cdot B^*_{l,p^*}(\xi_2) \cdot w_{k,l}} \tag{41}$$

where $B_{i,p}$ and B^*_{j,p^*} are two families of B-spline basis functions defined by two degrees p and p^* and two knot vectors $\eta = (\eta_1, \ldots, \eta_{n+p+1})$ and $\eta^* = (\eta^*_1, \ldots, \eta^*_{n^*+p^*+1})$.

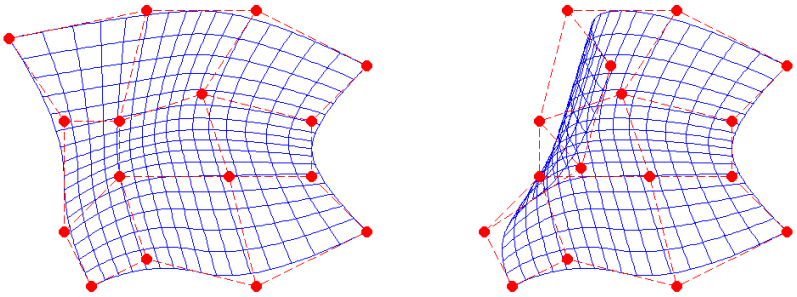
An example for a NURBS surface is depicted in Fig. 23(a). In order to exclude mappings that penetrate themselves, we require G to be bijective. An example for a NURBS surface that is not injective is illustrated in Fig. 23(b).

Some properties of B-spline basis functions are inherited by NURBS basis functions. The support of the NURBS basis function $\mathbf{R}_{(i,j)}^{(p,p^*)}$ is contained in $[\eta_i, \eta_{i+p+1}] \times [\eta^*_j, \eta^*_{j+p^*+1}]$. Accordingly, changing the position of one control point effects the shape of the NURBS surface only locally. The continuity of the NURBS basis function is the same as in case of B-splines, i.e. if an interior knot is repeated k times, the number of continuous derivatives decreases by k.

If all weights $w_{i,j}$ are equal, we again have

$$\mathbf{R}_{(i,j)}^{(p,p^*)}(\xi_1, \xi_2) \;=\; B_{i,p}(\xi_1) \cdot B^*_{j,p^*}(\xi_2)$$

and (40) defines a *B-spline surface*.



(a) Example for NURBS surface.

(b) Example for self-penetrating NURBS surface.

Figure 23: Examples for NURBS surfaces depicted with control polygons. Self-penetrating mappings such as in Fig. (b) are not considered in this thesis.

Definition 5.3. From now on, whenever we discuss the global mapping

$$\begin{aligned} G: \quad Q &\rightarrow \Omega \subset \mathbb{R}^2 \\ \xi &\mapsto G(\xi), \end{aligned}$$

we will be referring to a continuous, bijective NURBS surface as defined in (40), unless specified otherwise. Its Jacobian will be denoted by

$$J := \nabla_\xi G = \begin{pmatrix} \partial G_1/\partial \xi_1 & \partial G_1/\partial \xi_2 \\ \partial G_2/\partial \xi_1 & \partial G_2/\partial \xi_2 \end{pmatrix}.$$

Edges in the parameter domain are mapped by G to curved lines in the physical domain. To simplify descriptions, we will also refer to these lines as "edges" in some places.

5.2 Different Representations of the same Physical Domain

The representation of a physical domain Ω as a NURBS surface with parameter domain Q is not unique. An example illustrates the different properties of two different geometry mappings representing the same physical domain:

5.2.1 Illustrative Example

Fig. 24 shows a plate with a circular hole around the origin. Fig. 24(a) and Fig. 24(b) illustrate how segments of the boundary of the parameter domain are mapped to segments of the boundary of the physical domain. The "southern" boundary $(\xi_1, 0) \in Q$ is mapped to the circular curve which

can be represented exactly using quadractic NURBS basis functions. The point $(0.5, 1) \in Q$ on the "northern" boundary is mapped to the point $(-4, 4) \in \Omega$, which is a corner. We can construct this corner in two different ways.

Using quadratic NURBS basis functions, a basis function is interpolatory at an interior knot that appears twice. Thus, we can construct the corner by using a double knot at at $\xi_1 = 0.5$, which corresponds to the diagonal $x_2 = -x_1$ in the physical domain. We will refer to this parametrization as "K". The geometry and the control polygon are depicted in Fig. 24(c), the complete geometry data is given in Appendix B.2. As a consequence of the double interior knot, the mapping G is not continuously differentiable at the line $\xi_1 = 0.5$. This is indicated in Fig. 24(c), where the edges plotted in blue bend sharply along the diagonal.

We can also construct the corner by placing a double control point at the coordinates $(-4, 4)$, as illustrated in Fig. 24(d). We will refer to this parametrization as "P". The geometry data is given in Appendix B.3. The mapping G remains C^1-continuous in the interior of Q. The disadvantage of this method is that some cells in Ω are strongly distorted, and that the gradient of the mapping can change rapidly. Even though the same, equally spaced subdivision of the parameter domain was used in Fig. 24(c) and Fig. 24(d), the cells in the physical domain are distorted much stronger in Fig. 24(d).

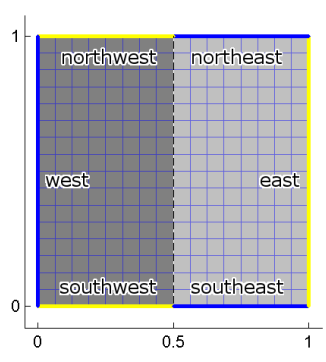
5.2.2 A Method for more Balanced Edge-Splitting in the Physical Domain

A method for compensating this effect is discussed in [8] and can be applied to our subdivisions. It suggests a non-equally spaced subdivision of the parameter domain that leads to a more balanced subdivision of the physical domain.

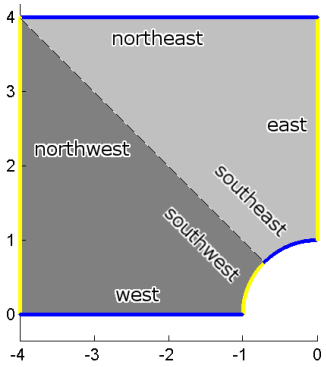
Consider a subdivision of Q and a "vertical column" of cells between the ξ_1-ticks $\xi_1^{(i)}$ and $\xi_1^{(i+1)}$ (like the cells shaded gray in Fig. 25). To refine the whole column, we introduce a new ξ_1-tick $\bar{\xi}_1$ between those two ticks. If the refinement, i.e. the "edge-splitting" was equally spaced in the parameter domain, we would simply choose

$$\bar{\xi}_1 := \frac{\xi_1^{(i)} + \xi_1^{(i+1)}}{2}.$$

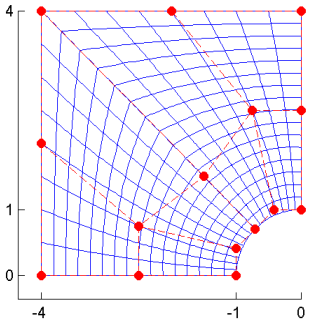
This choice, however, might fail to split the images of the "southern" edge $[\xi_1^{(i)}, \xi_1^{(i+1)}] \times \{0\}$ and the "northern" edge $[\xi_1^{(i)}, \xi_1^{(i+1)}] \times \{1\}$ in half. Even



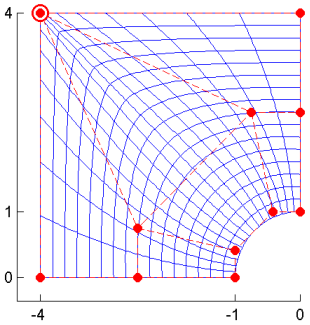
(a) Parameter Domain with equally spaced subdivision.



(b) Physical Domain.



(c) Double knot: $\eta_4 = \eta_5 = 0.5$.



(d) Double control point: $P_{2,3} = P_{3,3} = (-4, 4)$.

Figure 24: Geometry "plate with circular hole". Image of the same equally spaced subdivisions using different knot vectors and control polygons to construct the corner at $(-4, 4)$. For geometry data see Appendix B.2 and B.3.

if we find a coordinate $\bar{\xi}_1$ that splits the "southern image" in half, it might fail to do so with the "northern image", and vice versa (see blue box and green diamond in Fig. 25).

The method proposed in [8] suggests determining $\bar{\xi}_1$ through the following procedure:

Procedure 5.4. (Non-equally spaced ticks-insertion)

- Find the coordinate $\bar{\xi}_a$ such that, in the physical domain, $G(\bar{\xi}_a, 0)$ best approximates the midpoint of the edge from $G(\xi_1^{(i)}, 0)$ to $G(\xi_1^{(i+1)}, 0)$. Likewise, find the coordinate $\bar{\xi}_b$ such that $G(\bar{\xi}_b, 1)$ best approximates the midpoint of the edge from $G(\xi_1^{(i)}, 1)$ to $G(\xi_1^{(i+1)}, 1)$.
- Take the weighted average of $\bar{\xi}_a$ and $\bar{\xi}_b$ to get the ξ_1-coordinate $\bar{\xi}_1$. The weights are the magnitudes of the parametric derivatives at $(\bar{\xi}_a, 0)$ and

$(\bar{\xi}_b, 1)$, i.e.:

$$\bar{\xi}_1 \quad := \quad \frac{\omega_a \bar{\xi}_a + \omega_b \bar{\xi}_b}{\omega_a + \omega_b}$$

where

$$\omega_a := \left\| \frac{\partial G}{\partial \xi_1}(\bar{\xi}_a, 0) \right\|_{l_2} \qquad \text{and} \qquad \omega_b := \left\| \frac{\partial G}{\partial \xi_1}(\bar{\xi}_b, 1) \right\|_{l_2} .$$

The procedure can be applied analogously in the other coordinate direction.

Choosing the parametric derivatives as weights results in a better splitting of the edge where the position in the physical domain changes more rapidly with variation of the parameter.

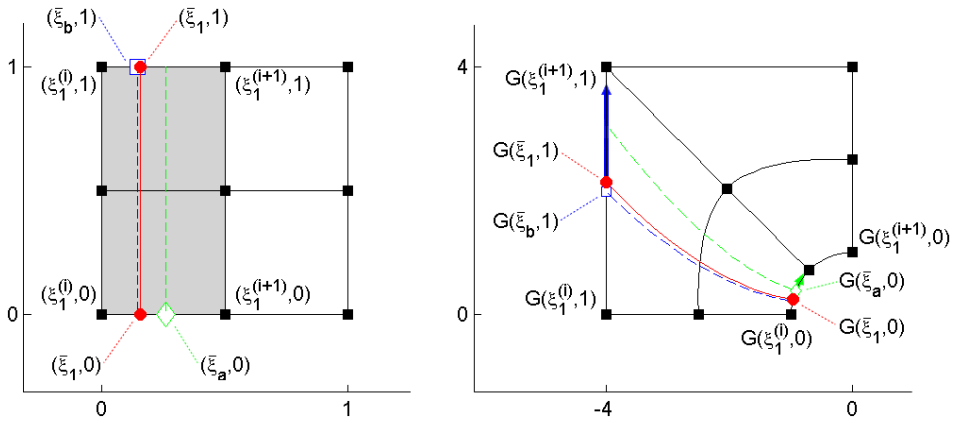


Figure 25: Determination of new ξ_1-tick $\bar{\xi}_1$ between $\xi_1^{(i)} = 0$ and $\xi_1^{(i+1)} = 0.5$ as weighted average of $\bar{\xi}_a$ and $\bar{\xi}_b$. Derivatives with respect to ξ_1 at $(\bar{\xi}_a, 0)$ and $(\bar{\xi}_b, 1)$ indicated by arrows.

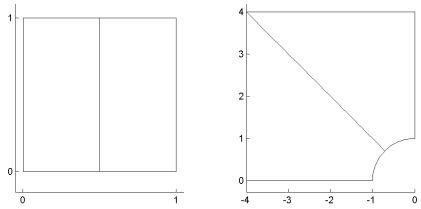
In Fig. 25, the point $G(\bar{\xi}_a, 0)$ marked with the green diamond would provide the best splitting of the "southwestern image", while the point $G(\bar{\xi}_b, 0)$ marked with the blue box would provide the best splitting of the "northwestern image". The dashed lines illustrate how the cells in the physical space would be split, if either of those two coordinates was chosen. As indicated by the arrows, ω_b is much larger than ω_a, thus the calculated coordinate $\bar{\xi}_1$ is closer to $\bar{\xi}_b$ than to $\bar{\xi}_a$. The red dots and the solid, red line indicate the chosen coordinate $\bar{\xi}_1$ and the resulting splitting.

The effect of this method is illustrated in Fig. 26 for the geometry mapping "P" as in Fig. 24(d) and Appendix B.3. Starting from the initial subdivision as depicted in Fig. 26(a), three refinement steps are shown in Fig. 26(b) and Fig. 26(c).

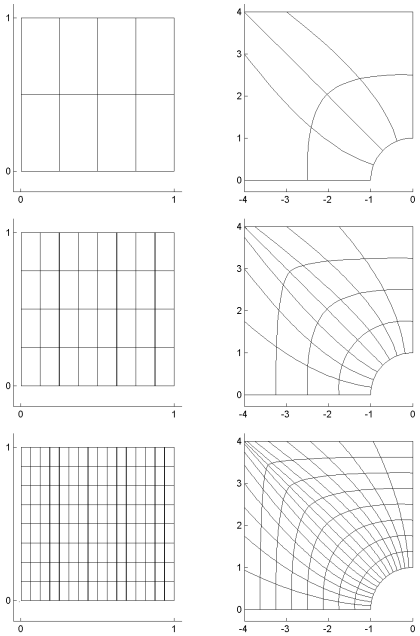
In Fig. 26(b), the cells are equally spaced in the parameter domain, leading to the mentioned unbalanced cell-spacing in the physical domain.

In Fig. 26(c), the coordinates of the edges in the parameter domain were determined through the described procedure, resulting in unequally sized cells in the parameter domain, and in a better balance of cell-sizes near the corner $(-4, 4)$ in the physical domain.

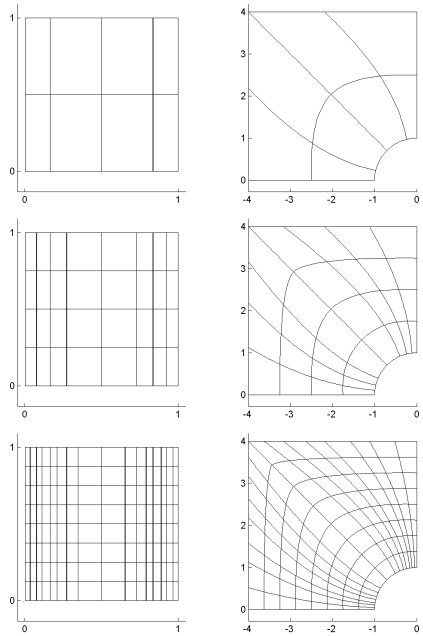
At the circular part of the boundary, however, the effects are opposite: While the circular curve is split equally in Fig. 26(b), the method in Fig. 26(c) leads to a non-equally spaced splitting. As we will see in the numerical tests in Section 7.1, it depends on the area of interest, which method provides better results.



(a) Initial subdivision.



(b) Equally spaced subdivision of parameter domain.



(c) Choice of coordinates via described method, non-equally spaced subdivision of parameter domain.

Figure 26: Comparison of three refinement steps with and without the method presented in Section 5.2.2, starting from the same initial subdivision (a).

5.3 Global Mapping and Finite Element Space

5.3.1 Finite Element Space over the Physical Domain

In the previous sections, we have only defined and discussed the basis functions $\widehat{\varphi}_\alpha^{(i)}$ over the parameter space Q. With G denoting the bijective global geometry mapping that maps the parameter space Q to the physical space $\Omega = G(Q)$, the definition of our basis functions over Ω is rather straightforward:

$$\varphi_\alpha^{(i)}(x) := \widehat{\varphi}_\alpha^{(i)}(G^{-1}(x)).$$

The finite-dimensional function space V_h is then defined as the span of these basis functions:

$$V_h := \operatorname{span}\Big\{\varphi_\alpha^{(i)},\ (i,\alpha) \in \mathcal{I}_P \times I_\alpha\Big\}.$$

Since we require G to be bijective, we know that there exists a global inverse G^{-1}. If J is regular over a cell K, we know from the inverse function theorem that the inverse G^{-1} is continuously differentiable over $G(K)$ [7]. Hence, if the subdivision is chosen in such a way that J is irregular only on edges, but not in the interior of any cells, we have $\varphi_\alpha^{(i)}|_{G(K)} \in H^1(G(K))$ for all cells $K \in \mathcal{K}$. Since the inverse mapping G^{-1} is continuous on Ω, we obtain $\varphi_\alpha^{(i)} \in C(\Omega)$, and therefore $\varphi_\alpha^{(i)} \in H^1(\Omega)$. Hence, $V_h \subset H^1(\Omega)$, and we are working with conforming finite elements [4].

5.3.2 Effects of Non-C^1-Continuous Global Mappings

Recall that we represent the discrete solution as

$$u_h(x) \ = \ \sum_{i \in \mathcal{I}_P} \sum_{\alpha \in I_\alpha} u_\alpha^{(i)} \varphi_\alpha^{(i)}(x).$$

Let $\widehat{u}_h(\xi) := u_h(G(\xi))$ be its transformation to the parameter domain, which we can represent as

$$\widehat{u}_h(\xi) \ = \ \sum_{i \in \mathcal{I}_P} \sum_{\alpha \in I_\alpha} u_\alpha^{(i)} \widehat{\varphi}_\alpha^{(i)}(\xi).$$

From the definition of the basis functions $\widehat{\varphi}_\alpha^{(i)}$ over the parameter domain, we know that

$$\nabla_\xi \widehat{u}_h(\zeta^{(i)}) \ = \ \begin{pmatrix} u_1^{(i)} \\ u_2^{(i)} \end{pmatrix}. \tag{42}$$

For $\beta \in \{1, 2\}$, let J_β denote the parametric derivative of G with respect to ξ_β, i.e.:

$$J_1 := \begin{pmatrix} \partial G_1/\partial \xi_1 \\ \partial G_2/\partial \xi_1 \end{pmatrix} = \begin{pmatrix} J_{11} \\ J_{21} \end{pmatrix},$$
$$J_2 := \begin{pmatrix} \partial G_1/\partial \xi_2 \\ \partial G_2/\partial \xi_2 \end{pmatrix} = \begin{pmatrix} J_{12} \\ J_{22} \end{pmatrix}.$$

Let $x^{(i)}$ be the image of $\zeta^{(i)}$ in the physical domain, i.e. $x^{(i)} := G(\zeta^{(i)})$. With the relation

$$\nabla_\xi = J^T \nabla_x$$

and (42), this means that the coefficient $u_\beta^{(i)}$ is the directional derivative of u_h at node $x^{(i)}$ in the direction J_β.

In Fig. 27, the directions J_1 and J_2 are indicated by light blue and dark blue arrows, respectively, at two nodes $x^{(a)} = G(\zeta^{(a)})$ and $x^{(b)} = G(\zeta^{(b)})$. The depicted geometry is the "plate with circular hole" with the parametrization "K" (with the double knot as in Fig. 24(c) and Appendix B.2). Let $(\xi_1^{(b)}, \xi_2^{(b)})$ be the coordinates of the node $\zeta^{(b)}$ in the parameter domain. The node $x^{(b)}$ in the physical domain marked by the red dot is on the diagonal $x_1 = -x_2$. There, the geometry mapping G is not continuously differentiable with respect to ξ_1, i.e. the directions

$$J_1^-(\zeta^{(b)}) := \lim_{\xi_1 \to \xi_1^{(b)-}} J_1(\xi_1, \xi_2^{(b)}), \qquad J_1^+(\zeta^{(b)}) := \lim_{\xi_1 \to \xi_1^{(b)+}} J_1(\xi_1, \xi_2^{(b)}).$$

are not equal. This is indicated by the two corresponding light blue arrows in Fig. 27. Even though the two directional derivatives are different in general, they are coupled by the coefficient $u_1^{(b)}$ of the discrete solution.

This coupling of different directional derivatives can appear at all nodes where the geometry mapping is not continuously differentiable. Thus, we have to expect inaccuracies of the discrete solution in cells that have such a node as an ancestor. This is confirmed in the numerical experiments in Sections 7.1 and 7.2.

Possible approaches for solving this problem could be the use of additional degrees of freedom or other basis functions in such areas, or the definition of internal interfaces. This, of course, raises the question of how to detect such areas automatically and efficiently. These mentioned approaches were not implemented in the course of this thesis.

5.4 Isogeometric Aspects

Recall that the functions in X_h are globally C^1-continuous and piecewise bi-cubic. As mentioned in Section 5.1, if all weights $w_{i,j}$ of the geometry

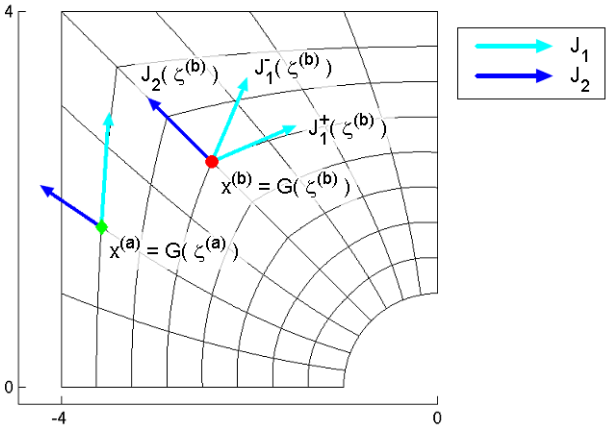


Figure 27: Parametric derivatives of a geometry mapping that is not globally C^1-continuous.

mapping G are equal, the NURBS basis functions in (41) can be written as

$$\mathbf{R}_{(i,j)}^{(p,p^*)}(\xi_1,\xi_2) \;=\; B_{i,p}(\xi_1)\cdot B^*_{j,p^*}(\xi_2).$$

If the degrees p and p^* are both less or equal to 3, the NURBS basis functions $\mathbf{R}_{(i,j)}^{(p,p^*)}$ are in X_h, i.e. they can be exactly represented as linear combinations of the basis functions $\widehat{\varphi}_\alpha^{(i)}$ of X_h. Hence, we are working in an isogeometric setting in that sense that the discrete solution u_h can be expressed by the NURBS basis functions and new coefficients $u'_{i,j}$:

$$u_h(x) \;=\; \sum_{i\in\mathcal{I}_P}\sum_{\alpha\in I_\alpha} u_\alpha^{(i)}\cdot\left(\widehat{\varphi}_\alpha^{(i)}\circ G^{-1}\right)(x) \tag{43}$$

$$\;=\; \sum_{i=1}^{n}\sum_{j=1}^{n^*} u'_{i,j}\cdot\left(\mathbf{R}_{(i,j)}^{(p,p^*)}\circ G^{-1}\right)(x). \tag{44}$$

If not all weights are equal, the denominator in (41) is non-trivial and this is no longer the case. We can write (41) as

$$\mathbf{R}_{(i,j)}^{(p,p^*)}(\xi_1,\xi_2) \;=\; \frac{R_N(\xi_1,\xi_2)}{R_D(\xi_1,\xi_2)},$$

where the nominator R_N and denominator R_D are defined over Q as follows:

$$R_N(\xi_1,\xi_2) \;:=\; B_{i,p}(\xi_1)\cdot B^*_{j,p^*}(\xi_2)\cdot w_{i,j},$$

$$R_D(\xi_1,\xi_2) \;:=\; \sum_{k=1}^{n}\sum_{l=1}^{n^*} B_{k,p}(\xi_1)\cdot B^*_{l,p^*}(\xi_2)\cdot w_{k,l}.$$

Dividing the basis functions of X_h by the denominator R_D, we can define the functions

$$\widehat{\varphi}^{(i)}_{\alpha,D}(\xi) \;:=\; \frac{\widehat{\varphi}^{(i)}_{\alpha}(\xi)}{R_D(\xi)}$$

and the function space

$$X_{h,D} \;:=\; \operatorname{span}\left\{\widehat{\varphi}^{(i)}_{\alpha,D},\ (i,\alpha) \in \mathcal{I}_P \times I_\alpha\right\}.$$

If $p \leq 3$ and $p^* \leq 3$, the numerator $R_N(\xi_1, \xi_2)$ again is bi-cubic, i.e. $R_N \in X_h$, and therefore

$$\mathbf{R}^{(p,p^*)}_{(i,j)} \in X_{h,D}$$

for all i, j. This way, we are in an isogeometric setting again, i.e. we can calculate a discrete solution $u_{h,D} \in X_{h,D}$ and express this discrete solution as a linear combination of the NURBS basis functions analogously to (43) and (44):

$$\begin{aligned} u_{h,D}(x) &= \sum_{i \in \mathcal{I}_P} \sum_{\alpha \in I_\alpha} u^{(i)}_{\alpha,D} \cdot \left(\widehat{\varphi}^{(i)}_{\alpha,D} \circ G^{-1}\right)(x) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n^*} u'_{i,j,D} \cdot \left(\mathbf{R}^{(p,p^*)}_{(i,j)} \circ G^{-1}\right)(x). \end{aligned}$$

The computational examples discussed in Section 7 were carried out using both function spaces X_h and $X_{h,D}$. In these examples, however, no significant differences between the discrete solutions u_h and $u_{h,D}$ were detected.

6 A Posteriori Error Estimator

6.1 General Remarks re Error Estimators

6.1.1 Definition, Reliability and Efficiency

The following definitions are given as in [3]:

Let u denote the (unknown) exact solution of the variational problem. A quantity η which approximates $\|u-u_h\|$ in an appropriate norm $\|\cdot\|$ is called *a posteriori error estimator*, or *error estimator* for brevity, if it is a function of known quantities of the data and the discrete solution u_h.

For an error estimator to be useful, it must be possible to compute η easily, and η must provide lower and upper bounds of the error in the following sense:

Definition 6.1. An error estimator η is called *reliable* if

$$\|u - u_h\| \leq C_{\text{rel}} \cdot \eta + \text{higher order terms} \tag{45}$$

with C_{rel} being independent of the cell sizes. This condition guarantees that the error estimator will only tend to zero, if the true error $\|u - u_h\|$ tends to zero.

Definition 6.2. An error estimator η is called *efficient* if

$$\eta \leq C_{\text{eff}} \cdot \|u - u_h\| + \text{higher order terms} \tag{46}$$

with C_{eff} being independent of the cell sizes. This condition guarantees that, if the true error $\|u-u_h\|$ tends to zero, this must be detected and indicated by η tending to zero as well.

The "higher order terms" in (45) and (46) can be any additional terms that could depend on known or unknown data. They are not the same in the two equations, but both are much smaller than the error $\|u-u_h\|$. Let $(\mathcal{K}_h)_{h\in\mathcal{H}}$ denote a family of subdivisions with a maximal cell-size h in a parameter set $\mathcal{H} \subset (0,\infty)$. Then, the higher order terms can be defined by the property

$$\lim_{h\to 0} \frac{\text{higher order terms}}{\|u - u_h\|} \;=\; 0.$$

6.1.2 Local Error Estimation and Local Refinement

If we can calculate the global error estimator η as a sum of cell-wise computable contributions η_K, such as, for example,

$$\eta = \sum_{K \in \mathcal{K}} \eta_K, \quad \text{or} \quad \eta^2 = \sum_{K \in \mathcal{K}} \eta_K^2, \tag{47}$$

we can identify the cells with the highest contributions η_K to the global error. Then we can refine those cells $K \in \mathcal{K}$ for which the criterion

$$\eta_K \ > \ \theta$$

is fulfilled with a certain threshold θ. We will use two choices for determining this threshold that will be referred to as "Value" and as "Quantile":

$$\text{Value}: \quad \theta \ = \ \alpha \cdot \max_{K \in \mathcal{K}} \{\eta(K)\}, \quad 0 < \alpha < 1$$

In this setting, obviously, all cells K for which η_K is larger than α-times the largest appearing local error are refined. The actual number of refined cells varies.

$$\text{Quantile}: \quad \theta \ = \ \alpha\text{−quantile}_{K \in \mathcal{K}} \{\eta(K)\}, \quad 0 < \alpha < 1$$

In this case, θ is chosen such that $(1-\alpha) \cdot 100$ percent of all cells are refined (for example, if $\alpha = 0.85$, then those 15% of all cells will be refined that have the largest local errors η_K).

For brevity, we will also refer to these two criterions with "Val[α]" and "Qua[α]".

6.2 Implemented Error Estimator

6.2.1 Basic Concept and Implementation with Bubble Functions

The concept discussed in [1] is based on enlarging of the function space V_h with additional basis functions. Let $B_h \subset V$ denote the space that is spanned by those additional basis functions, then we can define

$$\bar{V}_h \ := \ V_h \oplus B_h \ \subset V.$$

Solving the differential equation in $\bar{V}_h$ leads to a solution $\bar{u}_h \in \bar{V}_h$, i.e.

$$a(\bar{u}_h, \bar{v}_h) \ = \ \langle F, \bar{v}_h \rangle \quad \forall\, \bar{v}_h \in \bar{V}_h \tag{48}$$

where $\bar{u}_h$ can be expressed as

$$\bar{u}_h \;=\; u_h + e_h, \quad u_h \in V_h,\ e_h \in B_h. \tag{49}$$

Since $\bar{V}_h$ is a better approximation of V than V_h, one would expect $\bar{u}_h$ to be a better approximation of the exact solution than u_h. Therefore, if the error $\|\bar{u}_h - u_h\| = \|e_h\|$ is large in some areas of the computational domain, this might indicate that the error $\|u - u_h\|$ is large in those areas as well.

The question is how to compute $\|e_h\|$. One could do this by assembling the enlarged stiffness matrix for (48) and solving the larger system of equations, thereby calculating e_h. A much more efficient method is possible with the following choice of B_h:

We enlarge V_h by the so-called *bubble functions*. A univariate bubble function is defined by

$$\widehat{\beta}_{a,b}(\xi) \;:=\; \begin{cases} \dfrac{\xi - a}{b - a} \cdot \dfrac{b - \xi}{b - a}, & \xi \in (a, b) \\ 0 \qquad\qquad\quad , & \text{else} \end{cases}$$

For each cell $K \in \mathcal{K}$, we define the bivariate bubble function $\widehat{\beta}^{(K)}$:

$$\widehat{\beta}^{(K)}(\xi_1, \xi_2) \;:=\; \widehat{\beta}_{a_1,b_1}(\xi_1) \cdot \widehat{\beta}_{a_2,b_2}(\xi_2), \quad \text{where } K = (a_1, b_1) \times (a_2, b_2).$$

We define their transformations to the physical domain and B_h as their span:

$$\begin{aligned} \beta^{(K)}(x) &:= \widehat{\beta}^{(K)}(G^{-1}(x)) \\ B_h &:= \operatorname{span}\Big\{\beta^{(K)},\ K \in \mathcal{K}\Big\}. \end{aligned}$$

These bubble functions $\widehat{\beta}^{(K)}$ have the following important properties:

- The bubble functions are continuous over Q and piecewise in H^1 (i.e. $\beta^{(K)}|_{K'} \in H^1(K')$ for all $K' \in \mathcal{K}$). Hence, we have $\beta^{(K)} \in H^1(Q)$ for all $\beta^{(K)} \in B_h$ [4].
- The support of $\widehat{\beta}^{(K)}$ is contained in K, and two different bubble functions have disjoint support. Therefore, we have

$$a(\beta^{(K)}, \beta^{(K')}) \;=\; 0, \quad \text{for } K \neq K'. \tag{50}$$

The function $e_h \in B_h$ in (49) can be expressed as

$$e_h \;=\; \sum_{K \in \mathcal{K}} e^{(K)} \beta^{(K)}, \quad e^{(K)} \in \mathbb{R}.$$

For a fixed $K \in \mathcal{K}$, we set $\bar{v}_h := \beta^{(K)} \in \bar{V}_h$ in (48). Then, it follows that

$$\begin{aligned} a(u_h, \beta^{(K)}) + \sum_{K' \in \mathcal{K}} e^{(K')} \underbrace{a(\beta^{(K')}, \beta^{(K)})}_{=0 \text{ for } K \neq K', \ (50)} &= \langle F, \beta^{(K)} \rangle, \\ a(u_h, \beta^{(K)}) + e^{(K)} a(\beta^{(K)}, \beta^{(K)}) &= \langle F, \beta^{(K)} \rangle. \end{aligned}$$

Thus, we can calculate the coefficients $e^{(K)}$ of e_h, as well as the energy norm of e_h from known quantities [14]:

$$\begin{aligned} e^{(K)} &= \frac{1}{a(\beta^{(K)}, \beta^{(K)})} \Big(\langle F, \beta^{(K)} \rangle - a(u_h, \beta^{(K)}) \Big), \\ \|e_h\|_E^2 &= \sum_{K \in \mathcal{K}} (e^{(K)})^2 a(\beta^{(K)}, \beta^{(K)}). \end{aligned}$$

We use this to define our a posteriori error estimator as the following cell-wise computable sum in the form as in (47):

$$\eta^2 := \|e_h\|_E^2 = \sum_{K \in \mathcal{K}} \eta_K^2 \tag{51}$$

$$\text{where} \quad \eta_K := \frac{\langle F, \beta^{(K)} \rangle - a(u_h, \beta^{(K)})}{\|\beta^{(K)}\|_E}. \tag{52}$$

6.2.2 Reliability and Efficiency

Omitting the higher order terms, the condition for the reliability of the error estimator can be re-formulated as

$$R_{\text{rel}} := \frac{\|u - u_h\|}{\eta} \leq C_{\text{rel}}. \tag{53}$$

In the numerical examples in Section 7 in which the exact solution is known, we will plot R_{rel} as an indicator for the reliability of η.

The efficiency of the error estimator can be shown explicitly. For this, we define the restriction of the bilinear form $a(\cdot, \cdot)$ to the image of a cell K:

$$a(u, v)_K := \int_{G(K)} (\nabla v)^T A \nabla u + (b^T \nabla u) v + cuv \, dx$$

and the corresponding restricted energy norm:

$$\|v\|_{E,K} := \sqrt{a(v, v)_K}.$$

We can write

$$a(u, v) = \sum_{K \in \mathcal{K}} a(u, v)_K \quad \text{and} \quad \|v\|_E^2 = \sum_{K \in \mathcal{K}} \|v\|_{E,K}^2.$$

In particular, because the support of a bubble function $\beta^{(K)}$ is contained in K, we also have for all $v \in V$ and for all $K \in \mathcal{K}$:

$$a(v, \beta^{(K)}) \;=\; a(v, \beta^{(K)})_K \quad \text{and} \quad \|\beta^{(K)}\|_E^2 \;=\; \|\beta^{(K)}\|_{E,K}^2.$$

With this, we derive from (52)

$$\begin{aligned} \eta_K \;&=\; \frac{1}{\|\beta^{(K)}\|_{E,K}} \Big(\underbrace{\langle F, \beta^{(K)} \rangle}_{=a(u,\beta^{(K)})} - a(u_h, \beta^{(K)}) \Big) \\ &=\; \frac{1}{\|\beta^{(K)}\|_{E,K}}\, a(u - u_h, \beta^{(K)})_K, \end{aligned} \tag{54}$$

which we use to show the efficiency of η:

Lemma 6.3. *The a posteriori error estimator defined in (51) and (52) is efficient.*

Proof. Since $a(\cdot,\cdot)$ is bounded, it follows from (54) that

$$\eta_K \;\le\; \frac{1}{\|\beta^{(K)}\|_{E,K}}\, \mu_2 \|u - u_h\|_{1,K} \|\beta^{(K)}\|_{1,K},$$

where $\|\cdot\|_{1,K}$ denotes the H^1-norm on the image $G(K)$ of K and where the constant μ_2 does not depend on the particular cell K (see Appendix A). Since also $\|\beta^{(K)}\|_{1,K} = \|\beta^{(K)}\|_1$, it follows that

$$\eta_K \;\le\; \frac{\|\beta^{(K)}\|_1}{\|\beta^{(K)}\|_E}\, \mu_2 \|u - u_h\|_{1,K} \;\le\; \frac{\mu_2}{\sqrt{\mu_1}}\, \|u - u_h\|_{1,K}.$$

Therefore,

$$\eta^2 \;\le\; \frac{\mu_2^2}{\mu_1} \sum_{K \in \mathcal{K}} \|u - u_h\|_{1,K}^2 \;=\; \frac{\mu_2^2}{\mu_1}\, \|u - u_h\|_1^2 \;\le\; \frac{\mu_2^2}{\mu_1^2}\, \|u - u_h\|_E^2,$$

i.e. (46) holds with $C_{\text{eff}} = \mu_2/\mu_1$. If $a(\cdot,\cdot)$ is symmetric, it follows from (54) that (46) holds with $C_{\text{eff}} = 1$, because in this case $|a(u,v)_K| \le \|u\|_{E,K} \|v\|_{E,K}$ (see also Appendix A). □

7 Computational Examples

7.1 Plate with Circular Hole

We compare the aspects of the the geometry mapping "plate with circular hole" that were discussed in Sections 5.2, 5.3, and 5.4.

7.1.1 Problem Setting

The physical domain Ω is depicted in Fig. 28. We solve the partial differential equation

$$\begin{aligned} -\Delta u + u &= f \\ \nabla u \cdot n &= g_N \quad \text{on } \Gamma_N := \partial\Omega \end{aligned}$$

where f and g_N are determined by the following exact solution:

$$\begin{aligned} u: \quad \Omega &\to \mathbb{R} \\ x &\mapsto -\log|x - x_0| \end{aligned}$$

The function u has a singularity at x_0. This enables us to focus on two different parts of the subdivision with two different choices of x_0 (see Fig. 28):

Case 1: $x_0 = (-0.75, 0.55)$ near the circular hole.

Case 2: $x_0 = (-1.75, 4.25)$ near the image of the "northern boundary".

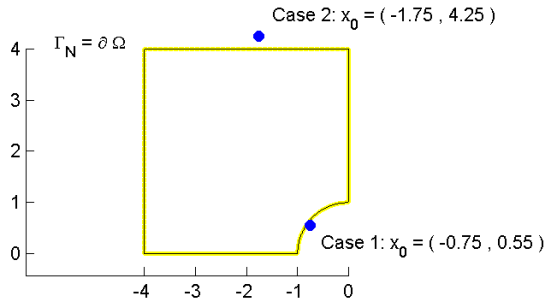


Figure 28: Positions of x_0 for example "plate with circular hole".

7.1.2 Geometry Mappings

As in Section 5.2, the two different parametrizations will be referred to as "K" (with the double interior knot) and "P" (with the double control point at the corner). The corresponding control nets are depicted in Fig. 24(c) and Fig. 24(d), the geometry data is given in Appendix B.2 and B.3, respectively.

The abbreviation "param" will be referring to an equally-spaced edge-splitting in the parameter domain, while "phys" will be referring to the method introduced in Section 5.2.2 for a more balanced splitting in the phyiscal domain.

7.1.3 Numerical Results, Uniform Refinement

The results for uniform refinements are illustrated in Fig. 29. The errors $\|u - u_h\|$ in the L_2-norm and the energy norm are plotted versus the degrees of freedom (DoF).

It can be seen in Fig. 24(c) and Fig. 27 that the parametrization "K" already provides a balanced splitting of the edges in the physical domain. Accordingly, the subdivisions generated with "param" and "phys" and the corresponding discrete solutions are very similar.

With the parametrization "P", however, the differences between "param" and "phys" can be seen clearly. "Param" provides a balanced splitting of the circular part of the boundary, while "phys" fails to properly resolve this part in coarse subdivisions. This is very well illustrated by the errors plotted in Fig. 29(c) and Fig. 29(e) for Case 1.

In Case 2, where the singularity is situated near the "northern" boundary, the results are opposite, as "param" does not resolve the area near the singularity very well. The corresponding errors are plotted in Fig. 29(d) and Fig. 29(f).

In both cases, it can be observed that the convergence rate with parametrization "K" is slower than that with "P", due to the reduced smoothness of the parametrization "K" and the coupling of different directional derivatives that was discussed in Section 5.3. The results of the adaptive refinement will visualize this problem even clearer.

Note that these two cases were specifically chosen to illustrate the different effects of equally spaced and non-equally spaced edge splitting in the parameter domain. In general, one can not automatically predict a priori which method is better suited for a specific problem. The results, however,

indicate that the convergence rates are the same after a certain fineness has been reached.

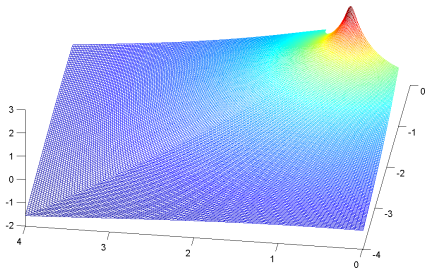
The ratios $R_{\text{rel}} = \|u - u_h\|/\eta$ calculated with the L_2-norm and the energy norm for the used parametrizations and refinement criteria are plotted in Fig. 30. The numerical results indicate that the condition for the reliability of the a posteriori error estimator, namely $R_{\text{rel}} \leq C_{\text{rel}}$ for a constant C_{rel} is fulfilled.

In Section 5.4, we discussed that we can divide the basis functions $\widehat{\varphi}_{\alpha}^{(i)}$ of X_h by the denominator of the NURBS geometry mapping, thereby defining basis functions $\widehat{\varphi}_{\alpha,D}^{(i)}$ which span the function space $X_{h,D}$. The examples were also computed using this function space $X_{h,D}$, resulting in discrete solutions which we denote by $u_{h,D}$. As a representative example, the errors $\|u-u_h\|$ and $\|u-u_{h,D}\|$ for Case 1 are compared in Fig. 31, and as mentioned in Section 5.4, the differences are insignificant.

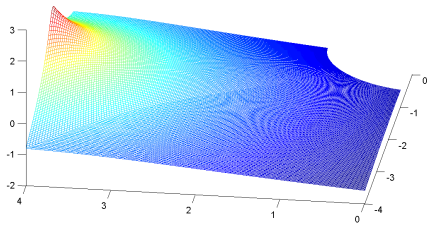
7.1.4 Numerical Results, Adaptive Refinement

The tests with adaptive refinement were only carried out using "param". The chosen refinement criteria were Qua[0.70], Qua[0.85] and Val[0.30]. The errors are plotted in Fig. 32, the refined subdivisions in Fig. 33 for Case 1, and in Fig. 34 for Case 2.

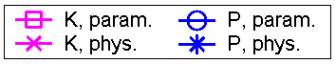
With parametrization "K", in both cases and for all three criteria, the error estimator marked the cells along the diagonal, i.e. where the geometry mapping is not continuously differentiable, as those with the highest error. Hence, the refined subdivisions and the errors are similar for all three refinement criteria. As we have seen in the previous Section 7.1.3, the parametrization "P" with "param" does not properly resolve the singularity in Case 2. Still, it only takes a few refinement steps to obtain solutions that provide a better approximation of the exact solution than those calculated with "K".

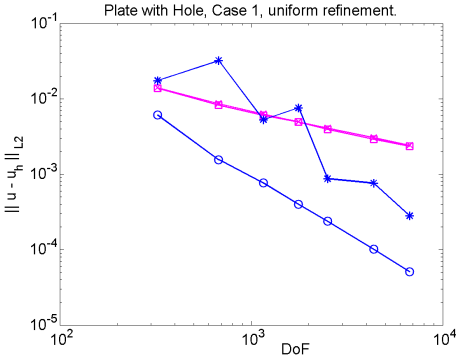


(a) Solution for Case 1.

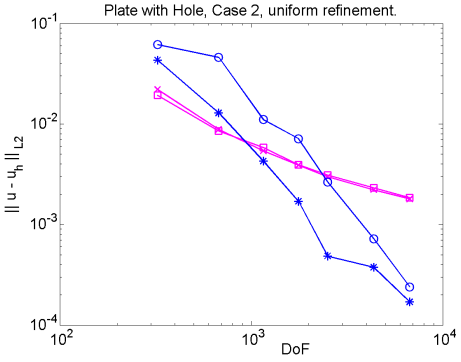


(b) Solution for Case 2.

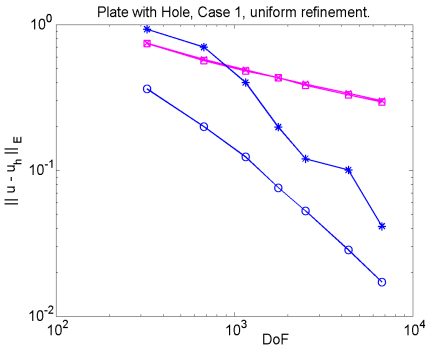




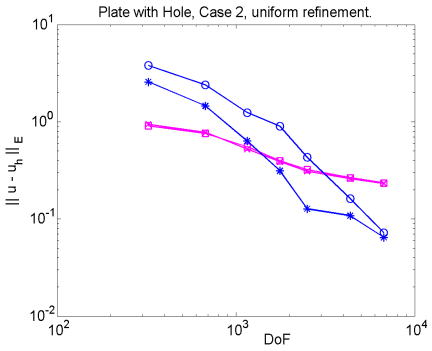
(c) Error in L_2-norm, Case 1.



(d) Error in L_2-norm, Case 2.

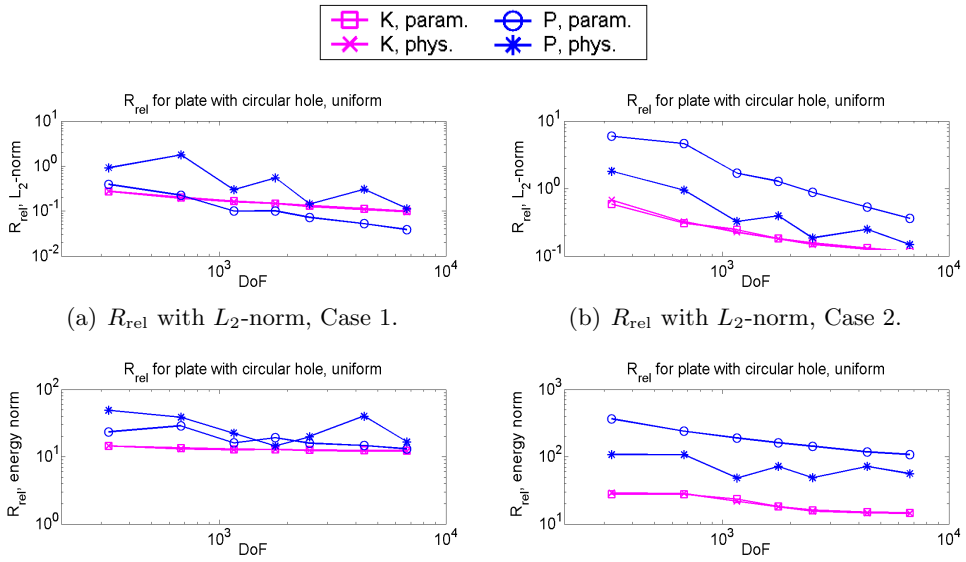


(e) Error in energy norm, Case 1.



(f) Error in energy norm, Case 2.

Figure 29: Comparison of uniform refinement of “plate with circular hole” with parametrizations “P” and “K”, and different edge-splitting-methods “param” and“phys”.



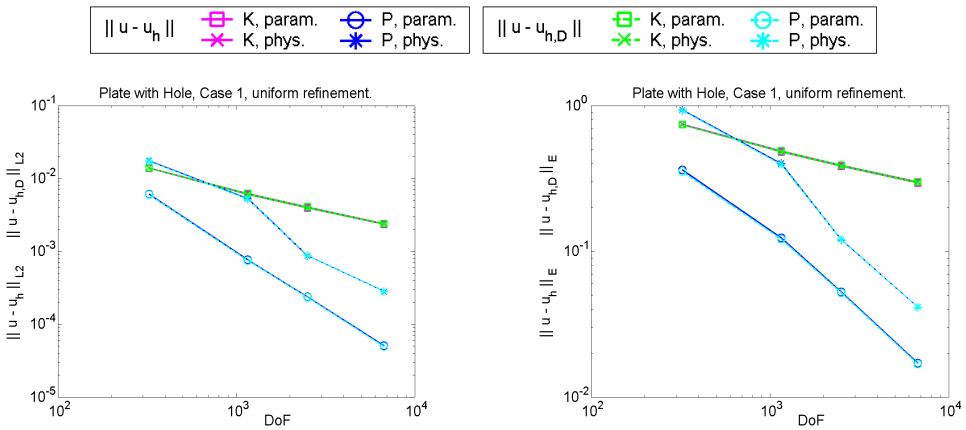
(a) R_{rel} with L_2-norm, Case 1.

(b) R_{rel} with L_2-norm, Case 2.

(c) R_{rel} with energy norm, Case 1.

(d) R_{rel} with energy norm, Case 2.

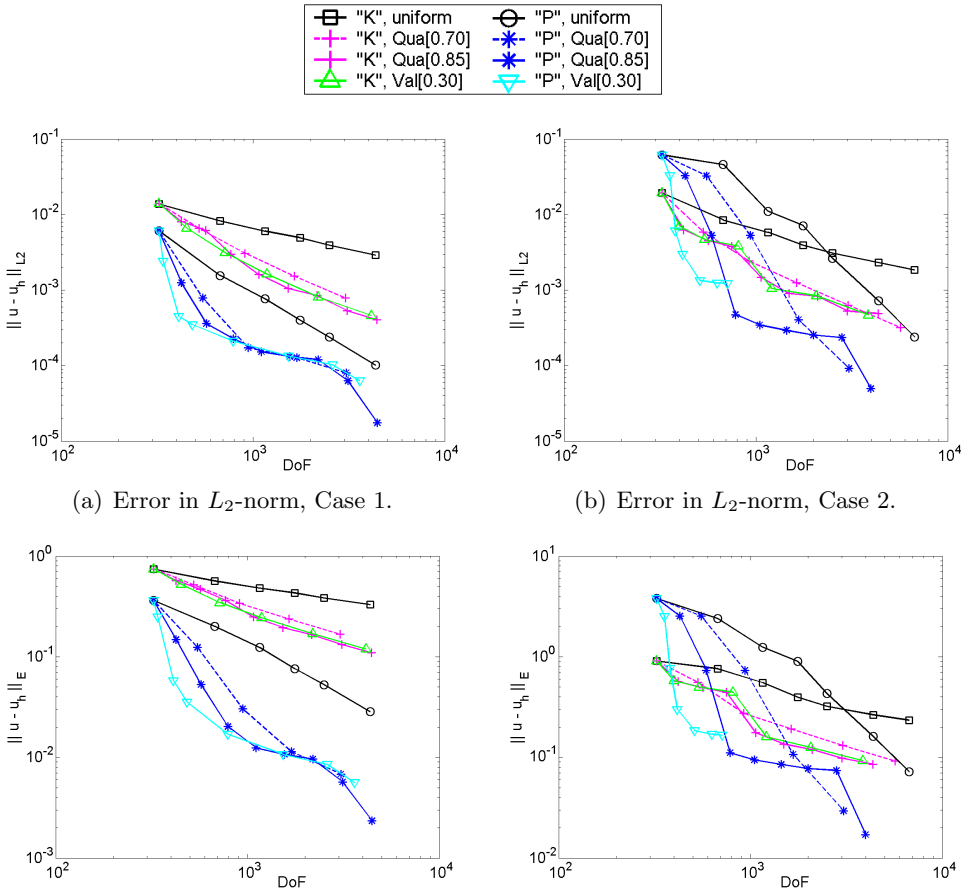
Figure 30: R_{rel} for "plate with circular hole", uniform ref'.



(a) Comparison in L_2-norm.

(b) Comparison in energy norm.

Figure 31: Comparison of $\|u - u_h\|$ and $\|u - u_{h,D}\|$ in Case 1.



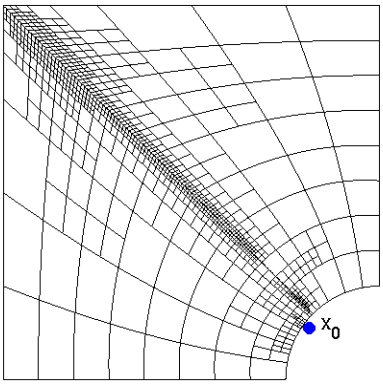
(a) Error in L_2-norm, Case 1.

(b) Error in L_2-norm, Case 2.

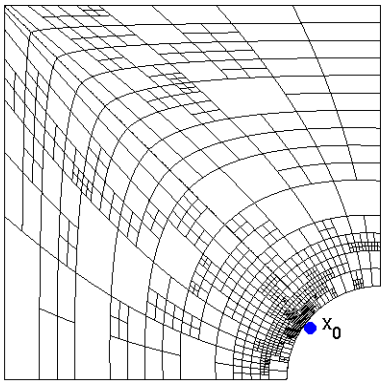
(c) Error in energy norm, Case 1.

(d) Error in energy norm, Case 2.

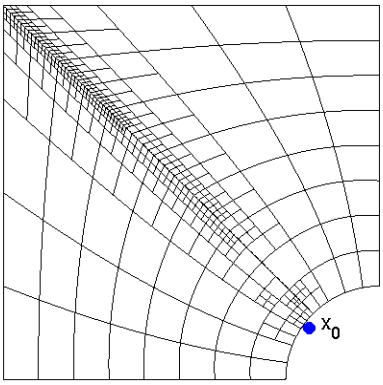
Figure 32: Errors in L_2-norm and energy norm for uniform and adaptive refinement with different refinement criteria in Case 1 and Case 2.



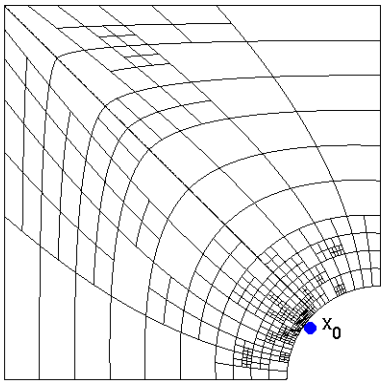
(a) “K”, Qua[0.70], after 4 ref’: 3036 DoF.



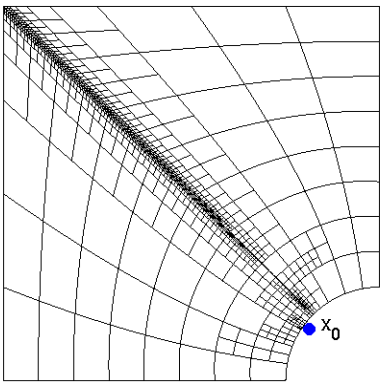
(b) “P”, Qua[0.70], after 4 ref’s: 3068 DoF.



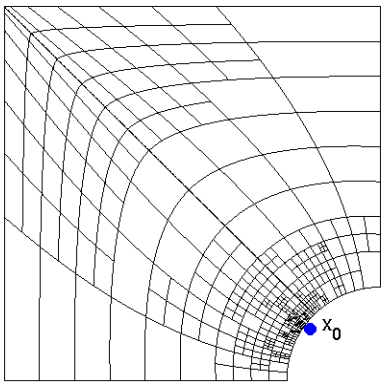
(c) “K”, Qua[0.85], after 5 ref’s: 1528 DoF.



(d) “P”, Qua[0.85], after 5 ref’s: 1556 DoF.

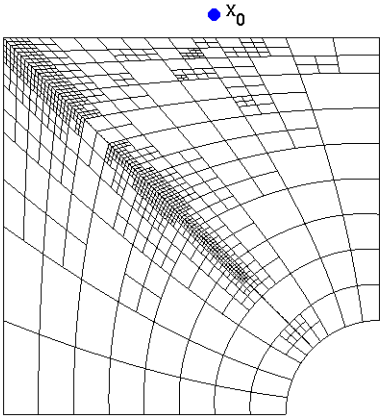


(e) “K”, Val[0.30], after 5 ref’s: 4152 DoF.

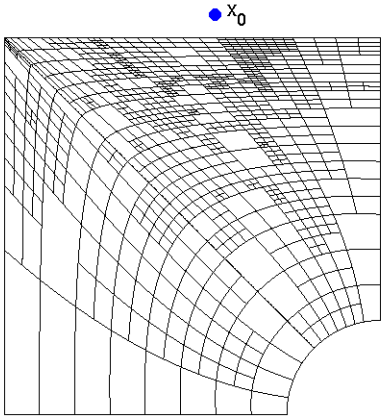


(f) “P”, Val[0.30], after 5 ref’s: 1532 DoF.

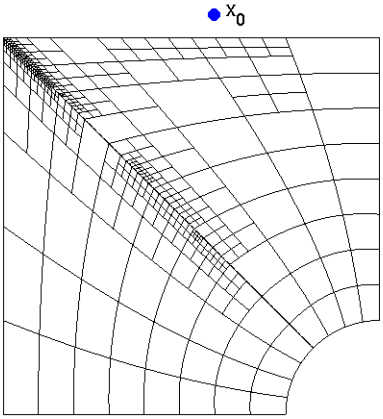
Figure 33: Refined subdivisions of the domain “plate with circular hole” in Case 1 (singularity marked by blue dot near circular hole).



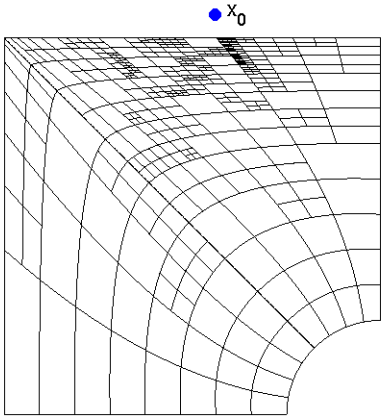
(a) “K”, Qua[0.70], after 4 ref’: 3024 DoF.



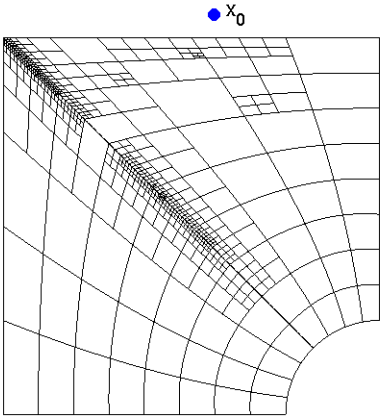
(b) “P”, Qua[0.70], after 4 ref’s: 3060 DoF.



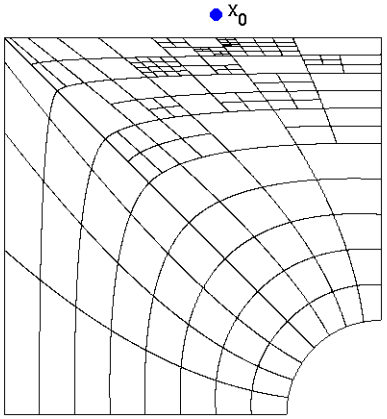
(c) “K”, Qua[0.85], after 5 ref’s: 1488 DoF.



(d) “P”, Qua[0.85], after 5 ref’s: 1448 DoF.



(e) “K”, Val[0.30], after 5 ref’s: 2064 DoF.



(f) “P”, Val[0.30], after 5 ref’s: 628 DoF.

Figure 34: Refined subdivisions of the domain “plate with circular hole” in Case 2 (singularity marked by blue dot near “northern boundary”).

7.2 Laplace Equation in L-Shaped Domain

The physical domain and the exact solution are as in [6].

7.2.1 Problem Setting

We solve the Laplace equation

$$\Delta u \;=\; 0$$

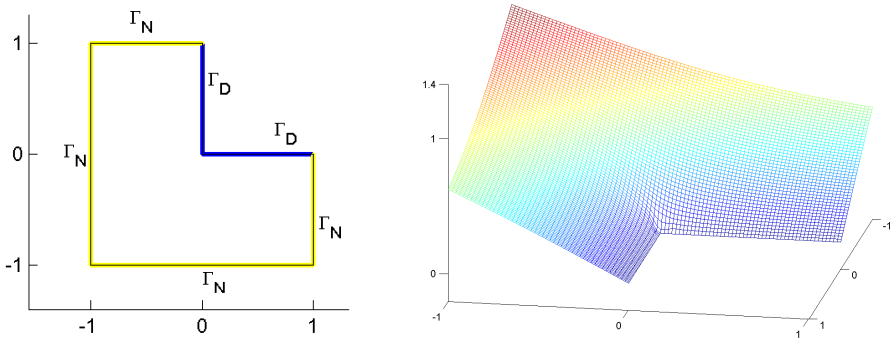
on the L-shaped domain $\Omega = [-1,1]^2 \backslash [0,1]^2$ that is illustrated in Fig. 35(a).

The exact solution in polar coordinates (see also Fig. 35(b)) is given by

$$\begin{aligned} u: \quad \mathbb{R}^+ \times (0, 2\pi] \quad &\to \quad \mathbb{R} \\ (r, \varphi) \quad &\mapsto \quad r^{\frac{2}{3}} \sin\left(\frac{2\varphi - \pi}{3}\right) \end{aligned}$$

which, after tranformation to Cartesian coordinates, satisfies the Laplace equation. The boundary conditions are set as follows (see also Fig. 35(a)):

$$\begin{aligned} \Gamma_D &= \{0\} \times [0,1] \cup [0,1] \times \{0\} \\ g_D &= f\left(r, \frac{\pi}{2}\right) = f\left(r, \frac{3\pi}{2}\right) = 0 \\ \Gamma_N &= \partial\Omega \backslash \Gamma_D \\ g_N &\quad \text{determined by exact solution } u \end{aligned}$$



(a) Physical domain with Dirichlet boundary Γ_D (blue) and Neumann boundary Γ_N (yellow).

(b) Exact solution.

Figure 35: Physical domain, Dirichlet- and Neumann-boundaries, and exact solution for the numerical test "Laplace equation on L-shaped domain".

7.2.2 Geometry Mappings

We compare a total of four different parametrizations that realize the corners at $(0, 0)$ and $(-1, -1)$ differently. The parametrization "K" uses a double knot, while "Pa", "Pb", and "Pc" use double control points. The corresponding control polygons are depicted in Fig. 36. While both "Pa" and "Pb" use the same control polygon, different knot vectors result in different distortions of the cells which is illustrated by the blue lines in Fig. 36(b) and Fig. 36(c). With "Pa" and "Pb", the parametric derivative with respect to ξ_1 vanishes on the whole diagonal $x_1 = x_2$. With "Pc" this is only the case at the two corners, but not in the interior of the domain.

The following table provides an overview of the parametrizations and where to find the complete data:

Parametrization	Control polygon	Complete geometry data
"K"	Fig. 36(a)	Appendix B.4
"Pa"	Fig. 36(b)	Appendix B.5
"Pb"	Fig. 36(c)	Appendix B.5
"Pc"	Fig. 36(d)	Appendix B.6

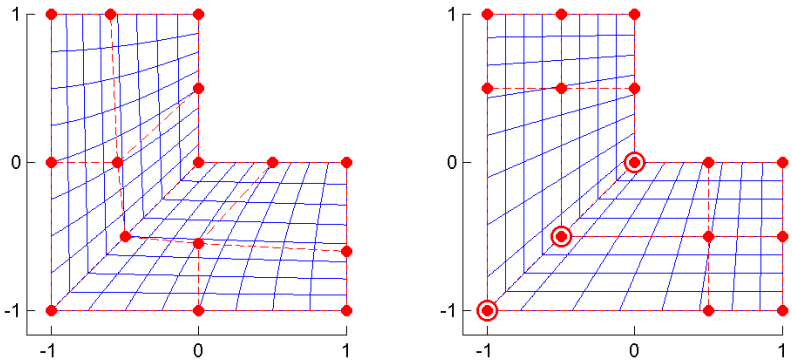
7.2.3 Numerical Results

The errors $\|u - u_h\|$ for uniform refinement and adaptive refinement with the criterion Qua[0.72], starting from an 8×4-subdivision, are plotted in Fig. 37, and illustrate the differences between the chosen parametrizations.

Similar to the case in the previous example "plate with circular hole", the parametrization "K" is not continuously differentiable along the diagonal $x_1 = x_2$. The subdivision in Fig. 38(a) clearly shows that mainly the cells along the diagonal were marked for refinement.

The parametrizations "Pa" and "Pb" use the same control polygon. The subdivisions in Fig. 38(b) and Fig. 38(c) illustrate how the different knot vectors influence the refinement processes, while Fig. 37(a) and Fig. 37(b) indicate that the differences between the corresponding discrete solutions decrease as the subdivisions get finer.

With "Pb", the error estimator mainly marks the cells near the corner $(0, 0)$ for refinement (see Fig. 38(c)). With "Pa", however, the cells on both sides of the diagonal are marked for refinement, which can be seen in Fig. 38(b). In Fig. 39(a) the basis functions resulting from the knot vector for "Pa" are plotted. The derivatives of the basis functions $B_{3,2}$ and $B_{4,2}$ quickly change near the coordinates 0.45 and 0.55. These coordinates correspond to the



(a) “K” (b) “Pa”

(c) “Pb” (d) “Pc”

Figure 36: Control polygons for parametrizations “K”, “Pa”, “Pb”, and “Pc”.

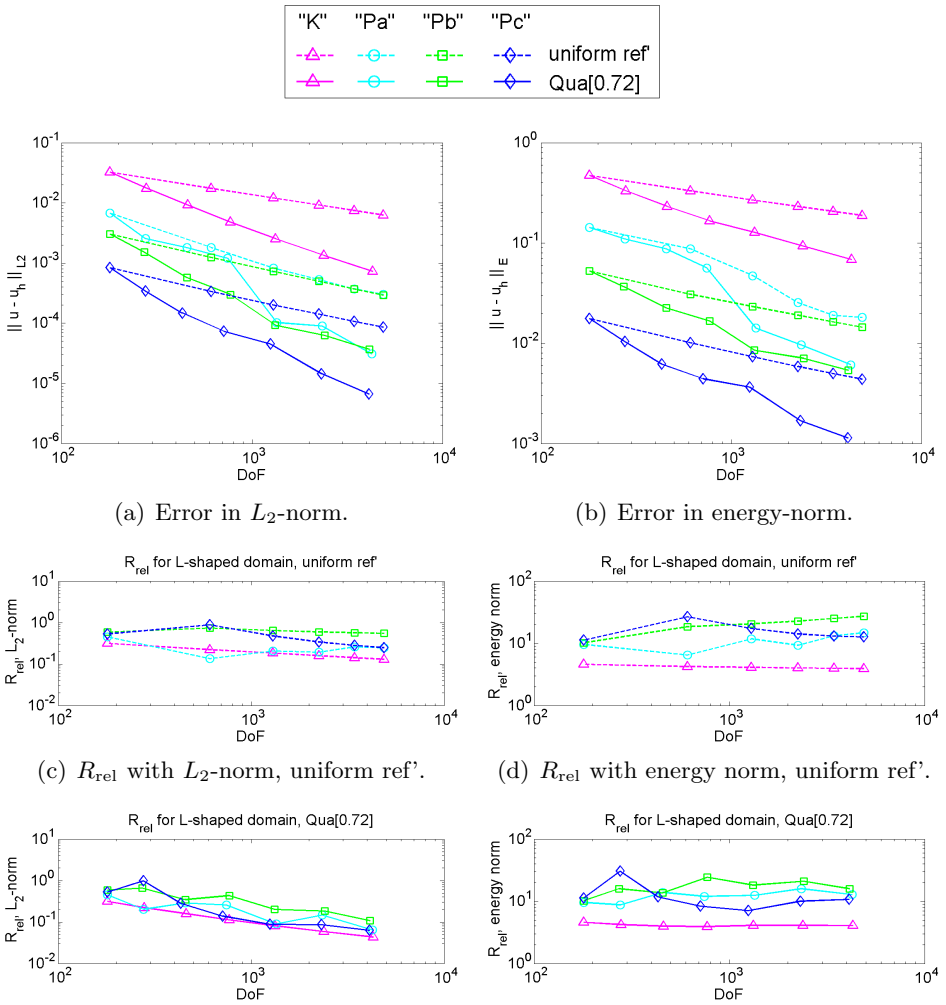
refined areas alongside the diagonal. To check whether this phenomenon is specific for the used error estimator, another test was carried out, using the true local error as a refinement criterion, i.e. $\eta_{p,K} := \|u - u_h\|_{p,K}$ with the L_2-norm and the energy norm.

The results are shown in Fig. 40. For comparison, Fig. 40(a) depicts the same subdivision as in Fig. 38(b). The subdivisions resulting from refinement based on the exact local error and the criterion Qua[0.72] are depicted in Fig. 40(b) and Fig. 40(c). The plots indicate that the error in the L_2-norm is not so sensitive towards the quickly changing derivatives of the basis functions, while the error in the energy norm is. This is correctly detected by the a posteriori error estimator.

The results using the parametrization “Pc” is shown in Fig. 38(d). The refinement with this parametrization results in the expected local refinement near the corner at $(0, 0)$ and the best approximation of the exact solution.

The ratios $R_{\text{rel}} = \|u - u_h\|/\eta$ calculated with the L_2-norm and the energy

norm for the used parametrizations and refinement criteria are plotted in Fig. 37. As in the previous example "plate with circular hoe", the numerical results indicate that the condition for the reliability of the a posteriori error estimator, namely $R_{\text{rel}} \leq C_{\text{rel}}$ for a constant C_{rel} is fulfilled. For the parametrizations "Pa" and "Pb", however, R_{rel} is increasing slightly Fig. 37(d) which indicates some difficulties when combining the error estimator with these specific parametrizations.



(a) Error in L_2-norm.

(b) Error in energy-norm.

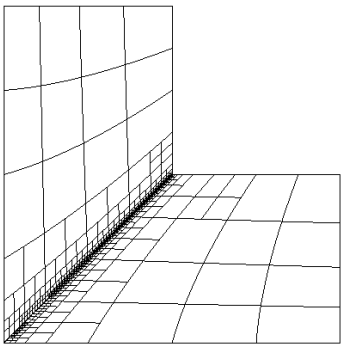
(c) R_{rel} with L_2-norm, uniform ref'.

(d) R_{rel} with energy norm, uniform ref'.

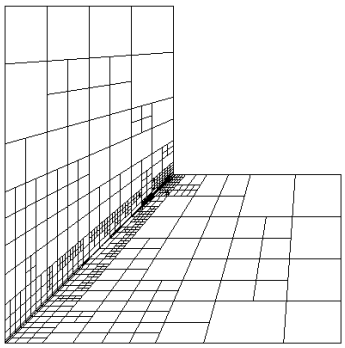
(e) R_{rel} with L_2-norm, Qua[0.72].

(f) R_{rel} with energy norm, Qua[0.72].

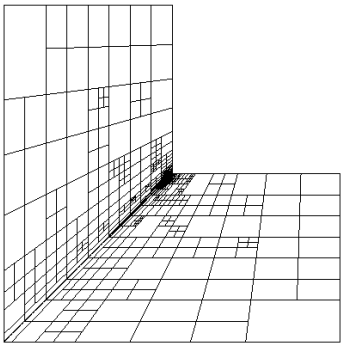
Figure 37: Errors and R_{rel} for "L-shaped domain".



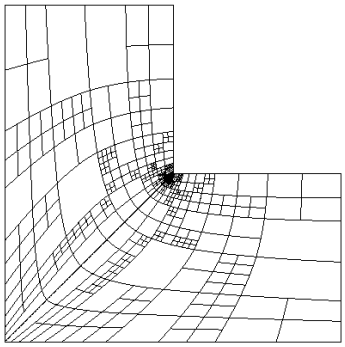
(a) “K”, Qua[0.72], after 5 ref’s: 2380 DoF.



(b) “Pa”, Qua[0.72], after 5 ref’s: 2336 DoF.

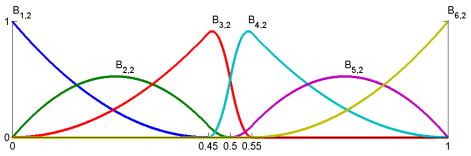


(c) “Pb”, Qua[0.72], after 5 ref’s: 2404 DoF.

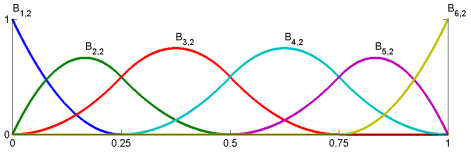


(d) “Pc”, Qua[0.72], after 5 ref’s: 2320 DoF.

Figure 38: Refined subdivisions of L-shaped domain. Criterion Qua[0.72].

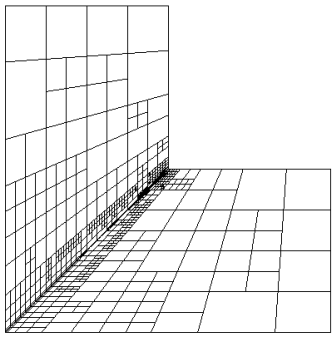


(a) "Pa": $\eta = (0, 0, 0, 0.45, 0.5, 0.55, 1, 1, 1)$

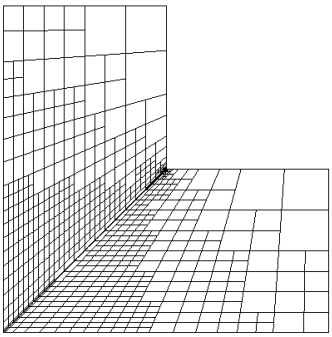


(b) "Pb": $\eta = (0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1)$

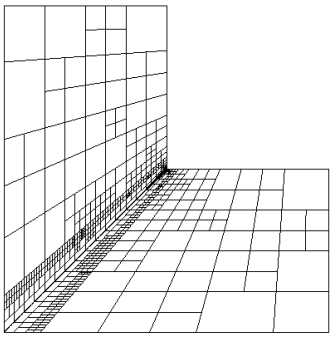
Figure 39: Basis functions of degree 2 resulting from the knot vector η for parametrizations "Pa" and "Pb".



(a) "Pa", a posteriori error estimator η_K with Qua[0.72], after 5 ref's.



(b) "Pa", after 5 ref's based on true local error in L_2-norm and Qua[0.72].



(c) "Pa", after 5 ref's based on true local error in energy norm and Qua[0.72].

Figure 40: Refined subdivisions of "L-shaped domain" based on true local error.

7.3 Advection-Diffusion Problems

We solve an advection-diffusion equation in three settings. In each case, the boundary conditions and advection velocity are chosen such that sharp interior layers and boundary layers appear. The setting that will be referred to as "Case 1" is used in [6] to illustrate an unwanted, consecutive insertion of additional knots in the process of refining T-meshes, which leads to almost global refinement.

7.3.1 Problem Settings and Geometry Mappings

We solve the advection-diffusion equation

$$\begin{aligned} -\kappa\Delta u + b\cdot\nabla u &= 0 \\ \text{with}\quad \kappa &= 10^{-6} \\ b(x_1,x_2) &= \begin{cases} (\cos\theta,\sin\theta)^T,\ \theta = 45^\circ & \text{in Case 1} \\ (x_2,-x_1)^T & \text{in Case 2} \\ (\cos\theta,\sin\theta)^T,\ \theta = 15^\circ & \text{in Case 3} \end{cases} \end{aligned}$$

In Case 1, the physical domain is the unit square, i.e. $\Omega = [0,1]^2$ and $G = Id$.

In Case 2 and Case 3, we solve the equation on the curved domain illustrated in Fig. 42(a) and Fig. 43(a). In polar coordinates, this domain is given by $\Omega = \{(r,\varphi) \in [1,2]\times[0,\pi/2]\}$; the complete geometry data is given in Appendix B.7.

The Peclet number Pe is defined by $Pe = L\cdot|b|/\kappa$, where L is the side length of the domain. If $Pe > 1$, the advection dominates the diffusion, which is clearly the case in our example with $Pe \approx 10^6$ [2]. This requires the use of a stabilization method, and as in [6, 8], the Streamline Upwinding Petrov Galerkin (SUPG) stabilization method is used [12]. The test functions $\varphi_\alpha^{(i)}$ are replaced by $\varphi_\alpha^{(i)} + \tau(b\cdot\nabla_x\varphi_\alpha^{(i)})$ with the parameter τ set to

$$\tau(K) := \frac{h_b(K)}{2\cdot|b|},$$

where $h_b(K)$ is the length of the cell K in direction of the flow b.

We set the following boundary conditions:

Case 1 (see Fig. 41(a)):

$$\begin{aligned} \Gamma_D &= \partial\Omega \\ g_D(x_1,x_2) &= \begin{cases} 0, & \text{if } (x_1 = 1)\vee(x_2 = 1)\vee(x_1 = 0\wedge x_2 > 0.2) \\ 1, & \text{else} \end{cases} \end{aligned}$$

Case 2 (see Fig. 42(a)):

$$\begin{aligned} \Gamma_D &= \partial\Omega \backslash [1,2] \times \{0\} \\ g_D(x_1, x_2) &= \begin{cases} 1, & \text{if } (x_1 = 0) \wedge (1.5 < x_2 < 1.75) \\ 0, & \text{else} \end{cases} \\ \Gamma_N &= [1,2] \times \{0\} \\ g_N(x_1, x_2) &= 0 \end{aligned}$$

Case 3 (see Fig. 43(a)):

$$\begin{aligned} \Gamma_D &= \partial\Omega \\ g_D(x_1, x_2) &= \begin{cases} 1, & \text{if } (x_1 < 0.8) \wedge (x_2 < 1.2) \\ 0, & \text{else} \end{cases} \end{aligned}$$

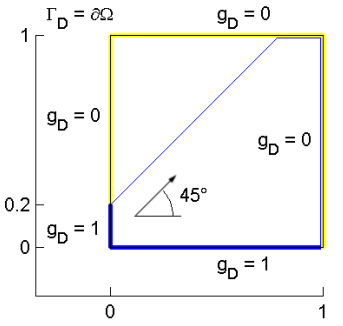
The discontinuous boundary conditions and the strong advection result in sharp layers that are indicated by thin blue lines in Fig. 41(a), Fig. 42(a), and Fig. 43(a)

Since there is no exact solution, we only test whether the error estimator is able to detect the constructed sharp layers and whether the locally refined numerical solution provides a better approximation of the discontinuities.

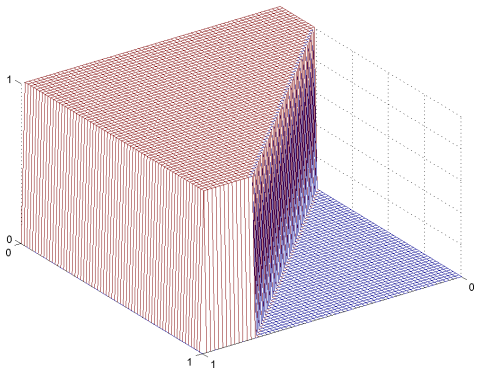
7.3.2 Numerical Results

The estimated positions of the layers and the expected solutions together with the refined subdivisions and the calculated numerical solutions are illustrated in Fig. 41, Fig. 42, and Fig. 43 for Cases 1, 2, and 3, respectively.

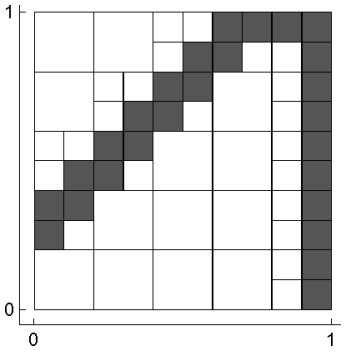
Clearly, the sharp layers are correctly detected in all three cases. Also, the refinement is successfully restriced to small areas around those sharp layers without unwanted refinement of other areas.



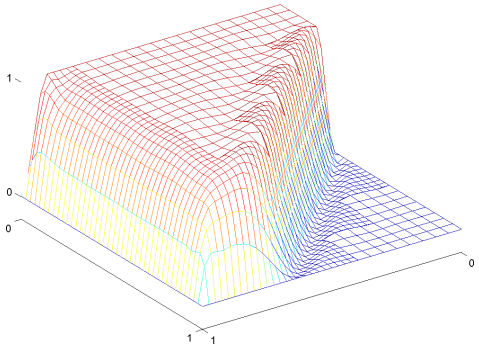
(a) Boundary conditions, estimated position of sharp layers.



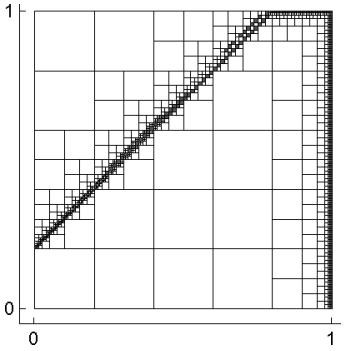
(b) Illustration of expected solution.



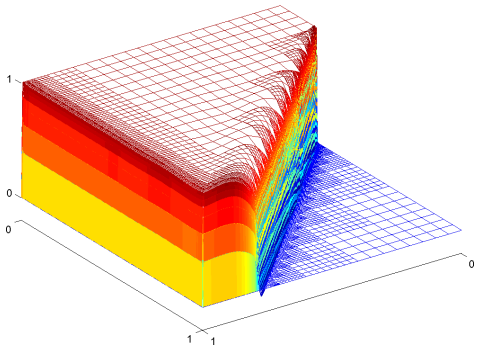
(c) After 1 ref': 280 DoF.



(d) Solution after 1 ref'.

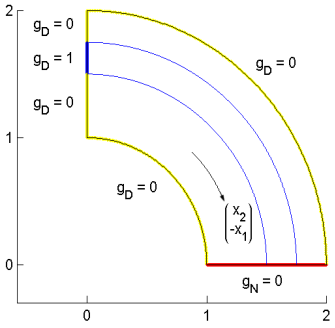


(e) After 5 ref's: 3768 DoF.

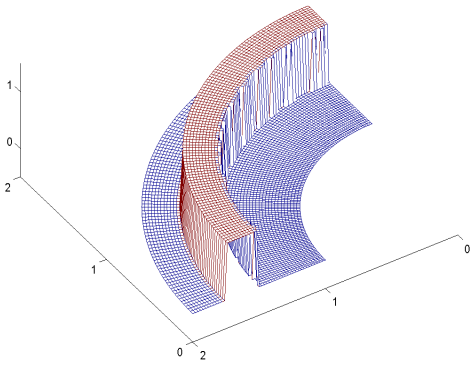


(f) Solution after 5 ref's.

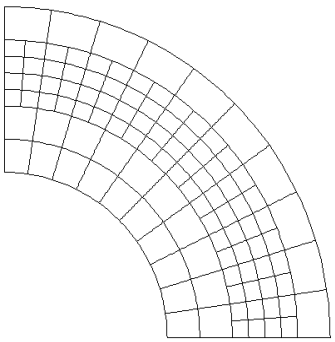
Figure 41: Subdivisions and numerical solutions after several refinement steps. Criterion Val[0.20], cells marked for refinement shaded gray.



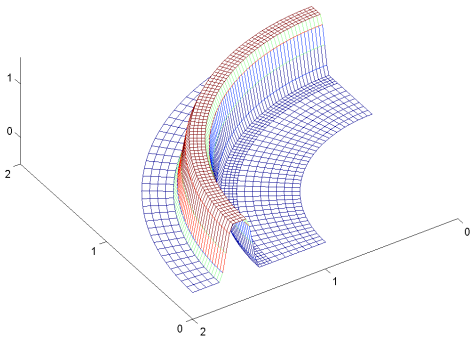
(a) Boundary conditions, estimated position of sharp layers.



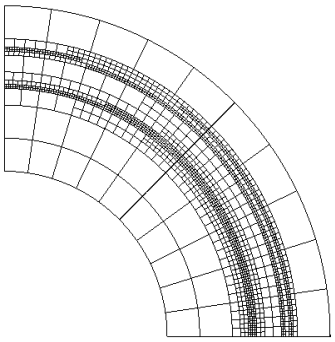
(b) Illustration of expected solution.



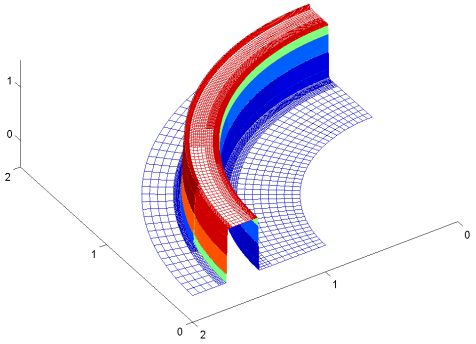
(c) After 1 ref': 472 DoF.



(d) Solution after 1 ref'.

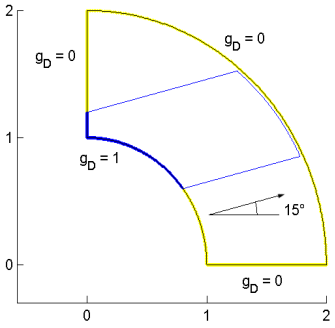


(e) After 4 ref's: 5160 DoF.

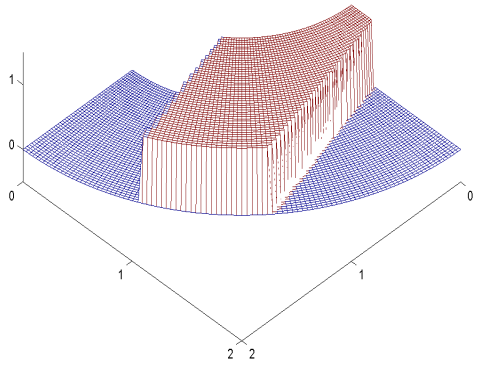


(f) Solution after 4 ref's.

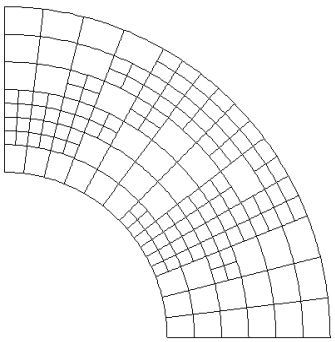
Figure 42: Subdivisions and numerical solutions after several refinement steps. Criterion Val[0.20].



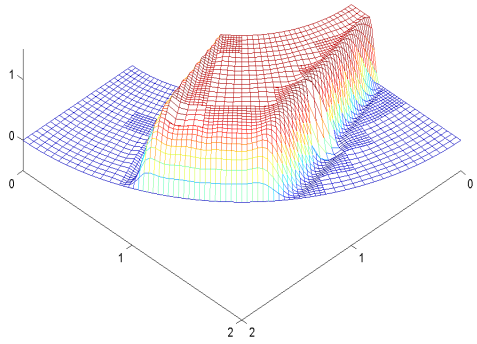
(a) Boundary conditions, estimated position of sharp layers.



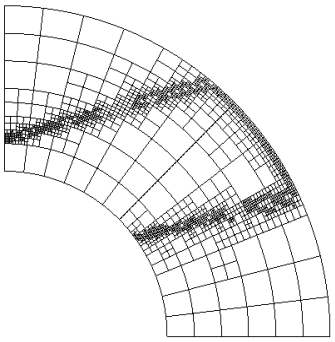
(b) Illustration of expected solution.



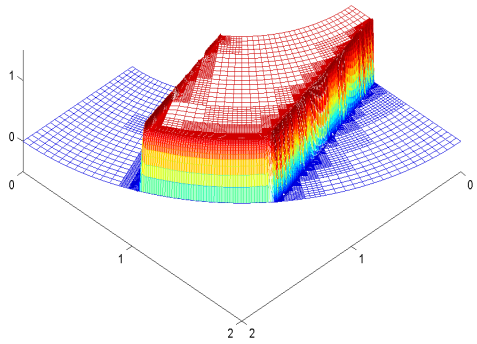
(c) After 1 ref': 656 DoF.



(d) Solution after 1 ref'.



(e) After 4 ref's: 5760 DoF.



(f) Solution after 4 ref's.

Figure 43: Subdivisions and numerical solutions after several refinement steps. Criterion Qua[0.60].

8 Summary

We have defined and described hierarchical subdivisions of the parameter domain with rectangular cells that may contain hanging nodes. The globally C^1-continuous Bogner-Fox-Schmit rectangle was successfully extended to such subdivisions with hanging nodes. As indicated by the computed numerical examples in Section 7.3, stabilization of advection-dominated advection-diffusion problems using the SUPG method can be applied to the used finite elements.

While NURBS geometry mappings preserve the exact geometry of the physical domain, several aspects have to be considered when combining these mappings with the used finite element basis functions. The examples in Sections 7.1 and 7.2 illustrate how even simple physical domains can be represented by different parametrizations, and how these parametrizations affect the distortedness of cells in the physical domain and the accuracy of the discrete solution. In combination with the implemented a posteriori error estimator, geometry mapping strongly influence the adaptive refinement process. Especially non-C^1-continuous parametrizations can lead to the refinement of areas where inaccuracies are caused by the geometry mapping rather than the discrete solution itself.

While these problems can be reduced or overcome by choosing different parametrizations as illustrated in Section 7.2, the question is how to find a general approach. As mentioned in the introduction to this theses, the goal is to work directly with geometry representations from CAD programmes. The necessity of automatically or even manually re-parametrizing given geometries is the opposite of what we are aiming for.

As mentioned in Section 5.3.2, possible ways of dealing with non-C^1-continuous geometry mappings include combinations of the finite element basis functions and NURBS basis functions, additional degrees of freedom or the definition of internal interfaces. However, these approaches raise the questions of linear independence of the basis functions and, again, of how to realize them in an automatic and efficient way.

A Appendix - Remarks re Bilinear Form

Energy Norm

For a bounded and coercive bilinear form $a(\cdot,\cdot) : V \times V \to \mathbb{R}$, the *energy norm* is defined as

$$\|v\|_E \quad := \quad \sqrt{a(v,v)}.$$

The following two properties follow immediately from the bilinearity and the coerciveness for all $v \in V$ and all $\lambda \in \mathbb{R}$:

- Positive homogenity: $\|\lambda v\|_E = \sqrt{\lambda^2 a(v,v)} = |\lambda| \|v\|_E$.
- Positive definiteness: $\|v\|_E = 0 \Leftrightarrow v \equiv 0$.

Thus, it remains to show the triangle inequality. Obviously,

$$0 \;\leq\; \|u - \lambda \cdot v\|_E^2 \quad = \quad a(u - \lambda \cdot v, u - \lambda \cdot v), \quad u, v \in V, \lambda \in \mathbb{R}.$$

Setting $\lambda := \frac{a(u,v)+a(v,u)}{2a(v,v)}$, we arrive at

$$\begin{aligned} 0 \quad &\leq \quad a(u,u) \cdot a(v,v) - \Big(\frac{a(u,v) + a(v,u)}{2}\Big)^2 \\ \frac{a(u,v) + a(v,u)}{2} \quad &\leq \quad \|u\|_E \cdot \|v\|_E. \end{aligned} \tag{55}$$

- Triangle inequality:

$$\begin{aligned} \|u+v\|_E \quad &= \quad \Big(a(u,u) + 2 \underbrace{\frac{a(u,v) + a(v,u)}{2}}_{\leq \|u\|_E \|v\|_E} + a(v,v)\Big)^{\frac{1}{2}} \\ &\leq \quad \sqrt{(\|u\|_E + \|v\|_E)^2} \;=\; \|u\|_E + \|v\|_E. \end{aligned}$$

Hence, $\|\cdot\|_E$ fulfills the properties of a norm on V. If $a(\cdot,\cdot)$ is symmetric, it defines an inner product

$$(v,u)_A \quad := \quad a(u,v)$$

and (55) is the associated Cauchy-Schwarz inequality

$$|a(u,v)| \;=\; |(u,v)_A| \;\leq\; \|u\|_E \|v\|_E.$$

Boundedness and Coercivity

As defined in Section 6.2.2, we denote the restrictions of the bilinear form $a(\cdot,\cdot)$ and of a norm $\|\cdot\|_p$ to the image $G(K)$ of a cell $K \in \mathcal{K}$ with $a(\cdot,\cdot)_K$ and $\|\cdot\|_{p,K}$, respectively.

Boundedness in the Symmetric Case, $b \equiv 0$

If $b \equiv 0$, then $a(\cdot,\cdot)$ is symmetric. Assume that the coefficients A and c are bounded, i.e. $A, c \in L_\infty$, then

$$\begin{aligned} |a(u,v)| &\leq \|A\|_\infty |u|_1 |v|_1 + \|c\|_\infty \|u\|_0 \|v\|_0 \\ &\leq \max(\|A\|_\infty, \|c\|_\infty) \cdot (|u|_1|v|_1 + \|u\|_0\|v\|_0) \\ &\leq \max(\|A\|_\infty, \|c\|_\infty) \cdot \|u\|_1 \|v\|_1. \end{aligned} \tag{56}$$

As discussed above, a symmetric bilinear form $a(\cdot,\cdot)$ is bounded in the energy norm:

$$|a(u,v)| \leq \|u\|_E \|v\|_E. \tag{57}$$

Since (57) is independent from the domain over which $a(\cdot,\cdot)$ is defined, we can conclude that

$$|a(u,v)_K| \leq \|u\|_{E,K} \|v\|_{E,K}$$

holds for all $K \in \mathcal{K}$.

Boundedness in the Nonsymmetric Case, $b \not\equiv 0$

Assume that the coefficients A, b and c are bounded, i.e. $A, b, c \in L_\infty$. Then

$$\begin{aligned} \int_\Omega (b^T \nabla u) v \, dx &= \int_\Omega (b_1 \partial_1 u + b_2 \partial_2 u) v \, dx \\ &\leq \|b\|_\infty \|v\|_0 \left(\sqrt{\int_\Omega (\partial_1 u)^2 \, dx} + \sqrt{\int_\Omega (\partial_2 u)^2 \, dx} \right) \\ &\leq 2\|b\|_\infty \|v\|_0 |u|_1 \leq 2\|b\|_\infty \|v\|_1 \|u\|_1. \end{aligned}$$

Together with (56), we obtain the boundedness of $a(\cdot,\cdot)$:

$$|a(u,v)| \leq \Big(\underbrace{\max(\|A\|_\infty, \|c\|_\infty) + 2\|b\|_\infty}_{=:\mu_2} \Big) \cdot \|u\|_1 \|v\|_1.$$

For the restriction to a cell $K \in \mathcal{K}$, we obtain

$$|a(u,v)_K| \leq \mu_{2,K} \cdot \|u\|_{1,K} \|v\|_{1,K}$$

where

$$\mu_{2,K} := \max(\|A\|_{\infty,K}, \|c\|_{\infty,K}) + 2\|b\|_{\infty,K}$$

Since $G(K) \subseteq \Omega$ implies $\|f\|_{\infty,K} \leq \|f\|_\infty$ for any function $f \in L_\infty$, we have $\mu_{2,K} \leq \mu_2$ for all $K \in \mathcal{K}$ and

$$a(u,v)_K \leq \mu_2 \cdot \|u\|_{1,K} \|v\|_{1,K},$$

i.e. the restriction of $a(\cdot,\cdot)$ to a cell K is bounded with a constant μ_2 that does not depend on the specific cell K.

Coercivity in the Symmetric Case, $b \equiv 0$

If the following holds for the coefficients of $a(\cdot,\cdot)$:

$$A_1, A_2 \;\geq\; A_0 \;>\; 0, \quad b \;\equiv\; 0, \quad c \;\geq\; c_0 \;>\; 0 \quad \text{on } \Omega,$$

the coercivity of $a(\cdot,\cdot)$ immediately follows:

$$a(v,v) \;\geq\; A_0|v|_1 + c_0\|v\|_0 \;\geq\; \min(A_0, c_0)\cdot\|v\|_1.$$

If we only know that $c \geq 0$ in Ω, then

$$a(v,v) \;\geq\; A_0|v|_1.$$

In this case, coercivity follows from the Friedrich's inequality, if $\Gamma_D \neq \emptyset$ [15].

Coercivity in the Nonsymmetric Case, $b \not\equiv 0$

We can re-write $a(u,v)$ as follows:

$$\begin{aligned} a(u,v) \;\;=\;\; & \int_\Omega (\nabla v)^T A \nabla u \; dx + \frac{1}{2}\int_\Omega \underbrace{(b^T\nabla u)v - (b^T\nabla v)u}_{=0,\ \text{for } u=v} \; dx \\ & + \int_\Omega \left(-\frac{1}{2}(\nabla\cdot b) + c\right) uv \; dx + \frac{1}{2}\underbrace{\int_\Omega \nabla\cdot(buv)\; dx}_{=\int_{\partial\Omega}(buv)\cdot n \; dx}, \end{aligned}$$

In the example in Section 7.3, we have

$$A_1, A_2 \geq 10^{-6} > 0, \quad \nabla\cdot b \equiv 0, \quad c \equiv 0 \quad \text{in } \Omega.$$

For $u = v$ and $v \in H_0^1(\Omega)$, the integral $\int_{\partial\Omega}(bv^2)\cdot n \; dx$ vanishes on Γ_D. In Case 1 and Case 3 of the example in Section 7.3, we have $\Gamma_D = \partial\Omega$. In Case 2, where $\Gamma_N \neq \emptyset$, we have $n(x_1,x_2) = (0,-1)^T$ and $b(x_1,x_2) = (0,-x_1)^T$ on Γ_N. Hence, $\int_{\partial\Omega}(bv^2)\cdot n \; dx = \int_{\Gamma_N} x_1 v^2 \; dx \geq 0$. Since $\Gamma_D \neq \emptyset$, it follows from the Friedrich's inequality that, in all three cases, the bilinear form $a(\cdot,\cdot)$ is coercive on $H_0^1(\Omega)$ [13].

B Appendix - Geometry Data

B.1 Unit Circle

Data for the unit circle in Fig. 22 [8]:

Degree: $p = 2$
Knot vector: $\eta = (0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4)$

Coordinates and weights of control points:

i	P_i	w_i
1	$(1, 0)$	1
2	$(1, 1)$	$\frac{1}{\sqrt{2}}$
3	$(0, 1)$	1
4	$(-1, 1)$	$\frac{1}{\sqrt{2}}$
5	$(-1, 0)$	1

i	P_i	w_i
6	$(-1, -1)$	$\frac{1}{\sqrt{2}}$
7	$(0, -1)$	1
8	$(1, -1)$	$\frac{1}{\sqrt{2}}$
9	$(1, 0)$	1

B.2 Plate with Circular Hole, Double Knot

Data for the geometry "plate with circular hole", Fig. 24(c), using the double interior knot as in [6]:

Degrees: $p = p^* = 2$
Knot vectors: $\eta = (0, 0, 0, 0.5, 0.5, 1, 1, 1)$
$\eta^* = (0, 0, 0, 1, 1, 1)$

Coordinates and weights of control points:

i	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$	$w_{i,1}$	$w_{i,2}$	$w_{i,3}$
1	$(-1, 0)$	$(-2.5, 0)$	$(-4, 0)$	1	1	1
2	$(-1, \sqrt{2} - 1)$	$(-2.5, 0.75)$	$(-4, 2)$	$(1 + \frac{1}{\sqrt{2}})/2$	1	1
3	$(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$	$(-1.5, 1.5)$	$(-4, 4)$	$(1 + \frac{1}{\sqrt{2}})/2$	1	1
4	$(1 - \sqrt{2}, 1)$	$(-0.75, 2.5)$	$(-2, 4)$	$(1 + \frac{1}{\sqrt{2}})/2$	1	1
5	$(0, 1)$	$(0, 2.5)$	$(0, 4)$	1	1	1

B.3 Plate with Circular Hole, Double Control Point

Data for the geometry "plate with hole", Fig. 24(d), using a double control point as in [8]:

Degrees: $p = p^* = 2$
Knot vectors: $\eta = (0, 0, 0, 0.5, 1, 1, 1)$
$\eta^* = (0, 0, 0, 1, 1, 1)$

Coordinates of control points:

i	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$	$w_{i,1}$	$w_{i,2}$	$w_{i,3}$
1	$(-1, 0)$	$(-2.5, 0)$	$(-4, 0)$	1	1	1
2	$(-1, \sqrt{2} - 1)$	$(-2.5, 0.75)$	$(-4, 4)$	$(1 + \frac{1}{\sqrt{2}})/2$	1	1
3	$(1 - \sqrt{2}, 1)$	$(-0.75, 2.5)$	$(-4, 4)$	$(1 + \frac{1}{\sqrt{2}})/2$	1	1
4	$(0, 1)$	$(0, 2.5)$	$(0, 4)$	1	1	1

B.4 L-Shaped Domain, Double Knot "K"

Data for the parametrization "K" of the geometry "L-shaped domain" as used in Section 7.2 and as in [6].

Degrees: $p = p^* = 2$
Knot vectors: $\eta = (0, 0, 0, 0.5, 0.5, 1, 1, 1)$
$\eta^* = (0, 0, 0, 1, 1, 1)$
Weights: $w_{i,j} = 1$ for all i, j

Coordinates of control points:

i	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$
1	$(-1, 1)$	$(-0.6, 1)$	$(0, 1)$
2	$(-1, 0)$	$(-0.55, 0)$	$(0, 0.5)$
3	$(-1, -1)$	$(-0.5, -0.5)$	$(0, 0)$
4	$(0, -1)$	$(0, -0.55)$	$(0.5, 0)$
5	$(1, -1)$	$(1, -0.6)$	$(1, 0)$

B.5 L-Shaped Domain, "Pa" and "Pb"

Data for the parametrizations "Pa" and "Pb" of the geometry "L-shaped domain" as used in Section 7.2.

Degrees: $p = p^* = 2$
Knot vectors: $\eta = (0, 0, 0, 0.45, 0.5, 0.55, 1, 1, 1)$ for "Pa"
$\eta = (0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1)$ for "Pb"
$\eta^* = (0, 0, 0, 1, 1, 1)$
Weights: $w_{i,j} = 1$ for all i, j

Coordinates of control points:

i	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$
1	$(-1, 1)$	$(-0.5, 1)$	$(0, 1)$
2	$(-1, 0.5)$	$(-0.5, 0.5)$	$(0, 0.5)$
3	$(-1, -1)$	$(-0.5, -0.5)$	$(0, 0)$
4	$(-1, -1)$	$(-0.5, -0.5)$	$(0, 0)$
5	$(0.5, -1)$	$(0.5, -0.5)$	$(0.5, 0)$
6	$(1, -1)$	$(1, -0.5)$	$(1, 0)$

B.6 L-Shaped Domain, "Pc"

Data for the parametrization "Pc" of the geometry "L-shaped domain" as used in Section 7.2.

Degrees: $p = p^* = 2$
Knot vectors: $\eta = (0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1)$
$\eta^* = (0, 0, 0, 0.5, 1, 1, 1)$
Weights: $w_{i,j} = 1$ for all i, j

Coordinates of control points:

i	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$	$P_{i,4}$
1	$(-1, 1)$	$(-0.75, 1)$	$(-0.3, 1)$	$(0, 1)$
2	$(-1, 0.2)$	$(-0.7, 0.35)$	$(-0.3, 0.5)$	$(0, 0.5)$
3	$(-1, -1)$	$(-0.6, -0.3)$	$(-0.2, -0.05)$	$(0, 0)$
4	$(-1, -1)$	$(-0.3, -0.6)$	$(-0.05, -0.2)$	$(0, 0)$
5	$(0.2, -1)$	$(0.35, -0.7)$	$(0.5, -0.3)$	$(0.5, 0)$
6	$(1, -1)$	$(1, -0.75)$	$(1, -0.3)$	$(1, 0)$

B.7 Curved Domain

Data for the curved domain in Cases 2 and 3 in Section 7.3.

Degrees: $p = p^* = 2$
Knot vectors: $\eta = (0, 0, 0, 1, 1, 1)$
$\eta^* = (0, 0, 0, 0.5, 1, 1, 1)$

Coordinates of control points:

i	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$	$P_{i,4}$
1	$(1, 0)$	$(1, \sqrt{2}-1)$	$(\sqrt{2}-1, 1)$	$(0, 1)$
2	$(1.5, 0)$	$(1.5, 1.5(\sqrt{2}-1))$	$(1.5(\sqrt{2}-1), 1.5)$	$(0, 1.5)$
3	$(2, 0)$	$(2, 2(\sqrt{2}-1))$	$(2(\sqrt{2}-1), 2)$	$(0, 2)$
i	$w_{i,1}$	$w_{i,2}$	$w_{i,3}$	$w_{i,4}$
1,2,3	1	$(1+\frac{1}{\sqrt{2}})/2$	$(1+\frac{1}{\sqrt{2}})/2$	1

C Appendix - Symbols and Notation

Commonly used symbols and notation

See e.g. [4, 9, 14, 15].

∇u	Gradient of u; in 2d: $\nabla u = (\partial u/\partial x_1, \partial u/\partial x_2)^T$. Also denoted grad u.
$\nabla \cdot u$	Divergence of u; in 2d: $\nabla \cdot u = \partial u/\partial x_1 + \partial u/\partial x_2$. Also denoted div u.
Δu	Laplace-Operator applied to u; in 2d: $\Delta u = \partial^2 u/\partial x_1^2 + \partial^2 u/\partial x_2^2$.
Ω	Physical domain.
Γ_D	Dirichlet-boundary.
Γ_N	Neumann-boundary.
$\overline{A}$	Closure of a set A.
∂A	Boundary of a set A.
$C(\Omega)$	Space of functions that are continuous over Ω.
$C^1(\Omega)$	Space of functions that are continuously differentiable over Ω.
$L_2(\Omega)$	Space of square-integrable functions over Ω.
$H^1(\Omega)$	Space of square-integrable functions with square-integrable first weak derivatives over Ω.
$H_0^1(\Omega)$	Space of functions in $H^1(\Omega)$ that are zero on Γ_D.
$\|\cdot\|_0$	L_2-norm: $\|u\|_0^2 = \int_\Omega \|u\|^2 \, dx$.
$\|\cdot\|_1$	H^1-seminorm: $\|u\|_1^2 = \int_\Omega \|\nabla_x u\|^2 \, dx$.
$\|\cdot\|_1$	H^1-norm: $\|u\|_1^2 = \|u\|_0^2 + \|u\|_1^2$.
$\|\cdot\|_\infty$	L_∞-norm: $\|u\|_\infty = \sup_\Omega u$.
$\|\cdot\|_E$	Energy-norm: $\|u\|_E^2 = a(u,u)$.
$\|\cdot\|_{p,K}$	Restriction of the norm $\|\cdot\|_p$ to $G(K)$ (e.g. $\|u\|_{0,K}^2 = \int_{G(K)} \|u\|^2 \, dx$).
$\|\cdot\|_{l_2}$	l_2-norm: $\|v\|_{l_2}^2 = \sum_{i=1}^{\|v\|} v_i^2$.
supp	Support of a function: supp $u = \overline{\{x : u(x) \neq 0\}}$.
$\|\cdot\|$	Absolute value of a scalar or cardinality of a vector, depending on argument.

Symbols and notation specific to this thesis

Abbreviations for derivatives:

$\partial_0 v = v$
$\partial_1 v = \partial v/\partial \xi_1$
$\partial_2 v = \partial v/\partial \xi_2$
$\partial_3 v = \partial^2 v/\partial \xi_1 \partial \xi_2$

Subdivision and Domains:

Q	Parameter domain; $Q = [0,1]^2$ unless specified otherwise.
G	Geometry mapping; $\Omega = G(Q)$.
ξ, ξ_1, ξ_2	Coordinates in the parameter domain.
x, x_1, x_2	Coordinates in the physical domain.
$\mathcal{K}$	Subdivision of Q.

K	A cell of the subdivision.
$\widehat{K}$	Reference element; $\widehat{K} = [0,1]^2$.
$\zeta^{(i)}$	Node of the subdivision in the parameter domain.
$x^{(i)}$	Node of the subdivision in the physical domain.
$\mathcal{K}^{(i)}$	Set of cells $K \in \mathcal{K}$ that contain $\zeta^{(i)}$ in their closure.
$E^{(p,q)}$	Edge from node $\zeta^{(p)}$ to $\zeta^{(q)}$.
N_E	Total number of cells.
N_X	Total number of nodes.
$\mathsf{Anc}(\cdot)$	Ancestors of the node $\zeta^{(i)}$ or a cell K (see Sec. 4.4).

Index Sets:

I_α	Set of indices $I_\alpha = \{0,1,2,3\}$.
$\mathcal{I}_P$	Set of indices of all primary nodes.
$\mathcal{I}_H$	Set of indices of all hanging nodes.
$\mathcal{I}_X$	Set of indices of all nodes, $\mathcal{I}_X = \mathcal{I}_P \cup \mathcal{I}_H$.
$\mathcal{I}_K$	Set of indices of the vertices of a cell K.

Special function sets (see Section 3.2):

$\mathcal{Q}_3(K)$	Set of bi-cubic polynomials over K.
$\mathcal{Q}_3(\mathcal{K})$	Set of piecewise bi-cubic functions over $\mathcal{K}$.
$\mathcal{Q}_3^1(\mathcal{K})$	Set of piecewise bi-cubic and globally C^1-continuous functions over $\mathcal{K}$.

Nabla-Operator in parameter and physical domain:

∇_x	Nabla-Operator in physical domain; in 2d: $\nabla_x = (\partial/\partial x_1, \partial/\partial x_2)^T$.
∇_ξ	Nabla-Operator in parameter domain; in 2d: $\nabla_\xi = (\partial/\partial \xi_1, \partial/\partial \xi_2)^T$.

Finite element spaces and basis functions:

V_h	Finite element space over parameter domain Ω.
$\varphi_\alpha^{(i)}$	Basis functions of V_h.
X_h	Finite element space over parameter domain Q.
$\widehat{\varphi}_\alpha^{(i)}$	Basis functions of X_h.
$\widehat{p}_\alpha^{(i)}$	Basis functions of $\mathcal{Q}_3(\widehat{K})$.
Ψ_K	Local basis of $\mathcal{Q}_3(K)$.
$\widehat{\psi}_\alpha^{(i)}$	Local basis functions of $\mathcal{Q}_3(K)$ of a specific cell $K \in \mathcal{K}$.

Other abbreviations:

DoF	Degrees of freedom.
ref'	Refinement.

List of Figures

References

[1] R. E. Bank, R. K. Smith. *A Posteriori Error Estimates Based on Hierarchical Bases.* SIAM Journal on Numerical Analysis, Vol. 30, No. 4: 921-935, 1993.

[2] W. J. Beek, K. M. K. Muttzall, J. W. Van Heuven. *Transport Phenomena.* John Wiley & Sons, Ltd, West Sussex, England, 1999.

[3] C. Carstensen. *Some remarks on the history and future of averaging technique in a posteriori finite element error analysis.* ZAMM 84, No. 1: 3-21, 2004.

[4] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland Publishing Co., Amsterdam, 1978.

[5] J. Deng, F. Chen, Y. Feng. *Dimensions of spline spaces over T-meshes.* Journal of Computational and Applied Mathematics, 194: 267-283, 2006.

[6] M. R. Dörfel, B. Jüttler, B. Simeon. *Adaptive isogeometric analysis by local h-refinement with T-splines.* Computer Methods in Applied Mechanics and Engineering, 199: 264-275, 2010.

[7] H. W. Engl, A. Neubauer. *Skriptum Analysis.* Industrial Mathematics Institute, Johannes Kepler University Linz, WS 2001.

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

[9] M. Jung, U. Langer. *Methode der Finiten Elemente für Ingenieure.* B. G. Teubner GmbH, Stuttgart, Leipzig, Wiesbaden, 2001.

[10] B. Jüttler. *Skriptum zur Vorlesung Computer Aided Geometric Design.* Institute of Applied Geometry, Johannes Kepler University Linz, SS 2004.

[11] L. Piegl, W. Tiller. *The NURBS Book.* Springer, Berlin, Heidelber, New York, 1997.

[12] O. Pironneau. *Finite Element Methods for Fluids.* Masson, Paris, 1989.

[13] A. Quarteroni, A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford University Press Inc., New York, 1999.

[14] W. Zulehner. *Lecture Notes for the Course Numerical Methods for Continuum Mechanics 1.* Institute of Computational Mathematics, Johannes Kepler University Linz, SS 2009.

[15] W. Zulehner. *Skriptum zur Vorlesung Numerik Partieller Differentialgleichungen.* Institute of Computational Mathematics, Johannes Kepler University Linz, WS 2003/2004.

Eidesstattliche Erklärung

Ich, Stefan Ken Kleiss, erkläre an Eides statt, dass ich die vorliegende Masterarbeit 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.

Linz, am 24. Juni 2010, _____

Stefan Ken Kleiss