Numerical Analysis of Isogeometric Methods

Stefan Takacs

Winter semester 2018/19

Contents

Chapter 1

What is Isogeometric Analysis?

1.1 Isogeometric Analysis

First of all, Isogeometric Analysis (IgA) is a method to approximate the solution of a partial differential equation (PDE), like the finite element method (FEM).

IgA was originally proposed in the year 2005 by T. Hughes, J. A. Cottrell and Y. Bazilevs in the paper [1] and has gained much interest since then. Due to Elsevier, this paper has been cited 2151 times (as of 1 Oct 2018)[1].

IgA is an abbreviation for **I**so**g**eometric **A**nalysis:

- **Iso** *(Greek)* means equal. This indicates that the same basis functions are used for geometry transformation and as ansatz functions, i.e., that we use an *isoparametric* discretization. We will see later that this requirement can be dropped.
- **Geometric** *(Greek)* means originally measuring the earth. It indicates the original design goal of IgA to bring together both the world of computer aided design (CAD) and the world of finite element simulation.
- **Analysis** means that the method was invented by engineers. They also say Finite Element Analysis (FEA) instead of Finite Element Method (FEM). So, this means that IgA is a method for the numerical solution of PDEs. Depending of the viewpoint, IgA is a generalization of FEM or a variant of FEM.

The main aims of IgA are:

1. Better connection between computer aided design (CAD) and finite element (FEM) simulation. The classical approach in FEM requires meshing the domain. IgA tries to avoid this. (We will come to this part later). In many cases, we can represent the geometry exactly in IgA (like circles), while in FEM the domain is often only approximated and refinement is also done to get a better representation of the geometry.
2. IgA is a high-order discretization with fewer degrees of freedom than classical high-order FEM. As the solutions of PDEs are often smooth (at least in the interior of the domain), it is appropriate to choose smooth ansatz functions.

[1] see `https://www.sciencedirect.com/science/article/pii/S0045782504005171`

Consider in 1D a domain $\Omega = (0,1)$, which is subdivided into n subintervals. For standard low-order FEM, we use globally C^0, piecewise linear functions. This function space has $n+1$ degrees of freedom, cf. Fig. 1.1. For standard high-order FEM, we use globally C^0, piecewise polynomials of degree p. This function space has $np+1$ degrees of freedom, cf. Fig. 1.4. For standard IgA, we use globally C^{p-1}, piecewise polynomials of degree p. This function space has $n+p$ degrees of freedom, cf. Fig. 1.2.

1.2 Univariate Splines and NURBS

1.2.1 Spline functions

Let p and n be positive integers. We call

$$\Xi := (\xi_1, \dots, \xi_{n+p+1})$$

a *p-open knot vector* if

$$\xi_1 = \cdots = \xi_{p+1} < \xi_{p+2} \leq \cdots \leq \xi_n < \xi_{n+1} = \cdots = \xi_{n+p+1}$$

and

$$\xi_j < \xi_{j+p} \text{ for all } j \in \{2, \dots, n\}.$$

For the univariate case, we can choose without loss of generality $\xi_1 = 0$ and $\xi_{n+p+1} = 1$.

For each *knot vector*, we find the corresponding vectors of *break points* and *multiplicities* (and vice versa).

We call

$$Z = (\zeta_1, \dots, \zeta_N)$$

the vector of *breakpoints* and

$$M = (m_1, \dots, m_N)$$

the vector of *multiplicities* if

$$\Xi = (\underbrace{\zeta_1, \dots, \zeta_1}_{m_1 \text{ times}}, \underbrace{\zeta_2, \dots, \zeta_2}_{m_2 \text{ times}}, \cdots, \underbrace{\zeta_N, \dots, \zeta_N}_{m_N \text{ times}}).$$

Simple calculations yield

$$\sum_{i=1}^{N} m_i = n + p + 1 \qquad \text{and} \qquad 0 < m_i \leq p+1 \qquad \text{for all} \qquad m_i = 1, \dots, N.$$

The vector of breakpoints Z forms a partition of $\Omega = (\zeta_1, \zeta_N)$, which we also call a *mesh* consisting of the *elements* $I_i = (\zeta_i, \zeta_{i+1})$. We call

$$h_i := \zeta_{i+1} - \zeta_i$$

the *local grid size* and

$$h = \max_{i=1,\dots,N-1} h_i \tag{1.1}$$

the *global grid size*.

The *B-spline basis functions* are defined via the *Cox-de Boor* formula.

Definition 1.1. *Let Ξ be an p-open knot vector. For $q = 0$ and $i = p+1, \dots, n$, we define*

$$\widehat{B}_{i,0,\Xi}(x) = \begin{cases} 1 & \text{if } \xi_i \le x < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}.$$

For $q = 1, \dots, p$ and $i = p-q+1, \dots, n$, we define recursively

$$\widehat{B}_{i,q,\Xi}(x) = \frac{x-\xi_i}{\xi_{i+q}-\xi_i}\widehat{B}_{i,q,\Xi}(x) + \frac{\xi_{i+q+1}-x}{\xi_{i+q+1}-\xi_{i+1}}\widehat{B}_{i,q-1,\Xi}(x).$$

Note that for $q := p$, the indices are $i = 1, \dots, n$. Thus, n is the number of basis functions.

Remark 1.2. *If the denominator is 0, we can derive that also the corresponding B-spline vanishes ($\xi_{i+p} - \xi_i \Rightarrow \widehat{B}_{i,p-1} = 0$ and $\xi_{i+p+1} - \xi_{i+1} \Rightarrow \widehat{B}_{i,p-1} = 0$). So, we obtain terms $\frac{0}{0}$. We define them to be 0.*

Often, we are interested in the *equidistant case*. For $1/h \in \mathbb{N}$, we define

$$\Xi(p,k,h) := (\underbrace{0,\cdots,0}_{p+1 \text{ times}}, \underbrace{h,\cdots,h}_{p-k \text{ times}}, \underbrace{2h,\cdots,2h}_{p-k \text{ times}}, \cdots, \underbrace{(h^{-1}-1)h,\cdots,(h^{-1}-1)h}_{p-k \text{ times}}, \underbrace{1,\cdots,1}_{p+1 \text{ times}})$$

and

$$\widehat{B}_{i,p,k,h} := \widehat{B}_{i,p,\Xi(p,k,h)} \qquad \text{and} \qquad \widehat{B}_{i,p,h} := \widehat{B}_{i,p,p-1,h}.$$

The B-splines have the following properties:

- *Non negativity:*
$$\widehat{B}_{i,p,\Xi}(x) \ge 0 \qquad \text{for all } x \in \Omega. \tag{1.2}$$
- *Partition of unity:*
$$\sum_{i=1}^{n} \widehat{B}_{i,p,\Xi}(x) = 1 \qquad \text{for all } x \in \Omega. \tag{1.3}$$
- *Bounded support:*
$$\operatorname{supp} \widehat{B}_{i,p,\Xi} = [\xi_i, \xi_{i+p+1}],$$
where $\operatorname{supp} f = \overline{\{x : f(x) \neq 0\}}$ and $\overline{M}$ is the closure of M.
- The B-splines $[\widehat{B}_{i,p,\Xi}]_{i=1}^n$ *form a basis* of the space $S_{p,\Xi}(0,1)$, given as follows.

Definition 1.3. *The spline space $S_{p,\Xi}(0,1)$ is the space of all functions v such that (a)*

$$v|_{I_i} \in \mathbb{P}_p,$$

i.e., the restriction of v to an element $I_i = (\zeta_i, \zeta_{i+1})$ is in $\mathbb{P}_p$, the space of polynomials of degree (at most) p, and (b)

$$\lim_{x \to \zeta_i^+} \frac{\partial^j}{\partial x^j} v(x) = \lim_{x \to \zeta_i^-} \frac{\partial^j}{\partial x^j} v(x) \qquad \text{for all } j = 0, \dots, k_i := p - m_i,$$

i.e., the function value and the derivatives (up to order k_i) agree on the break points. We call

$$K = (k_1, \dots, k_N) = p - M$$

the vector of continuities.

Obviously, we have $k_1 = k_N = -1$ and $0 \leq k_1, \dots, k_{N-1} < p$.

As for the B-spline basis functions, we define the equidistant spline spaces:

$$S_{p,k,h}(0,1) := S_{p,\Xi(p,k,h)} \qquad \text{and} \qquad S_{p,h}(0,1) := S_{p,p-1,h}(0,1).$$

- *Bounded dependence on knots*: The definition of a B-spline $\widehat{B}_{i,p,\Xi}$ only depends on the values of the following $p+2$ knots, which form the *local knot vector*:

$$\Xi(\widehat{B}_{i,p,\Xi}) := (\xi_i, \dots, \xi_{i+p+1}).$$

For any interval $I_j = (\zeta_j, \zeta_{j+1})$, we define the *support extension*

$$\widetilde{I}_j := \operatorname{int}\Big(\bigcup_{l=1,\dots,n \,:\, \operatorname{supp} \widehat{B}_{l,p,\Xi} \cap I_j \neq \emptyset} \operatorname{supp} \widehat{B}_{l,p,\Xi} \Big), \tag{1.4}$$

where int M is the interior of M. If i is such that $I_j = (\xi_i, \xi_{i+1})$, then

$$\widetilde{I}_j = (\xi_{i-p}, \xi_{i+p+1}).$$

Lemma 1.4. *Assume that $\Xi = (\xi_1, \dots, \xi_{n+p+1})$ is a p-open knot vector and that $\Xi' = (\xi_2, \dots, \xi_{n+p})$ is a $(p-1)$-open knot vector. Then, the derivative of a spline satisfies*

$$\frac{d}{dx}\widehat{B}_{i,p,\Xi}(x) = \frac{p}{\xi_{i+p} - \xi_i}\widehat{B}_{i,p-1,\Xi'}(x) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}}\widehat{B}_{i+1,p-1,\Xi'}(x).$$

This shows $\frac{d}{dx}\widehat{B}_{i,p,\Xi} \in S_{p-1,\Xi'}(0,1)$.

The mapping

$$\frac{d}{dx} : S_{p,\Xi}(0,1) \to S_{p-1,\Xi'}(0,1)$$

is surjective. The mapping

$$\frac{d}{dx} : S_{p,\Xi}(0,1) \backslash \mathbb{R} \to S_{p-1,\Xi'}(0,1) \tag{1.5}$$

is bijective, where

$$S_{p,\Xi}(0,1) \backslash \mathbb{R} = \left\{ v \in S_{p,\Xi}(0,1) : \int_\Omega v(x)\,\mathrm{d}x = 0 \right\}.$$

Examples of B-spline bases:

- For $p = 1$, we obtain that all basis functions are interpolatory, i.e., each basis function is 1 at its center and all other basis functions are 0 there.

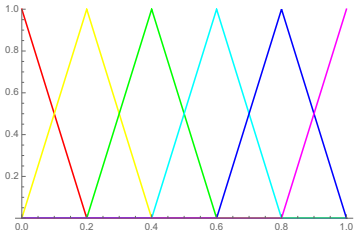


Figure 1.1: Splines of degree 1

- For $p = 2$ without repeated knots (i.e., multiplicity 1 everywhere), we obtain that only the basis functions on the boundary are interpolatory.

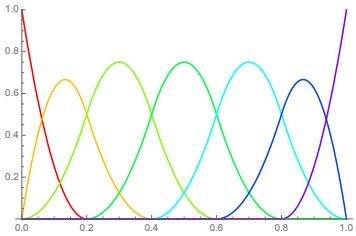


Figure 1.2: Splines of degree 2 without repeated knots

- For $p = 2$ with one repeated knot at $2/5$ (multiplicity 2), we obtain that the basis functions on the boundary and at $2/5$ are interpolatory.

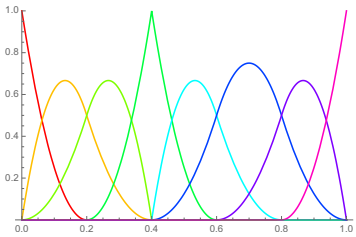


Figure 1.3: Splines of degree 2 with one repeated knot at $2/5$

- For $p = 2$ and only C^0-smoothness (which is the same as to have multiplicity 2 everywhere), we obtain for every knot one interpolatory basis function.

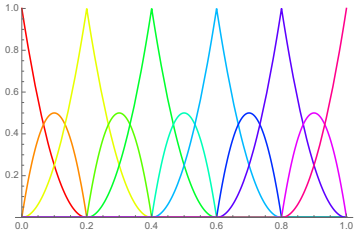


Figure 1.4: Splines of degree 2 with one repeated knots

This is a general rule: The B-splines are interpolatory on the boundary and everywhere in the interior where the multiplicity is p.

Based on the B-spline basis, we associate any function $u_h \in S_{p,\Xi}(0,1)$ with its *vector representation*:

$$\underline{u}_h = (u_1, \ldots, u_n) \qquad \text{such that} \qquad u_h(x) = \sum_{i=1}^{n} u_i \widehat{B}_{i,p,\Xi}(x).$$

1.2.2 Spline and NURBS curves

Let *control points* $C = (c_1, \ldots, c_n)$ with $c_i \in \mathbb{R}^d$ with $d \in \{2, 3\}$ be given. The polygon $(c_1, \ldots, c_n)$ is called the *control polygon*.

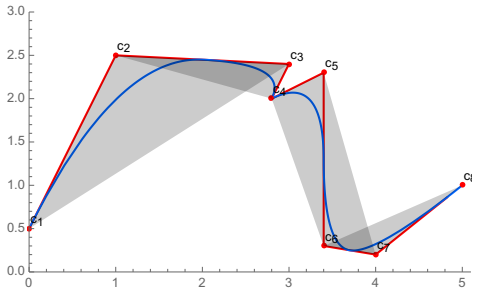


Figure 1.5: Spline curve

A spline curve is given by

$$c(x) = \sum_{i=1}^{n} c_i \widehat{B}_{i,p,\Xi}(x).$$

A spline curve is characterized by p, Ξ and C.

The spline curve has the following property:

- *Convex hull property:* The curve lies in the union of the convex hulls of all $p + 1$ consecutive control points:

$$c(x) \in \bigcup_{i=1}^{n-p} \text{convex-hull}\{c_i, \ldots, c_{i+p}\} \qquad \text{for all } x.$$

  Moreover, on the endpoints and on each knot where we have only C^0 smoothness (this is the same as multiplicity $p - 1$), the curve is interpolatory, cf. Fig. 1.5, where $p = 2$ and $\Xi = (0, 0, 0, 1/5, 2/5, 2/5, 3/5, 4/5, 1, 1, 1)$. That spline curve is interpolatory at c_4.

The construction of B-spline curves does not allow the construction of conic curves (except parabola). Conic curves can be represented by rational functions, which leads us to NURBS (**n**on-**u**niform **r**ational **B**-**s**plines).

A rational curve is given by

$$c(x) = \frac{1}{w(x)} \widetilde{c}(x),$$

where $\widetilde{c}(x)$ is a B-spline curve and the *weight function* $w(x)$ is a B-spline function.

Using this approach, we can represent a circle via

$$w(x) = 1 + x^2, \qquad \widetilde{c}(x) = (1 - x^2, 2x).$$

Note that in this example, the chosen "splines" are just polynomials.

This motivates the use of NURBS. For a given p-open knot vector and given *weights* $W = (w_1, \dots, w_n)$, we define the weight function

$$w(x) = \sum_{j=1}^{n} w_j \widehat{B}_{j,p,\Xi}(x)$$

and the NURBS basis functions

$$\widehat{N}_{i,p,\Xi,W}(x) = \frac{w_i \widehat{B}_{i,p,\Xi}(x)}{\sum_{j=1}^{n} w_j \widehat{B}_{j,p,\Xi}(x)}.$$

Note that we use the coefficients w_i also in the numerator. This is done to preserve the property that the basis functions form a partition of unity.

The NURBS space is given by

$$\widehat{N}_{p,\Xi,W}(0,1) := \operatorname{span}\{\widehat{N}_{1,p,\Xi,W}, \dots, \widehat{N}_{n,p,\Xi,W}\}.$$

Based on this definition of the NURBS function space, we can define NURBS curves completely analogous to spline curves:

$$c(x) = \sum_{i=1}^{n} c_i \widehat{N}_{i,p,\Xi,W}(x).$$

A NURBS curve is characterized by p, Ξ, C and W.

Remark 1.5. *Each NURBS curve c in $\mathbb{R}^d$ is the central projection of a spline curve*

$$\tilde{c}(x) = \begin{pmatrix} c_i \\ w_i \end{pmatrix} \widehat{B}_{i,p,\Xi}(x)$$

in $\mathbb{R}^{d+1}$ to the plane $\mathbb{R}^d \times \{1\}$.

1.2.3 Refinement

The goal of refinement is to be able to represent a function or curve of interest better.

1.2.3.1 h-refinement / knot insertion

Let $\Xi = (\xi_1, \dots, \xi_{n+p+1})$ be a knot vector and $\overline{\xi} \in [\xi_j, \xi_{j+1})$ be a knot to be inserted. Then, the updated knot vector is

$$\overline{\Xi} = (\xi_1, \dots, \xi_j, \overline{\xi}, \xi_{j+1}, \dots, \xi_{n+p+1})$$

and we assume that $\overline{\xi}$ is chosen such that we still have a p-open knot vector. We have

$$\widehat{B}_{i,p,\Xi}(x) = \alpha_i \widehat{B}_{i,p,\overline{\Xi}}(x) + (1 - \alpha_{i+1}) \widehat{B}_{i+1,p,\overline{\Xi}}(x),$$

where

$$\alpha_i = \begin{cases} 1 & \text{for } i = 1, \dots, j-p \\ \frac{\overline{\xi} - \xi_j}{\xi_{j+p} - \xi_j} & \text{for } i = j-p+1, \dots, j \\ 0 & \text{for } i = j+1, \dots, n+1. \end{cases}$$

So, all original B-splines can be represented by the refined ones.

Remark 1.6. *When $\overline{\xi}$ is equal to ξ_j, the knot insertion corresponds to the reduction of the smoothness at that point.*

Having a B-spline function

$$u_h(x) = \sum_{i=1}^{n} u_i \widehat{B}_{i,p,\Xi}(x),$$

we obtain the coefficients $\overline{u}_i$ such that

$$u_h(x) = \sum_{i=1}^{n+1} \overline{u}_i \widehat{B}_{i,p,\overline{\Xi}}(x)$$

via

$$\begin{pmatrix} \overline{u}_1 \\ \vdots \\ \overline{u}_{n+1} \end{pmatrix} = \left(\begin{array}{ccc|ccc|ccc} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ \hline & & & \alpha_{j-p+1} & & & & & \\ & & & 1-\alpha_{j-p+1} & \alpha_{j-p+2} & & & & \\ & & & & 1-\alpha_{j-p+2} & \ddots & & & \\ & & & & & \ddots & \alpha_j & & \\ & & & & & & 1-\alpha_j & & \\ \hline & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & 1 \end{array}\right) \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}.$$

1.2.3.2 p-refinement

The term p-refinement refers to increasing the spline degree p by 1, while keeping the continuities fixed or, in other words, increasing all multiplicities by 1.

Such a refinement corresponds to classical high-order FEM.

p-refinement is again a refinement, so the original space is a subset of the spline space obtained by p-refinement. All functions can be represented with the new basis functions.

1.2.3.3 k-"refinement"

The term k-"refinement" refers to increasing the spline degree p by 1, while keeping the multiplicities fixed or, in other words, increasing the smoothness by 1.

k-"refinement" is *not* a refinement in the classical sense: This approach yields a completely new function space.

1.2.4 Greville mesh

Let Ξ be a p-open knot vector. For each B-spline basis function $\widehat{B}_{i,p,\Xi}$, we associate a *Greville point*

$$\gamma_{i,p,\Xi} := \frac{\xi_{i+1} + \ldots + \xi_{i+p}}{p} \qquad \text{for} \qquad i = 1, \ldots, n.$$

The Greville points are the coefficients of the linear function $u(x) = x$:

$$x = \sum_{i=1}^{n} \gamma_{i,p,\Xi} \widehat{B}_{i,p,\Xi}(x).$$

The Greville points are distinct:

$$\gamma_{i,p,\Xi} < \gamma_{i+1,p,\Xi}.$$

We can associate the Greville points to the basis functions:

$$\gamma_{i,p,\Xi} \leftrightarrow \widehat{B}_{i,p,\Xi}.$$

The Greville points form a partition of Ω (*Greville mesh*)

$$\mathcal{M}_{p,\Xi} = (\gamma_{1,p,\Xi}, \ldots, \gamma_{n,p,\Xi}).$$

Given a spline (or NURBS) curve with control points $C = (c_1, \ldots, c_n)$, the control polygon is given by

$$d(x) = \sum_{i=1}^{n} c_i \widehat{B}_{i,p,\widetilde{\mathcal{M}}}(x),$$

where the functions $\widehat{B}_{i,p,\widetilde{\mathcal{M}}}$ are the piecewise linear functions defined on the Greville mesh $\mathcal{M}$ and $\widetilde{\mathcal{M}} = (\gamma_{1,p,\Xi}, \gamma_{1,p,\Xi}, \gamma_{2,p,\Xi}, \ldots, \gamma_{n,p,\Xi}, \gamma_{n,p,\Xi})$ is the corresponding 1-open knot vector. When applying knot insertion, the corresponding control polygon converges to the curve. If $c \in C^2(0,1)$, then

$$\sup_{x\in(0,1)} |c(x) - d(x)| \leq Ch^2 \sup_{x\in(0,1)} |c''(x)|,$$

where C is independent of h.

1.3 Multivariate splines and NURBS

Multivariate splines are simply defined as tensor-products of univariate splines.

1.3.1 Tensor-product spline and NURBS functions

Let $d \in \{2, 3\}$ be the *spatial dimension*. For every $\ell = \{1, \ldots, d\}$, let $\Xi_\ell = (\xi_{\ell,1}, \ldots, \xi_{\ell,n_\ell+p_\ell+1})$ be a p_ℓ-open knot vector for some p_ℓ and some n_ℓ.

Define

$$\mathbf{p} = (p_1, \ldots, p_d)$$

to be a *multi-index of spline degrees* and

$$\boldsymbol{\Xi} = \Xi_1 \times \cdots \Xi_d = \{(\xi_1, \ldots, \xi_d) : \xi_\ell \in \Xi_\ell\}$$

to be a *multivariate knot vector*.

The B-spline *basis functions* are defined by their tensor product:

$$\widehat{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}}(\mathbf{x}) = \widehat{B}_{i_1,p_1,\Xi_1}(x_1) \cdots \widehat{B}_{i_d,p_d,\Xi_d}(x_d), \tag{1.6}$$

where $\mathbf{i} = (i_1, \ldots, i_d)$ and $\mathbf{p} = (p_1, \ldots, p_d)$ are *multi-indices* and the functions $\widehat{B}_{i_\delta, p_\delta, \Xi_\delta}(x_\delta)$ are univariate B-splines. The *spline function space* is given by

$$\widehat{S}_{\mathbf{p},\boldsymbol{\Xi}}(\widehat{\Omega}) = \operatorname{span}\{\widehat{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}} \ : \ \mathbf{i} \in \mathbf{I}\}. \tag{1.7}$$

For the equidistant case, we again use the notation

$$\widehat{B}_{\mathbf{i},\mathbf{p},\mathbf{k},\mathbf{h}}(\mathbf{x}) = \widehat{B}_{i_1,p_1,k_1,h_1}(x_1) \cdots \widehat{B}_{i_d,p_d,k_d,h_d}(x_d) \quad \text{and} \quad \widehat{B}_{\mathbf{i},\mathbf{p},\mathbf{h}}(\mathbf{x}) = \widehat{B}_{\mathbf{i},\mathbf{p},\mathbf{p}-1,\mathbf{h}}(\mathbf{x})$$

and the analogous notation for the spline space. If we choose the spline degree $\mathbf{p}$ to be a scalar p, we always mean $\mathbf{p} = (p, \cdots, p)$. The same holds for the continuity $\mathbf{k}$ and the grid size $\mathbf{h}$.

We define $\mathbf{I}$ to be the *set of all possible multi-indices*:

$$\mathbf{I} = \{(i_1, \ldots, i_d) \ : \ i_\delta \in I_\delta = \{1, \ldots, N_\delta\}\}.$$

It is of importance for any implementation, to define a *lexicographical ordering*. So, we denote the spline $\widehat{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}}$ also as spline $\widehat{B}_{i,\mathbf{p},\boldsymbol{\Xi}}$, where

$$i = \sum_{\ell=1}^{d} \left(\prod_{j=1}^{\ell-1} n_j \right) (i_\ell - 1) + 1. \tag{1.8}$$

For $d = 3$, this yields $i = i_1 + n_1(i_2 - 1) + n_1 n_2 (i_3 - 1)$. Note that this formula introduces a bijective mapping

$$i \leftrightarrow \mathbf{i} \qquad \text{with} \qquad \{1, \ldots, n_1\} \times \cdots \times \{1, \ldots, n_d\} \leftrightarrow \{1, \ldots, n_1 \cdots n_d\}.$$

Analogously to the univariate case, we introduce the following concepts.

- The tensor-product

  $$\mathbf{Z} = Z_1 \times \cdots Z_d$$

  of *breakpoints* forms a Cartesian grid in the parameter domain $\widehat{\Omega} = (0, 1)^d$, giving the *Bézier mesh*

  $$\widehat{\mathcal{M}} = \{Q_j = I_{1,j_1} \times \cdots \times I_{d,j_d} : I_{\ell,j_\ell} = (\zeta_{\ell,j_\ell}, \zeta_{\ell,j_\ell+1}) \text{ for } 1 \le j_\ell \le N_\ell - 1\}.$$

- For a generic Bézier element $Q_j \in \widehat{\mathcal{M}}$, we also define its *support extension*:

  $$\widetilde{Q}_j := \widetilde{I}_{1,j_1} \times \cdots \times \widetilde{I}_{d,j_d},$$

  where the objects $\widetilde{I}_{\delta,j_\delta}$ are the univariate support extensions, cf. (1.4).

- The *Greville points* are defined by:

  $$\gamma_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}} = (\gamma_{i_1,p_1,\Xi_1}, \cdots, \gamma_{i_d,p_d,\Xi_d}).$$

  Again, the Greville points allow the representation of the linear function:

  $$\mathbf{x} = \sum_{\mathbf{i} \in \mathbf{I}} \gamma_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}} \widehat{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}}(\mathbf{x}),$$

  where both $\mathbf{x}$ and $\gamma_{\mathbf{i},\mathbf{p}}$ are vectors.

Multivariate NURBS are defined as rational tensor-product B-splines. Assume *weights* $w_{\mathbf{i}} \in \mathbb{R}$ to be given for all multi-indices $\mathbf{i} \in \mathbf{I}$. The weight function is given by

$$w(\mathbf{x}) = \sum_{\mathbf{j}\in\mathbf{I}} w_{\mathbf{j}} \widehat{B}_{\mathbf{j},\mathbf{p},\boldsymbol{\Xi}}(\mathbf{x})$$

Now, the *NURBS* basis functions are given by:

$$\widehat{N}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi},W}(\mathbf{x}) = \frac{w_{\mathbf{i}} \widehat{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}}(\mathbf{x})}{\sum_{\mathbf{j}\in\mathbf{I}} w_{\mathbf{j}} \widehat{B}_{\mathbf{j},\mathbf{p},\boldsymbol{\Xi}}(\mathbf{x})}.$$

Note that the multivariate NURBS functions are **not** tensor-products of the univariate NURBS functions. (They where if we would have assumed $w_{\mathbf{i}}$ to have tensor product structure, i.e., that there were coefficients w_{δ,i_δ} such that $w_{\mathbf{i}} = w_{1,i_1} \cdots w_{d,i_d}$.)

The *NURBS function space* is given by

$$\widehat{N}_{\mathbf{p},\boldsymbol{\Xi},W}(\widehat{\Omega}) = \operatorname{span}\{\widehat{N}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi},W} \ : \ \mathbf{i} \in \mathbf{I}\}.$$

The *refinement algorithms* of knot insertion and degree elevation can be generalized to the multivariate splines and NURBS:

- Refinements always apply to one of the spatial dimensions: There we just apply the corresponding algorithm to the B-spline basis.
- The control points and/or weights are chosen such that the manifold/weight function stays the same.

Typically, if we speak about refinement, we mean to apply it with respect to all spacial dimensions.

In practice, we might have implementations, where we do not compute weights corresponding to fine degrees. If we just need to evaluate the weight function, we can do the evaluation also based on the basis functions for the original (coarse) grid.

1.3.2 Spline and NURBS manifolds

As for the curves, we can new define spline parameterizations of multivariate geometries in $\mathbb{R}^m$ as follows. Assume that *control points* $\mathbf{c}_{\mathbf{i}} \in \mathbb{R}^m$ are given. The manifold is given by

$$F(\mathbf{x}) = \sum_{\mathbf{i}\in\mathbf{I}} c_{\mathbf{i}} \widehat{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}}(\mathbf{x}),$$

where $\mathbf{x} \in \widehat{\Omega} = (0,1)^d$. (For NURBS manifolds, just replace $\widehat{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}}$ by $\widehat{N}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi},W}$).

We always choose $m \geq d$:

- $d = 1$: curve in space ($m = 3$) or in the plane ($m = 2$) or just a parameterization of an univariate quantity ($m = d = 1$).
- $d = 2$: surface in space ($m = 3$) or in the plane ($m = 2$).

- $d = 3$: volume in space ($m = 3$).

We can define again the *control mesh* (analogously to the *control polygon*).

We define Bézier mesh on the physical domain Ω to be the F-image of the (open) elements in $\widehat{\mathcal{M}}$:

$$F(\widehat{\mathcal{M}}) = \{Q \subset \Omega \ : \ K = F(\widehat{Q}) \ : \ \widehat{Q} \in \widehat{\mathcal{M}}\}.$$

The Bézier mesh should not be confused with the Greville mesh $\mathcal{M}$, obtained by connecting the images of the Greville points, cf. Fig. 1.6.

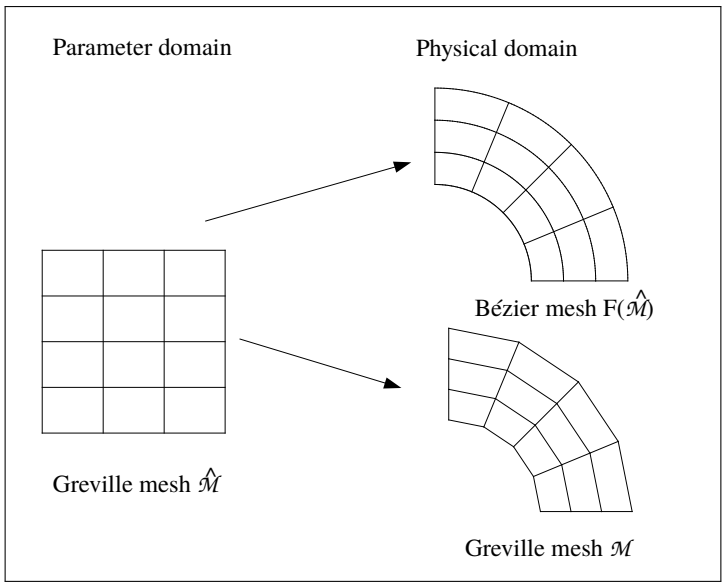


Figure 1.6: Multivariate parameterization

Let $\widehat{h}$ be the *grid size on the parameter domain* $\widehat{\Omega}$:

$$\widehat{h} = \max_{Q \in \widehat{\mathcal{M}}} \widehat{h}_Q \qquad \text{and} \qquad \widehat{h}_Q = \operatorname{diam} Q.$$

Let h be the *grid size on the physical domain* Ω:

$$h = \max_{Q \in \mathcal{M}} h_Q \qquad \text{and} \qquad h_Q = \operatorname{diam} Q.$$

The following assumption guarantees $\widehat{h}_{\widehat{Q}} \eqsim h_Q$:

Assumption 1.7. *The parameterization $F : \widehat{\Omega} \to \Omega$ satisfies*

$$\|\nabla F\|_{L_\infty(\widehat{\Omega})} \leq C \qquad \textit{and} \qquad \|(\nabla F)^{-1}\|_{L_\infty(\widehat{\Omega})} \leq C.$$

The assumption prevents the existence of singularities in F, cf. Fig. 1.7 and 1.8

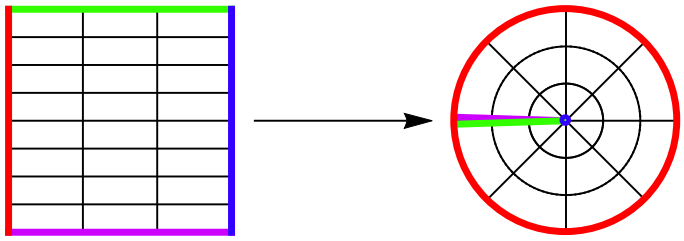


Figure 1.7: A singular parameterization

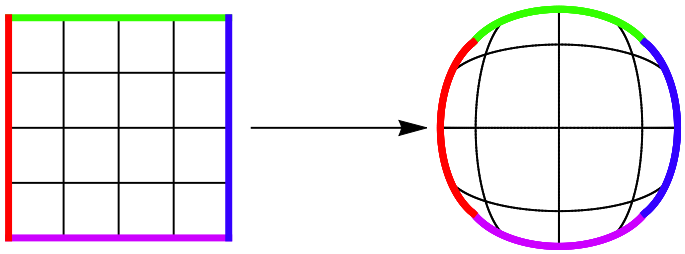


Figure 1.8: A non-singular parameterization

1.4 Isogeometric functions

Assume to have a NURBS or B-spline parameterization of the domain of interest (*physical domain* Ω):

$$F : \widehat{\Omega} = (0,1)^d \to \Omega \subset \mathbb{R}^m.$$

We define on Ω functions as follows (*pull-back principle*):

$$V_h := \{f \circ F^{-1} \ : \ f \in \widehat{V}_h\},$$

where $\widehat{V}_h$ is the corresponding function space on the parameter domain $\widehat{\Omega}$. So, we might choose

$$\widehat{V}_h := \widehat{N}_{\mathbf{p},\boldsymbol{\Xi},W}(\widehat{\Omega}) \quad \text{or} \quad \widehat{V}_h := \widehat{S}_{\mathbf{p},\boldsymbol{\Xi}}(\widehat{\Omega})$$

The basis for the space V_h is also given by the pull-back principle:

$$N_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi},W} = \widehat{N}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi},W} \circ F^{-1} \quad \text{or} \quad B_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}} = \widehat{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}} \circ F^{-1}. \tag{1.9}$$

1.5 Isogeometric Analysis

Now, we can formally introduce Isogeometric Analysis. Usually, we consider Galerkin methods. An alternative are collocation methods, which we will introduce in the second subsection.

1.5.1 Isogeometric Galerkin methods

1.5.1.1 Model problem

We restrict ourselves to partial differential equations of second order. Consider the following *model problem*, which is known as the *Poisson problem* or the *Laplace equation*.

For a given function f, find the unknown function u such that the PDE

$$-\Delta u = f \text{ in } \Omega,$$

is satisfied. We can generalize everything we present to other differential operators and to systems of differential equations.

To make the solution unique, we have to impose boundary conditions:

- *Dirichlet boundary conditions*:
$$u = g_D \text{ on } \Gamma_D,$$
where g_D is a given function.
- *Neumann boundary conditions*:
$$\frac{\partial u}{\partial n} = g_N \text{ on } \Gamma_N,$$
where g_N is a given function and $\frac{\partial u}{\partial n}$ denotes the partial derivative in the direction of the outer normal vector.
- *Robin boundary conditions*:
$$\alpha u + \beta \frac{\partial u}{\partial n} = g_R \text{ on } \Gamma_R,$$
where g_R is a given function and $\alpha > 0$ and $\beta > 0$ are given coefficients.

We have boundary conditions everywhere, i.e.,

$$\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma_R,$$

where $\partial\Omega$ denotes the boundary of Ω, and we only have one boundary condition everywhere, i.e.,

$$\Gamma_i \cap \Gamma_j = \emptyset \text{ for } i, j \in \{D, N, R\} \text{ with } i \neq j.$$

Remark 1.8. *For pure Neumann boundary conditions ($\Gamma_N = \partial\Omega$), we do not have a unique solution, because the constant functions are in the null-space. So, in this case, we have to require*

$$\int_\Omega u \, \mathrm{d}x = 0$$

to obtain uniqueness. Moreover, we have to require

$$\int_\Omega f \, \mathrm{d}x = 0$$

to obtain existence of a solution.

The combination of the PDE and the boundary conditions is called a *boundary value problem* (BVP).

1.5.1.2 Variational formulation

To obtain a Galerkin discretization, we first have to set up a *variational formulation*. First, we multiply the PDE with a *test function* v and integrate over Ω. Now the problem reads as follows.

Find $u \in V$ such that

$$\int_\Omega -\Delta u(x)v(x)\,\mathrm{d}x = \int_\Omega fv(x)\,\mathrm{d}x$$

for all test functions $v \in V$. The choice of the function space V is discussed below.

Now, we apply integration by parts and obtain

$$\int_\Omega \nabla u \cdot \nabla v\,\mathrm{d}x - \int_{\partial\Omega} \frac{\partial}{\partial n}u(x)v(x)\,\mathrm{d}s(x) = \int_\Omega fv(x)\,\mathrm{d}x$$

for all test functions v.

Now, we choose as follows:

$$u \in V_g = \{u \in V \ : \ u = g_D \text{ on } \Gamma_D\},$$

$$v \in V_0 = \{v \in V \ : \ v = 0 \text{ on } \Gamma_D\},$$

i.e., the *ansatz functions* u satisfy the Dirichlet boundary conditions and the test functions v satisfy the homogenous version of the Dirichlet boundary conditions.

Using this choice and using the Neumann and Robin boundary conditions, we obtain

$$\begin{aligned}\int_\Omega \nabla u \cdot \nabla v\,\mathrm{d}x - \int_{\Gamma_D} \frac{\partial}{\partial n}u(x)\underbrace{v(x)}_{=0}\,\mathrm{d}s(x) - \int_{\Gamma_N} \underbrace{\frac{\partial}{\partial n}u(x)}_{=g_N} v(x)\,\mathrm{d}s(x)\\ - \int_{\Gamma_R} \underbrace{\frac{\partial}{\partial n}u(x)}_{=\beta^{-1}g_R - \alpha\beta^{-1}u} v(x)\,\mathrm{d}s(x) = \int_\Omega fv(x)\,\mathrm{d}x.\end{aligned}$$

So, the problem now reads as follows.

Problem 1.9. *Find $u \in V_g$ such that*

$$a(u,v) = \langle f, v\rangle$$

holds for all $v \in V_0$, where

$$\begin{aligned}a(u,v) &:= \int_\Omega \nabla u \cdot \nabla v\,\mathrm{d}x + \frac{\alpha}{\beta}\int_{\Gamma_R} uv(x)\,\mathrm{d}s(x),\\ \langle f, v\rangle &:= \int_\Omega fv(x)\,\mathrm{d}x - \int_{\Gamma_N} g_N v(x)\,\mathrm{d}s(x) - \frac{1}{\beta}\int_{\Gamma_R} g_R v(x)\,\mathrm{d}s(x).\end{aligned}$$

The function spaces can be chosen as follows. Define the *Lebesgue space* $L_2(\Omega)$ to be

$$L_2(\Omega) := \left\{u \ : \ \int_\Omega u(x)^2\,\mathrm{d}x < \infty\right\}$$

and the corresponding norm by

$$\|u\|_{L_2(\Omega)} := \left(\int_\Omega u(x)^2 \, \mathrm{d}x \right)^{1/2}.$$

The term $\|\cdot\|_{L_2(\Omega)}$ is a norm if and only if we consider functions that differ only on a set of measure 0 as being equivalent, cf. courses on functional analysis (*Sobolev spaces*) and numerics of PDEs.

The norm $\|\cdot\|_{L_2(\Omega)}$ is a *Hilbert space norm,* so there is a corresponding scalar product

$$(u, v)_{L_2(\Omega)} = \int_\Omega u(x) v(x) \, \mathrm{d}x.$$

such that $\|u\|_{L_2(\Omega)} = (u, u)^{1/2}_{L_2(\Omega)}$.

Now, define the *Sobolev space* $H^1(\Omega)$ to be

$$H^1(\Omega) := \{u \ : \ u \in L_2(\Omega) \text{ and } \nabla u \in L_2(\Omega)\}$$

and the corresponding *seminorm* norm by

$$|u|_{H^1(\Omega)} := \|\nabla u\|_{L_2(\Omega)} = \left(\sum_{i=1}^{d} \left\| \frac{\partial}{\partial x_i} u \right\|^2_{L_2(\Omega)} \right)^{1/2}.$$

We can also define a *norm* via

$$\|u\|_{H^1(\Omega)} := \left(\|u\|^2_{L_2(\Omega)} + |u|^2_{H^1(\Omega)} \right)^{1/2}.$$

Again, we have a *scalar product*

$$(u, v)_{H^1(\Omega)} := (\nabla u, \nabla v)_{L_2(\Omega)} := \sum_{i=1}^{d} \left(\frac{\partial}{\partial x_i} u, \frac{\partial}{\partial x_i} v \right)_{L_2(\Omega)}$$

such that $|u|_{H^1(\Omega)} = (u, u)^{1/2}_{H^1(\Omega)}$.

If we have a close enough look onto the variational problem (cf. numerics of PDEs), we obtain that we should use

$$V = H^1(\Omega).$$

For this choice and using suitable assumptions (cf. numerics of PDEs), we can show that the variational formulation Problem 1.9 and the original formulation (*strong formulation*) are equivalent. Note that so far we did not deviate from standard finite elements.

Some remarks:

- Neumann boundary conditions are basically a special case of the Robin boundary conditions (for $\alpha = 0$, $\beta = 1$).
- Dirichlet boundary conditions are **not** a special case of the Robin boundary conditions (for $\alpha = 1$, $\beta = 0$), because we cannot divide by 0. But, we can choose β to be a very small number, as an approximation ($\rightarrow$ Nitsche method).

- Boundary conditions that are enforced by choosing the space (in our case: Dirichlet) are called *essential* boundary conditions, the other ones (in our case: Neumann, Robin) are called *natural* boundary conditions.
- Dirichlet boundary conditions can be imposed strongly using the open knot vector setting. Here, we use that exactly one basis function is nonzero on the boundary, cf. Fig 1.9.

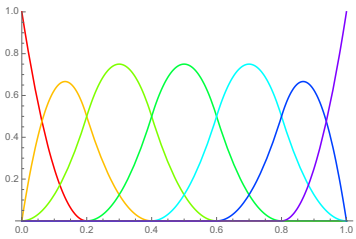


Figure 1.9: Splines of degree 2 without repeated knots

So, we might just eliminate that basis function.

In two or more dimensions, we can basically do the same. However, in general, it is not possible to enforce non-homogenous Dirichlet boundary conditions exactly.

1.5.1.3 Homogenization

For the theory, we observe that we can *homogenize* the variational problem using the following ansatz:

$$u = u_* + u_0,$$

where we choose one fixed $u_* \in V_g$ and use $u_0 \in V_0$. Then, the variational problem reads as follows.

Problem 1.10. *Find $u \in V_0$ such that*

$$a(u, v) = \langle f_0, v \rangle$$

holds for all $v \in V_0$, where

$$\langle f_0, v \rangle := \langle f, v \rangle - a(u_*, v)$$

is a linear functional.

1.5.1.4 Existence and uniqueness

Existence and uniqueness of the solution is guaranteed by the Lax Milgram theorem.

Theorem 1.11. *If a is bounded, i.e.,*

$$a(u, v) \leq \overline{\mu} \|u\|_V \|v\|_V \qquad \text{for all} \qquad u, v \in V_0,$$

and coercive, i.e.,

$$a(v,v) \geq \underline{\mu}\|v\|_V^2 \qquad \textit{for all} \qquad v \in V_0,$$

and f is bounded, i.e.,

$$\langle f, v\rangle \leq \overline{c}\|v\|_V \qquad \textit{for all} \qquad v \in V_0,$$

then the problem,

$$\textit{find} \quad u \in V_0 \quad \textit{such that} \quad a(u,v) = \langle f, v\rangle \quad \textit{for all} \quad v \in V_0, \tag{1.10}$$

has exactly one solution.

This theorem is applicable to our model problem.

1.5.1.5 Discretization

So far, we are still in the original function space (which has infinitely many degrees of freedom). To be able to compute something, we have to *discretize* the problem: We replace the space V by some space V_h (or, equivalently, V_0 by $V_{0,h} = V_0 \cap V_h$).

Conforming discretization: We choose $V_h \subset V$.

Galerkin principle: We use the same subspace for the ansatz functions and the test functions:

$$\text{Find} \quad u_h \in V_{0,h} \quad \text{such that} \quad a(u_h, v_h) = \langle f, v_h\rangle \quad \text{for all} \quad v_h \in V_{0,h}. \tag{1.11}$$

Note that u_h is the *a-orthogonal projection* of u into $V_{0,h}$, i.e.,

$$a(u - u_h, v_h) = 0 \quad \text{for all} \quad v_h \in V_{0,h}.$$

As we have a conforming Galerkin discretization, the conditions of the Lax Milgram theorem carry over to V_h (or $V_{h,0}$), so we have again existence and uniqueness.

1.5.1.6 Error estimates

Ceá's Lemma allows to estimate the error.

Lemma 1.12. *Assume that a and f satisfy the conditions of the Lax Milgram theorem. Let u be the solution of the original problem* (1.10) *and u_h be the solution of the discretized problem* (1.11)*. Then, we have*

$$\|u - u_h\|_V \leq \frac{\overline{\mu}}{\underline{\mu}} \inf_{v_h \in V_h} \|u - v_h\|_V,$$

i.e., we can bound the discretization error $\|u - u_h\|_V$ *by a constant times the* approximation error $\inf_{v_h \in V_h} \|u - v_h\|_V$.

This motivates approximation error estimates (see Chapters 2 and 3). We will see, that we can bound

$$\inf_{v_h \in S_{\mathbf{p},\boldsymbol{\Xi}}(\Omega)} \|u - v_h\|_{H^1(\Omega)} \leq C(\mathbf{p}, \Xi, r, F) h^{r-1} |u|_{H^r(\Omega)},$$

with $1 \leq r \leq \min_\delta p_\delta + 1$.

The Sobolev space $H^r(\Omega)$ is given by

$$H^r(\Omega) = \Big\{ u \ : \ \frac{\partial^{l_1+\cdots+l_d}}{\partial x_1^{l_1} \cdots \partial x_d^{l_d}} u \in L_2(\Omega) \text{ for all } \mathbf{l} = (l_1, \ldots, l_d) \in (\mathbb{Z}_0^+)^d \text{ such that } |\mathbf{l}| \leq r \Big\},$$

where $|\mathbf{l}| := l_1 + \cdots + l_d$. The corresponding seminorm is given by

$$|u|_{H^r(\Omega)} = \left(\sum_{\mathbf{l} \in (\mathbb{Z}_0^+)^d : |\mathbf{l}| = r} \Big\| \frac{\partial^{l_1+\cdots+l_d}}{\partial x_1^{l_1} \cdots \partial x_d^{l_d}} u \Big\|_{L_2(\Omega)}^2 \right)^{1/2}$$

and the corresponding seminorm is given by

$$\|u\|_{H^r(\Omega)} = \left(\sum_{\mathbf{l} \in (\mathbb{Z}_0^+)^d : |\mathbf{l}| \leq r} \Big\| \frac{\partial^{l_1+\cdots+l_d}}{\partial x_1^{l_1} \cdots \partial x_d^{l_d}} u \Big\|_{L_2(\Omega)}^2 \right)^{1/2}.$$

1.5.1.7 Regularity

Having such approximation error estimates motivates the next question: How large is $\|u\|_{H^r(\Omega)}$?

If $\|u\|_{H^r(\Omega)} = \infty$, the above result would be pointless.

There are many regularity results. Important examples are:

- If $\Omega \subset \mathbb{R}^2$ is *convex polygonal domain,* and all boundary conditions are of the same type, and Ω is simply connected, and $f \in L_2(\Omega)$, and $g_D = g_N = g_R = 0$, and ..., then $u \in H^2(\Omega)$.
- If $\Omega \subset \mathbb{R}^2$ has a *sufficiently smooth boundary,* and all boundary conditions are of the same type, and Ω is simply connected, and $f \in L_2(\Omega)$, and $g_D = g_N = g_R = 0$, and ..., then $u \in H^2(\Omega)$.

There are many more special cases, where $u \in H^2(\Omega)$ can be shown. In all cases, we have the following kind of result:

$$\|u\|_{H^2(\Omega)} \leq c \|f\|_{L_2(\Omega)}.$$

We cannot hope for results for more than H^2. If H^2-regularity does not hold, we might at least hope for some result $u \in H^{1+s}(\Omega)$ with $s \in (0, 1)$.

In *the interior of the open domain* Ω, we have everything we want. Let Ω_0 be an open domain such that $\overline{\Omega_0} \subset \Omega$, then

$$f \in H^r(\Omega) \Rightarrow u|_{\Omega_0} \in H^{r+2}(\Omega_0),$$

where $u|_{\Omega_0}$ is the restriction of u to Ω_0.

1.5.1.8 Matrix-vector formulation

The next step is the reformulation of the problem in a matrix-vector formulation. We make the following ansatz:

$$u_h(x) = \sum_{i=1}^{N} u_i \varphi_i(x), \qquad \text{and} \qquad v_h(x) = \sum_{i=1}^{N} v_i \varphi_i(x),$$

where the functions φ_i are the B-spline or NURBS basis functions (1.9), ordered in a lexicographical ordering (1.8).

We obtain

$$\sum_{i=1}^{N}\sum_{j=1}^{N} u_i v_j a(\varphi_i, \varphi_j) = \sum_{j=1}^{N} v_j \langle f, \varphi_j \rangle.$$

By defining

- the *solution vector* $\underline{u}_h := (u_1, \dots, u_N)$,
- the test function vector $\underline{v}_h := (v_1, \dots, v_N)$,
- the *load vector* $\underline{f}_h := [\langle f, \varphi_i \rangle]_{i=1}^N$, and
- the *stiffness matrix* $A_h := [a(\varphi_i, \varphi_j)]_{i,j=1}^N$,

the above linear system reads as follows

$$\underline{v}_h^T A_h \underline{u}_h = \underline{v}_h^T \underline{f}_h \qquad \text{for all} \qquad \underline{v}_h \in \mathbb{R}^N$$

or just

$$A_h \underline{u}_h = \underline{f}_h. \tag{1.12}$$

Note that $\underline{u}_h$ and $\underline{f}_h$ have a completely different definition.

The linear system has the following properties:

- It is large scale.
- The matrix A_h is sparse: It has only $\mathcal{O}((1+2p)^d)$ elements per row.
- A_h is symmetric and positive definite (as $a(\cdot,\cdot)$ has been symmetric and coercive).
- For solving the problem, we have to do particularly the following steps:
  - Compute the entries of A_h and $\underline{f}_h$.
  - Solve the linear system (1.12).

1.5.2 Isogeometric collocation methods

Isogeometric collocation methods are an alternative to Galerkin methods.

Consider only the univariate case. The model problem reads as follows

$$-u''(x) = f(x)$$

for $x \in (0,1)$ and we assume to have Dirichlet boundary conditions $u(0) = u(1) = 0$.

We discretize the problem as follows: We replace u by a spline function $u_h \in S_{p,\Xi}(0,1)$ and require

$$-u_h''(x_i) = f(x_i)$$

for some *collocation points* $(x_i)_{i=2}^{n-1}$ and $u_h(0) = u_h(1) = 0$. The number of collocation points plus the number of boundary conditions has to coincide with the number of basis functions.

We can – as for the Galerkin approach – make an ansatz

$$u_h(x) = \sum_{i=1}^{n} u_i \widehat{B}_{i,p,\Xi}(x)$$

and obtain

$$\begin{aligned}
&\sum_{j=1}^{n} u_j \widehat{B}''_{j,p,\Xi}(x_i) = f(x_i) \qquad \text{for all} \qquad i = 2, \dots, n-1, \\
&\sum_{j=1}^{n} u_j \widehat{B}_{j,p,\Xi}(0) = 0, \\
&\sum_{j=1}^{n} u_j \widehat{B}_{j,p,\Xi}(1) = 0.
\end{aligned}$$

As the spline functions are interpolatory on the boundary, the second two lines simplify to $u_1 = 0$ and $u_n = 0$. So, the overall problem simplifies to

$$\sum_{j=2}^{n-1} u_j \widehat{B}''_{j,p,\Xi}(x_i) = f(x_i) \qquad \text{for all} \qquad i = 2, \dots, n-1.$$

This linear system be rewritten in matrix-vector formulation as

$$A_h \underline{u}_h = \underline{f}_h,$$

where

- the *stiffness matrix* is given by $A_h = [B''_{j,p,\Xi}(x_i)]_{i,j=2}^{n-1}$,
- the *load vector* is given by $\underline{f}_h = [f(x_i)]_{i=2}^{n-1}$,
- the *solution vector* $\underline{u}_h = (u_2, \cdots, u_{n-1})$ is such that $u_h(x) = \sum_{i=2}^{n-1} u_i B_{i,p,\Xi}(x)$.

Some remarks:

- A proper choice of the points x_i is of importance. One possibility are the Greville points.
- Another possibility (which is understood better) are the DEMKO points.
- All boundary conditions (Dirichlet, Neumann, Robin) have to be imposed strongly (essential boundary conditions).
- The collocation stiffness matrix A_h is non-symmetric.
- The bandwidth of the stiffness matrix A_h is smaller than for the Galerkin discretization: The bandwidth is $2p + 1$ for Galerkin and $p + 1$ for collocation.
- In the interior, the collocation stiffness matrix for the spine degree $2p$ looks like the Galerkin stiffness matrix for degree p. At the boundary, the matrices look differently.

- There is only little convergence theory available. In practice, we have one of the following two cases:
  - The collocation method converges as well as the Galerkin method. In this case, the collocation method is nice because the stiffness matrix bandwidth is smaller and its entries are easier to compute.
  - The collocation method does not work at all.

1.6 Some conclusions

Isogeometric Analysis:

- global geometry mapping
- smooth basis functions
- global geometry mapping does not allow complicated domains → multi-patch constructions
- requires appropriate geometry representation (can be similar to meshing)

Finite element Analysis:

- local geometry mapping
- typically only C^0
- the construction does not allow to impose more smoothness
- no problems with complicated domains
- requires meshing

1.7 Literature

This chapter follows mainly [2]. The subsection 1.5.1 follows also [3]. The subsection 1.5.2 follows mainly [4].

Chapter 2

Approximation error estimates

Given a function u, we are interested in finding a spline $u_h \in S_{p,\Xi}$ such that the approximation error $\|u - u_h\|$ for some appropriately chosen norm $\|\cdot\|$ is small.

2.1 Univariate approximation

2.1.1 Interpolants for the Courant element and for the step function

The *Courant elements* are nothing but the B-spline basis for degree $p = 1$, see Fig. 2.1.

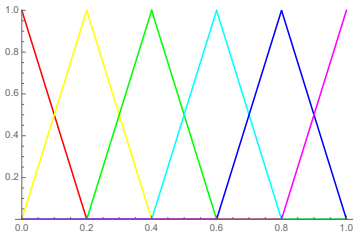


Figure 2.1: Courant element

Let

$$\Xi = (x_1, x_1, x_2, x_3, \ldots, x_{n-1}, x_n, x_n).$$

be a 1-open knot vector.

The Courant element forms a *nodal basis*, i.e., we have

$$\widehat{B}_{i,1,\Xi}(x_j) = \delta_{i,j},$$

where

$$\delta_{i,j} := \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

is the *Kronecker delta*.

Based on this property, we introduce the following *interpolation* operator:

$$\Pi_{1,\Xi} : C^0(0,1) \to S_{1,\Xi}(0,1)$$
$$\Pi_{1,\Xi} u := \sum_{i=1}^{n} u(x_i) B_{i,1,\Xi}.$$

The same is possible for the *step functions*, which we denote by $\widehat{B}_{i,0,\Xi}$, see Fig. 2.2. Here,

$$\Xi = (\xi_1, \ldots, \xi_{n+1})$$

is a 0-open knot vector, and

$$\widehat{B}_{i,0,\Xi}(x) = \begin{cases} 1 & \text{if } \xi_i \leq x < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}.$$

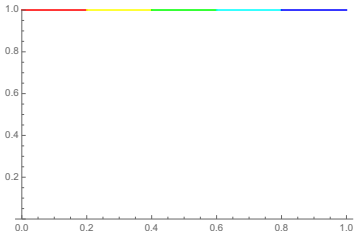


Figure 2.2: Piece-wise constants

Here, we might define the nodes $x_i := \frac{1}{2}(\xi_i + \xi_{i+1})$ to be the midpoints of the elements $I_i = (\xi_i, \xi_{i+1})$ and obtain again

$$\widehat{B}_{i,0,\Xi}(x_j) = \delta_{i,j},$$

and define a projector $\Pi_{0,\Xi}$ as above.
The operators $\Pi_{p,\Xi}$ for $p \in \{0,1\}$ have the following properties:

- They are *interpolatory* at the nodes x_i.
- They only depend on the function value of u at the nodes.
- They are *linear* operators.
- They are *projectors*, i.e., we have $\Pi_{p,\Xi}\Pi_{p,\Xi} u = \Pi_{p,\Xi} u$ for all u.

We have the following approximation error estimate.

Theorem 2.1. *The approximation error estimate*

$$\|u - \Pi_{0,\Xi} u\|_{L_2(0,1)} \leq \frac{h}{2} |u|_{H^1(0,1)}$$

holds for all $u \in H^1(0,1)$*, where* h *is the* global grid size (1.1).

Proof. Assume that u is arbitrary but fixed and define $u_h := \Pi_{0,\Xi} u$. Consider the intervals $I_i = (\xi_i, \xi_{i+1})$ and recall that that $u_h|I_i = u(\xi_{i+1/2})$, where $\xi_{i+1/2} := \xi_{i+1/2}$. So, we obtain

$$\begin{aligned}\|(I-\Pi_{0,\Xi})u\|^2_{L_2(0,1)} &= \sum_{i=1}^{n-1} \int_{\xi_i}^{\xi_{i+1}} \big(u(x)-u_h(x)\big)^2 \,\mathrm{d}x \\ &= \sum_{i=1}^{n-1} \int_{\xi_i}^{\xi_{i+1}} \Big(u(x)-u(\xi_{i+1/2})\Big)^2 \,\mathrm{d}x = \sum_{i=1}^{n-1} \int_{\xi_i}^{\xi_{i+1/2}} \Big(\pm \int_{\xi_{i+1/2}}^{x} u'(\eta)\,\mathrm{d}\eta\Big)^2 \,\mathrm{d}x,\end{aligned}$$

where we use the convention $\int_A^B \ldots \mathrm{d}x = \int_B^A \ldots \mathrm{d}x$ and where "$\pm$" = "$+$" if $x \geq \xi_{i+1/2}$ and "$\pm$" = "$-$" if $x < \xi_{i+1/2}$. In any case, we have $(\pm 1)^2 = 1$. Now, we obtain using the Cauchy-Schwarz inequality further

$$\begin{aligned}\|(I-\Pi_{0,\Xi})u\|^2_{L_2(0,1)} &\leq \sum_{i=1}^{n-1} \int_{\xi_i}^{\xi_{i+1}} \int_{\xi_{i+1/2}}^{x} u'(\eta)^2 \,\mathrm{d}\eta \int_{\xi_{i+1/2}}^{x} 1 \,\mathrm{d}\eta \,\mathrm{d}x \\ &\leq \frac{h}{2} \sum_{i=1}^{n-1} \int_{\xi_i}^{\xi_{i+1}} \int_{\xi_{i+1/2}}^{x} u'(\eta)^2 \,\mathrm{d}\eta \,\mathrm{d}x \\ &= \frac{h}{2} \sum_{i=1}^{n-1} \left(\int_{\xi_i}^{\xi_{i+1/2}} \int_{x}^{\xi_{i+1/2}} u'(\eta)^2 \,\mathrm{d}\eta \,\mathrm{d}x + \int_{\xi_{i+1/2}}^{\xi_{i+1}} \int_{\xi_{i+1/2}}^{x} u'(\eta)^2 \,\mathrm{d}\eta \,\mathrm{d}x \right) \\ &\leq \frac{h}{2} \sum_{i=1}^{n-1} \left(\int_{\xi_i}^{\xi_{i+1/2}} \int_{\xi_i}^{\xi_{i+1/2}} u'(\eta)^2 \,\mathrm{d}\eta \,\mathrm{d}x + \int_{\xi_{i+1/2}}^{\xi_{i+1}} \int_{\xi_{i+1/2}}^{\xi_{i+1}} u'(\eta)^2 \,\mathrm{d}\eta \,\mathrm{d}x \right) \\ &= \frac{h}{2} \sum_{i=1}^{n-1} \left(\int_{\xi_i}^{\xi_{i+1/2}} \mathrm{d}x \int_{\xi_i}^{\xi_{i+1/2}} u'(\eta)^2 \,\mathrm{d}\eta + \int_{\xi_{i+1/2}}^{\xi_{i+1}} \mathrm{d}x \int_{\xi_{i+1/2}}^{\xi_{i+1}} u'(\eta)^2 \,\mathrm{d}\eta \right) \\ &\leq \frac{h^2}{4} \sum_{i=1}^{n-1} \left(\int_{\xi_i}^{\xi_{i+1/2}} u'(\eta)^2 \,\mathrm{d}\eta + \int_{\xi_{i+1/2}}^{\xi_{i+1}} u'(\eta)^2 \,\mathrm{d}\eta \right) \\ &= \frac{h^2}{4} \sum_{i=1}^{n-1} \left(\int_{\xi_i}^{\xi_{i+1}} u'(\eta)^2 \,\mathrm{d}\eta \right) = \frac{h^2}{4} |u|^2_{H^1(0,1)},\end{aligned}$$

which was to show. □

Similarly, we can show the following theorem.

Theorem 2.2. *We have*

$$\|u - \Pi_{1,\Xi} u\|_{L_2(0,1)} \leq ch|u|_{H^1(0,1)}$$

for all $u \in H^1(0,1)$*, and*

$$|u - \Pi_{1,\Xi} u|_{H^1(0,1)} \leq ch|u|_{H^2(0,1)}$$

for all $u \in H^2(0,1)$*, where* h *is the* global grid size (1.1)*.*

As $\Pi_{1,\Xi}$ is a projector, we obtain $(I-\Pi_{1,\Xi})^2 = (I-\Pi_{1,\Xi})$ and therefore also

$$\|(I-\Pi_{1,\Xi})u\|_{L_2(0,1)} = \|(I-\Pi_{1,\Xi})^2 u\|_{L_2(0,1)} \leq ch|(I-\Pi_{1,\Xi})u|_{H^1(0,1)} \leq c^2h^2|u|_{H^2(0,1)}. \quad (2.1)$$

Obviously, the *approximation error* is bounded by the *interpolation error*:

$$\inf_{u_h \in S_{p,\Xi}(0,1)} |u-u_h|_{H^1} \leq |u - \Pi_{p,\Xi} u|_{H^1}.$$

2.1.2 A simple interpolant for splines

The results of the last subsection cannot be extended for $p \geq 2$ because splines do not form a nodal basis. In other words: there is no set of nodes x_i such that

$$\widehat{B}_{i,p,\Xi}(x_j) = \delta_{i,j}.$$

However, we can construct a subspace of the spline space such that such a condition is satisfied. Assume that p and Ξ are fixed and assume that $n/p \in \mathbb{Z}$, where n is the number of basis functions. (The last assumption can be dropped, see Remark 2.4.)

Then, we define basis functions $\varphi_1, \ldots, \varphi_m$ with $m = n/p$ as follows:

$$\widehat{V}_{i,p,\Xi} := \sum_{l=1}^{p} \widehat{B}_{(i-1)p+l,p,\Xi}.$$

We obtain a spline space $V_{i,p,\Xi}(0,1) := \operatorname{span}\{\widehat{V}_{1,p,\Xi}, \ldots, \widehat{V}_{m,p,\Xi}\} \subset S_{p,\Xi}(0,1)$.

For $p = 2$, the basis functions $\widehat{V}_{i,p,\Xi}$ are depicted in Fig. 2.3. The corresponding B-splines are depicted with dashed lines.

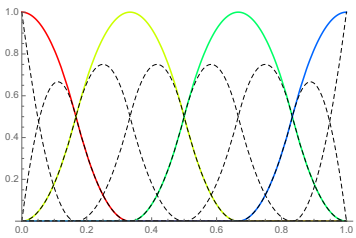


Figure 2.3: Splines $\widehat{V}_{i,p,\Xi}$ for $p = 2$

Observe that the basis functions $\widehat{V}_{i,p,\Xi}$ satisfy the following properties:

- *Non negativity*, cf. (1.2): $0 \leq \widehat{V}_{i,p,\Xi}(x) \leq 1$.
- *Partition of unity*, cf. (1.3)
- *Nodal basis:*

  $$\widehat{V}_{i,p,\Xi}(x_j) = \delta_{i,j},$$

  where

  $$x_j := \xi_{(j-1)p+(p+1)}.$$

- *Bounded support:*

  $$\operatorname{supp} \widehat{V}_{i,p,\Xi} = [x_{i-1}, x_{i+1}],$$

  where $x_{-1} := 0$ and $x_{m+1} := x_m$.

As we have a nodal basis, we define the following interpolation operator:

$$\Pi_{p,\Xi} : C^0(0,1) \to S_{p,\Xi}(0,1)$$

$$\Pi_{p,\Xi} u := \sum_{i=1}^{n/p} u(x_i) \widehat{V}_{i,p,\Xi}.$$

Theorem 2.3. *For $p \geq 1$, the approximation error estimate*

$$\|u - \Pi_{p,\Xi} u\|_{L_2(0,1)} \leq \sqrt{2} H |u|_{H^1(0,1)}$$

holds for all $u \in H^1(0,1)$, where H is the grid size *of the grid $(x_0, x_1, \ldots, x_m)$*

$$H = \max_{i=1,\ldots,m-1} x_{i+1} - x_i = \max_{i=1,\ldots,m-1} (\xi_{ip+(p+1)} - \xi_{(i-1)p+(p+1)}).$$

We have

$$H \leq ph. \tag{2.2}$$

Proof. Assume that u is arbitrary but fixed and define $u_h := \Pi_{p,\Xi} u$. Consider the intervals $I_i = (x_i, x_{i+1})$ for $i = 1, \ldots, m-1$. Observe that on this interval only the two basis functions

$$\varphi_i \qquad \text{and} \qquad \varphi_{i+1}$$

are active. Using this, non-negativity and partition of unity, we know that there are coefficients $\alpha(x) \in [0,1]$ such that

$$u_h(x) = \alpha(x) u(x_i) + (1 - \alpha(x)) u(x_{i+1}) \qquad \text{for} \qquad x \in I_i.$$

So, we obtain using $(A+B)^2 \leq 2(A^2 + B^2)$:

$$\begin{aligned}
\|(I - \Pi_{p,\Xi})u\|^2_{L_2(0,1)} &= \sum_{i=1}^{m-1} \int_{x_i}^{x_{i+1}} \big(u(x) - \alpha(x) u(x_i) - (1 - \alpha(x)) u(x_{i+1})\big)^2 \,\mathrm{d}x \\
&\leq 2 \sum_{i=1}^{m-1} \int_{x_i}^{x_{i+1}} \alpha(x)^2 \Big(u(x) - u(x_i)\Big)^2 + (1 - \alpha(x))^2 \Big(u(x) - u(x_{i+1})\Big)^2 \,\mathrm{d}x \\
&\leq 2 \sum_{i=1}^{m-1} \int_{x_i}^{x_{i+1}} \Big(u(x) - u(x_i)\Big)^2 + \Big(u(x) - u(x_{i+1})\Big)^2 \,\mathrm{d}x.
\end{aligned}$$

Observe that the main theorem of integration and Cauchy-Schwarz inequality yield

$$\Big(u(x) - u(x_i)\Big)^2 = \Big(\int_x^{x_i} u'(\xi) \,\mathrm{d}\xi\Big)^2 \leq |x - x_i| \int_x^{x_i} u'(\xi)^2 \,\mathrm{d}\xi \leq H \int_x^{x_i} u'(\xi)^2 \,\mathrm{d}\xi.$$

We obtain an analogous result for $\Big(u(x) - u(x_{i+1})\Big)^2$. So, we obtain further

$$\begin{aligned}
\|(I - \Pi_{p,\Xi})u\|^2_{L_2(0,1)} &\leq 2H \sum_{i=1}^{m-1} \int_{x_i}^{x_{i+1}} \int_{x_i}^{x} u'(\xi)^2 \,\mathrm{d}\xi + \int_{x}^{x_{i+1}} u'(\xi)^2 \,\mathrm{d}\xi \,\mathrm{d}x \\
&= 2H \sum_{i=1}^{m-1} \int_{x_i}^{x_{i+1}} \int_{x_i}^{x_{i+1}} u'(\xi)^2 \,\mathrm{d}\xi \,\mathrm{d}x \\
&= 2H^2 \sum_{i=1}^{m-1} \int_{x_i}^{x_{i+1}} u'(\xi)^2 \,\mathrm{d}\xi = 2H^2 |u|^2_{H^1(0,1)}
\end{aligned}$$

which was to show. The estimate (2.2) is trivial. □

Some comments:

- The dependence in the grid size is – as we have expected – optimal.
- In the spline degree, the result is optimal concerning the considered subspace. If we consider the full subspace, the result is not optimal.
- The interpolation operator is local.
- We do not need to stick to adding up exactly p basis functions:

  Remark 2.4. *The approximation error result of Theorem 2.3 also holds if always* at least p *basis functions are added up. Instead of* (2.2)*, we obtain that H is bounded by h times the largest number of basis functions added up.*

  If we do not assume $n/p \in \mathbb{Z}$, we can modify the definition of the basis functions $\widehat{V}_{i,p,\Xi}$ as follows: we define $m := \lfloor n/p \rfloor$ basis functions, where the first $m-1$ basis functions are the sum of p B-splines. The last basis function $\widehat{V}_{m,p,\Xi}$ is the sum of the remainder (at most $2p-1$ basis functions). So, we obtain both the approximation error estimate of Theorem 2.3 and $H \leq 2ph$.

- The interpolation error estimate (Theorem 2.3 in combination with Remark 2.4) yields again an approximation error estimate both for the spline space $V_{p,\Xi}(0,1)$ and for the original space $S_{p,\Xi}(0,1)$. For the latter:

$$\inf_{u_h \in S_{p,\Xi}(0,1)} \|u - u_h\|_{L_2(0,1)} \leq 2p\, h |u|_{H^1(0,1)}. \tag{2.3}$$

  The same error estimate is obtained for the L_2-orthogonal projector into $S_{p,\Xi}$.

Remark 2.5. *An interpolation error estimate of the form*

$$\|u - \widetilde{\Pi}_{p,\Xi} u\|_{L_2(0,1)} \leq c\, p\, h |u|_{H^1(0,1)}. \tag{2.4}$$

can be also derived for the following choice:

$$(\widetilde{\Pi}_{p,\Xi} u) := \sum_{i=1}^{n} u(\gamma_i) \widehat{B}_{i,p,\Xi},$$

where $(\gamma_i)_{i=1}^n$ are the Greville points. Note that this choice is simpler the discussed one. However, an error estimate of the order $\leq c\,p\,h$ is not *a (quasi-)optimal n-width for the spline space. An quasi-optimal n-width is a result of the form*

$$\|u - \Pi u\|_{L_2} \leq c \tfrac{1}{\dim V_h} |u|_{H^1},$$

where Π is a projector into V_h. As $\dim S_{p,\Xi} \eqsim h^{-1} + p$ and $\dim V_{p,\Xi} \eqsim h^{-1}/p$, the estimate of Theorem 2.3 yields a quasi-optimal n-with, while the estimate (2.4) *is not.*

The estimate 2.3 can be extended to higher Sobolev indices as follows.

Theorem 2.6. *For $0 \leq r \leq p$ with $r, p \in \mathbb{Z}$, the approximation error estimate*

$$\inf_{u_h \in S_{p,\Xi}(0,1)} |u - u_h|_{H^r(0,1)} \leq 2p\, h |u|_{H^{r+1}(0,1)}$$

holds for all $u \in H^{r+1}(0,1)$.

Proof. We show this theorem by induction. We knot that the result holds for $r = 0$, see (2.3). So, we assume to know

$$\inf_{u_h \in S_{p,\Xi}(0,1)} |u - u_h|_{H^r(0,1)} \leq 2p\, h |u|_{H^{r+1}(0,1)} \tag{2.5}$$

and that we are interested in showing

$$\inf_{u_h \in S_{p,\Xi}(0,1)} |u - u_h|_{H^{r+1}(0,1)} \leq 2p\, h |u|_{H^{r+2}(0,1)}. \tag{2.6}$$

Note that

$$\inf_{u_h \in S_{p,\Xi}(0,1)} |u - u_h|_{H^{r+1}(0,1)} = \inf_{u_h \in S_{p,\Xi}(0,1)} |u' - u_h'|_{H^r(0,1)}.$$

Using (1.5), we obtain

$$\inf_{u_h \in S_{p,\Xi}(0,1)} |u - u_h|_{H^{r+1}(0,1)} = \inf_{v_h \in S_{p-1,\Xi'}(0,1)} |u' - v_h|_{H^r(0,1)}.$$

Now, the induction hypothesis (2.5) shows (2.6). □

Note that we have proven this theorem based on the space $V_{p,\Xi}(0,1)$, however for that space directly we do not have such high-order approximation error estimates as in Theorem 2.6, i.e., we cannot expect

$$\inf_{u_h \in V_{p,\Xi}(0,1)} |u - u_h|_{H^r(0,1)} \leq 2p\, h |u|_{H^{r+1}(0,1)}$$

to be true for $r > 0$.

Corollary 2.7. *For $0 \leq s < r \leq p+1$ with $s, r, p \in \mathbb{Z}$, the approximation error estimate*

$$\inf_{u_h \in S_{p,\Xi}(0,1)} |u - u_h|_{H^s(0,1)} \leq (2p\, h)^{r-s} |u|_{H^r(0,1)}$$

holds for all $u \in H^r(0,1)$.

Proof. This is obtained by chaining Theorem 2.6 as in (2.1). □

2.1.3 The Schumaker quasi-interpolant

In the last section, we have introduced a local projector into the spline space that satisfies an $L_2 - H^1$-error estimates. We did not introduce a particular interpolant that also satisfies the higher order estimates.

Such an interpolant is given in the book [5].

Assume Ξ to be a p-open knot vector. Following the construction in [5], there is a dual basis $[\lambda_i]_{i=1}^n$ with

$$\lambda_i : H^q(0,1) \to \mathbb{R}$$

such that

$$\lambda_i(\widehat{B}_{j,p,\Xi}) = \delta_{i,j}.$$

Those basis functions have the form

$$\lambda_i(v) = \int_0^1 \psi_i(x)v(x)\,\mathrm{d}x,$$

where ψ_i is a spline with local support:

$$\operatorname{supp}\psi_i = [\xi_i, \xi_{i+p+1}].$$

The functions ψ_i are locally defined and only depend on the local knot vector $\Xi(\widehat{B}_{i,p,\Xi})$.
Having a nodal basis, we *again* define

$$\Pi_{p,\Xi}u := \sum_{i=1}^{n} \lambda_i(u)\widehat{B}_{i,p,\Xi}.$$

Based on these definitions, the following theorems are proven.

Theorem 2.8. *For any non-empty knot span $I_i = (\zeta_i, \zeta_{i+1})$, we have*

$$\|\Pi_{p,\Xi}u\|_{L_2(I_i)} \leq C\|u\|_{L_2(\widetilde{I}_i)},$$

and

$$|\Pi_{p,\Xi}u|_{H^1(I_i)} \leq C|u|_{H^1(\widetilde{I}_i)},$$

where $\widetilde{I}_i$ is the support extension and C only depends on p.

Theorem 2.9. *There is a positive constant C depending only on p such that for all $s \in \mathbb{Z}$ with $0 < s \leq p+1$, the estimate*

$$\|(I - \Pi_{p,\Xi})u\|_{L_2(I_i)} \leq C\widetilde{h}_i^s|u|_{H^s(\widetilde{I}_i)}$$

holds for all $u \in H^s(I)$. If he partition is locally quasi uniform, we also obtain for any $r, s \in \mathbb{Z}$ with $0 \leq r < s \leq p+1$

$$|(I - \Pi_{p,\Xi})u|_{H^r(I_i)} \leq C\widetilde{h}_i^{s-r}|u|_{H^s(\widetilde{I}_i)}.$$

2.2 Multivariate approximation

2.2.1 Approximation on the parameter domain

For simplicity, we assume without loss of generality that $\widehat{\Omega} = (0,1)^2$.

For each spatial dimension $l \in \{1,2\}$, we assume a discretization space to be given, say $V_l := S_{p_l,\Xi_l}(0,1)$. Moreover, we assume that we have chosen our favorite univariate projection operator $\Pi_l : C^\infty(0,1) \to V_l$.

On the parameter domain $\widehat{\Omega}$, we consider the space

$$\widehat{V}_h = V_1 \otimes V_2,$$

cf. (1.6) and (1.7).

Now, a projector for the parameter domain is introduced as follows. First we define the following projectors:

$$(\widehat{\Pi}_1 u)(x,y) = (\Pi_1 u(\cdot,y))(x)$$
$$(\widehat{\Pi}_2 u)(x,y) = (\Pi_2 u(x,\cdot))(y).$$

The operator $\widehat{\Pi}_1$ is to be understood as follows. For each fixed choice of y, we restrict the function u to be a univariate function $u(\cdot,y)$. We apply the projector Π_1 to that univariate function. Now, we define a bivariate function w such that for each y we have $w(\cdot,y) = \Pi_1 u(\cdot,y)$. The projector $\widehat{\Pi}_1$ is exactly the mapping $u \to w$.

Certainly, $\widehat{\Pi}_1$ maps $C^0(\widehat{\Omega}) \to V_1 \otimes C^0(0,1)$ and $\widehat{\Pi}_2$ maps $C^0(\widehat{\Omega}) \to C^0(0,1) \otimes V_2$.

Now, we define an operator on $C^0(\widehat{\Omega})$:

$$\widehat{\Pi} := \widehat{\Pi}_1 \widehat{\Pi}_2.$$

Lemma 2.10. *$\widehat{\Pi} := \widehat{\Pi}_1\widehat{\Pi}_2 = \widehat{\Pi}_2\widehat{\Pi}_1$ is a projector into $V_1 \otimes V_2$.*

Proof. The spaces V_δ have bases: $V_\delta = \operatorname{span}\{\varphi_i^{(\delta)} : i = 1,\dots,n_\delta\}$. We make use of the fact that each projector into the space V_δ can be represented as

$$\Pi_\delta u = \sum_{i=1}^{n_\delta} \langle \lambda_i^{(\delta)}(t), u(t)\rangle_t \varphi_i^{(\delta)},$$

where the functionals $\lambda_i^{(\delta)}$ are appropriately chosen dual basis functions, i.e., such that $\langle \lambda_i^{(\delta)}(t), \varphi_j^{(\delta)}(t)\rangle_t = \delta_{i,j}$. As Π_δ is linear, we immediately obtain that also the functionals $\lambda_i^{(\delta)}$ are linear.

So, we obtain

$$\widehat{\Pi}_1 u = \sum_{i=1}^{n_1} \langle \lambda_i^{(1)}(x), u(x,y)\rangle_x \varphi_i^{(1)}(x),$$

and

$$\begin{aligned}\widehat{\Pi}_2\widehat{\Pi}_1 u &= \sum_{j=1}^{n_2}\sum_{i=1}^{n_1} \langle \lambda_j^{(2)}(y), \langle \lambda_i^{(1)}(x), u(x,y)\rangle_x\rangle_y \varphi_i^{(1)}(x)\varphi_i^{(2)}(y) \\ &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \langle \lambda_i^{(1)}(x)\lambda_j^{(2)}(y), u(x,y)\rangle_{(x,y)} \varphi_i^{(1)}(x)\varphi_i^{(2)}(y).\end{aligned} \tag{2.7}$$

This term is completely symmetric in the spatial dimensions, so we obtain the first statement. We observe moreover that $\widehat{\Pi}_2\widehat{\Pi}_1 u$ is a linear combination of the basis functions, i.e., $\widehat{\Pi} := \widehat{\Pi}_2\widehat{\Pi}_1$ maps into $V_1 \otimes V_2$.

To show that $\widehat{\Pi}$ is a projector, we need to know that $\widehat{\Pi} u_h = u_h$ for all $u_h \in V_1 \otimes V_2$. Using linearity of $\widehat{\Pi}$, it suffices to show only

$$\widehat{\Pi}\varphi_{i,j} = \varphi_{i,j}$$

for all $\varphi_{i,j}(x,y) = \varphi_i^{(1)}(x)\varphi_i^{(2)}(y)$. By plugging into (2.7), we obtain

$$\begin{aligned}(\widehat{\Pi}\varphi_{i,j})(x,y) &= \sum_{k=1}^{n_1}\sum_{l=1}^{n_2}\langle\lambda_k^{(1)}(x)\langle\lambda_l^{(2)}(y),\varphi_i^{(1)}(x)\varphi_j^{(2)}(y)\rangle_y\rangle_x\varphi_k^{(1)}(x)\varphi_l^{(2)}(y)\\ &= \sum_{k=1}^{n_1}\sum_{l=1}^{n_2}\underbrace{\langle\lambda_k^{(1)}(x)\varphi_i^{(1)}(x)\rangle_x}_{=\delta_{i,k}}\underbrace{\langle\lambda_l^{(2)}(y),\varphi_j^{(2)}(y)\rangle_y}_{=\delta_{j,l}}\varphi_k^{(1)}(x)\varphi_k^{(2)}(y)\\ &= \varphi_i^{(1)}(x)\varphi_j^{(2)}(y) = \varphi_{i,j}(x,y),\end{aligned}$$

which was to show. □

Remark 2.11. *For the Schumaker interpolant, we have*

$$\langle\lambda_i^{(\delta)}(t),u(t)\rangle_t = \int_0^1 \psi_i^{(\delta)}(t)u(t)\,\mathrm{d}t$$

and the main part of the proof reads as follows:

$$\begin{aligned}\widehat{\Pi}_2\widehat{\Pi}_1 u &= \sum_{j=1}^{n_2}\sum_{i=1}^{n_1}\int_0^1\psi_j^{(2)}(y)\int_0^1\psi_i^{(1)}(x)u(x,y)\,\mathrm{d}x\,\mathrm{d}y\varphi_i^{(1)}(x)\varphi_i^{(2)}(y)\\ &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\int_{\widehat{\Omega}}\psi_i^{(1)}(x)\psi_j^{(2)}(y)u(x,y)\,\mathrm{d}(x,y)\varphi_i^{(1)}(x)\varphi_i^{(2)}(y).\end{aligned}$$

For the pointwise interpolants, we have

$$\langle\lambda_i^{(\delta)}(t),u(t)\rangle_t = u(x_i^{(\delta)})$$

and the main part of the proof reads as follows:

$$\widehat{\Pi}_2\widehat{\Pi}_1 u = \sum_{j=1}^{n_2}\sum_{i=1}^{n_1}u(x_i^{(1)},x_i^{(2)})\varphi_i^{(1)}(x)\varphi_i^{(2)}(y).$$

Theorem 2.12. *Provided that the error estimates*

$$\|(I-\Pi_1)u\|_{L_2(0,1)} \le h\Phi_1|u|_{H^1(0,1)} \qquad \textit{and} \qquad \|(I-\Pi_2)u\|_{L_2(0,1)} \le h\Phi_2|u|_{H^1(0,1)},$$

hold for all $u \in H^1(0,1)$ and that the stability estimate

$$\|\Pi_1 u\|_{L_2(0,1)} \le \Psi_1\|u\|_{L_2(0,1)} \tag{2.8}$$

holds for all $u \in L_2(0,1)$, we obtain

$$\|(I-\widehat{\Pi})u\|_{L_2(\widehat{\Omega})} \le \sqrt{2}h\max\{\Phi_1,\Psi_1\Phi_2\}|u|_{H^1(\widehat{\Omega})}.$$

for all $u \in C^0(\widehat{\Omega})$.

Proof. Let u be arbitrary but fixed. Observe that (2.8) implies

$$\|\Pi_1 u(\cdot,y)\|_{L_2(0,1)}^2 \le \Psi_1^2\|u(\cdot,y)\|_{L_2(0,1)}^2$$

and therefore also

$$\int_0^1 \|\Pi_1 u(\cdot,y)\|^2_{L_2(0,1)}\,\mathrm{d}y \le \int_0^1 \Psi_1^2 \|u(\cdot,y)\|^2_{L_2(0,1)}\,\mathrm{d}y$$

and further

$$\|\widehat{\Pi}_1 u\|^2_{L_2(\widehat{\Omega})} \le \Psi_1^2 \|u(\cdot,y)\|^2_{L_2(\widehat{\Omega})}.$$

We can do the same for the approximation error estimates and obtain

$$\|(I-\widehat{\Pi}_1)u\|_{L_2(\widehat{\Omega})} \le h\Phi_1 \|\tfrac{\partial}{\partial x}u\|_{L_2(\widehat{\Omega})} \qquad \text{and} \qquad \|(I-\widehat{\Pi}_2)u\|_{L_2(\widehat{\Omega})} \le h\Phi_2 \|\tfrac{\partial}{\partial y}u\|_{L_2(\widehat{\Omega})}.$$

We obtain using these estimates and the triangle inequality

$$\begin{aligned}
\|(I-\widehat{\Pi})u\|_{L_2(0,1)} &= \|(I-\widehat{\Pi}_1\widehat{\Pi}_2)u\|_{L_2(0,1)} \\
&\le \|(I-\widehat{\Pi}_1)u\|_{L_2(0,1)} + \|\widehat{\Pi}_1(I-\widehat{\Pi}_2)u\|_{L_2(0,1)} \\
&\le h\Phi_1 \|\tfrac{\partial}{\partial x}u\|_{L_2(\widehat{\Omega})} + \Psi_1 \|\widehat{\Pi}_1(I-\widehat{\Pi}_2)u\|_{L_2(0,1)} \\
&\le h\Phi_1 \|\tfrac{\partial}{\partial x}u\|_{L_2(\widehat{\Omega})} + \Psi_1 h\Phi_2 \|\tfrac{\partial}{\partial y}u\|_{L_2(\widehat{\Omega})} \\
&\le \sqrt{2}\max\{\Phi_1\Psi_1,\Phi_2\} h \left(\|\tfrac{\partial}{\partial x}u\|^2_{L_2(\widehat{\Omega})} + \|\tfrac{\partial}{\partial y}u\|^2_{L_2(\widehat{\Omega})} \right)^{1/2} \\
&= \sqrt{2}\max\{\Phi_1\Psi_1,\Phi_2\} h |u|_{H^1(\widehat{\Omega})},
\end{aligned}$$

which was to show. □

Remark 2.13. *Note that we cannot assume the stability estimate* (2.8) *to be satisfied in general. If we are only interested in approximation error estimates, we can choose the projectors* Π_i *to be the* L_2*-orthogonal projectors, as we have*

$$\|(I-\Pi_i)u\|_{L_2(0,1)} = \inf_{u_h \in V_1} \|u-u_h\|_{L_2(0,1)} \le \|(I-\widetilde{\Pi}_i)u\|_{L_2(0,1)} \le \cdots$$

for $\widetilde{\Pi}_i$ *being any other projector.*

As the L_2*-orthogonal projector satisfies* (2.8) *by construction, we obtain the desired approximation error estimates.*

Theorem 2.14. *Provided that the approximation error estimates*

$$\|(I-\Pi_1)u\|_{L^2(0,1)} \le \Phi_{0,1} h |u|_{H^1(0,1)}, \qquad |(I-\Pi_1)u|_{H^1(0,1)} \le \Phi_{1,1} h |u|_{H^2(0,1)},$$

$$\|(I-\Pi_2)u\|_{L^2(0,1)} \le \Phi_{0,2} h |u|_{H^1(0,1)}, \qquad |(I-\Pi_2)u|_{H^1(0,1)} \le \Phi_{1,2} h |u|_{H^2(0,1)},$$

and the stability estimates

$$|\Pi_1 u|_{H^1(0,1)} \le \Psi_1 |u|_{H^1(0,1)}, \qquad \textit{and} \qquad |\Pi_2 u|_{H^1(0,1)} \le \Psi_2 |u|_{H^1(0,1)}$$

hold, we obtain

$$\|(I-\widehat{\Pi})u\|_{L_1(0,1)} \le \sqrt{2} h \max\{\Phi_{1,1}, \Phi_{0,2}\Psi_1, \Phi_{1,2}, \Phi_{0,1}\Psi_2\} |u|_{H^2(0,1)}$$

for all $u \in C^0(\widehat{\Omega})$.

The proof is left as an exercise to the reader.

We had been very careful with defining the projector $\widehat{\Pi}$ as we have required u to be in $C^0(\widehat{\Omega})$. Now, having approximation error estimates, we can extend its definition using a *density argument*. Assume that the univariate projectors Π_1 and Π_2 satisfy the assumptions of Theorem 2.12, so we have

$$\|I - \Pi u\|_{L_2(\widehat{\Omega})} \leq ch|u|_{H^1(\widehat{\Omega})}.$$

Note that $C^0(\widehat{\Omega})$ is dense in $H^1(\widehat{\Omega})$. So, for each $\epsilon > 0$ and for each $u \in H^1(\Omega)$, we can find a function $u^\epsilon \in C^0(\widehat{\Omega})$ such that

$$\|u - u^\epsilon\|_{H^1(\widehat{\Omega})} \leq \epsilon \|u\|_{H^1(\widehat{\Omega})}.$$

So, we can set up a functional A^ϵ, which assigns such a function u^ϵ to each given u.
Now, we can construct projectors $\Pi^\epsilon := \Pi A^\epsilon$ and observe using the triangle inequality

$$\begin{aligned}\|u - \Pi^\epsilon u\|_{L_2(\widehat{\Omega})} &\leq \|u - A^\epsilon u\|_{L_2(\widehat{\Omega})} + \|(I - \Pi)A^\epsilon u\|_{L_2(\widehat{\Omega})} \\ &\leq \epsilon \|u\|_{H^1(\widehat{\Omega})} + ch|A^\epsilon u|_{H^1(\widehat{\Omega})} \\ &\leq \epsilon(1 + ch)\|u\|_{H^1(\widehat{\Omega})} + ch|u|_{H^1(\widehat{\Omega})}\end{aligned}$$

and an inverse inequality (see Chapter 4) further

$$\|\Pi^\epsilon u\|_{H^1(\widehat{\Omega})} \leq \Upsilon h^{-1}\|\Pi^\epsilon u\|_{L_2(\widehat{\Omega})} \leq \Upsilon h^{-1}(\|u\|_{L_2(\widehat{\Omega})} + \|(I - \Pi^\epsilon)u\|_{L_2(\widehat{\Omega})}) \leq (h^{-1} + c)\Upsilon \|u\|_{H^1(\widehat{\Omega})}.$$

Because the Π^ϵ are uniformly bounded in H^1, we can take the limit $\epsilon \to 0$ and obtain an operator $\Pi^0 : H^1(\widehat{\Omega}) \to \widehat{V}$ with

$$\|u - \Pi^0 u\|_{L_2(\widehat{\Omega})} \leq ch|u|_{H^1(\widehat{\Omega})}.$$

2.2.2 Approximation in the physical domain

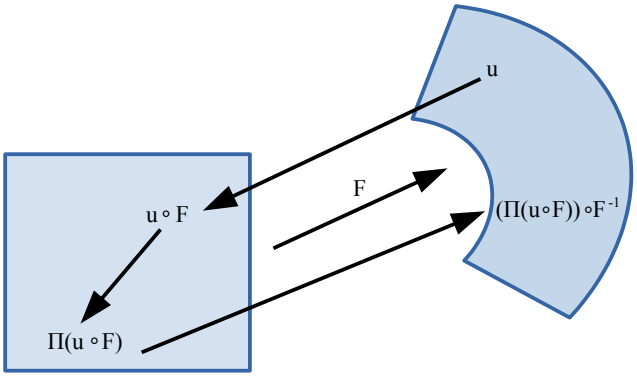


Figure 2.4: Approximation on the physical domain

Let $\widehat{\Pi} : H^1(\widehat{\Omega}) \to \widehat{V}_h$ be a projector on the parameter domain, where $\widehat{V}_h$ is some discretization, e.g., $\widehat{V}_h = \widehat{S}_{p,\Xi}(\widehat{\Omega})$.

Let $u \in H^1(\Omega)$ be arbitrary but fixed and define $\widehat{u} := u \circ F$.

Then, we compute as follows:

$$\|u - (\widehat{\Pi}(u \circ F)) \circ F^{-1}\|_{L_2(\Omega)} \leq \Psi_1 \|\widehat{u} - \widehat{\Pi}\widehat{u}\|_{L_2(\widehat{\Omega})} \leq \Psi_1 \Psi_2 |\widehat{u}|_{H^1(\widehat{\Omega})} \leq \Psi_1 \Psi_2 \Psi_3 |u|_{H^1(\widehat{\Omega})},$$

where

$$\Psi_1 := \sup_{u \in L_2(\Omega)} \frac{\|u\|_{L_2(\Omega)}}{\|u \circ F\|_{L_2(\widehat{\Omega})}}, \qquad \Psi_3 := \sup_{u \in H^1(\Omega)} \frac{|u \circ F|_{H^1(\widehat{\Omega})}}{|u|_{H^1(\Omega)}},$$

and

$$\Psi_2 := \sup_{\widehat{u} \in H^1(\widehat{\Omega})} \frac{\|\widehat{u} - \Pi\widehat{u}\|_{L_2(\widehat{\Omega})}}{|\widehat{u}|_{H^1(\widehat{\Omega})}}.$$

An upper bound for Ψ_2 is obtained by any *approximation error estimate* for the parameter domain, like the results shown in the last subsection.

In the remainder of this subsection, we discuss how to estimate Ψ_1 and Ψ_2 and similar terms.

Theorem 2.15. *Assume that $F : \widehat{\Omega} = (0,1)^d \to \Omega \subset \mathbb{R}^d$. Then we have*

$$\|u\|_{L_2(\Omega)} \lesssim \|\nabla F\|^{d/2}_{L_\infty(\widehat{\Omega})} \|u \circ F\|_{L_2(\widehat{\Omega})} \qquad \text{and} \qquad \|u \circ F\|_{L_2(\widehat{\Omega})} \lesssim \|(\nabla F)^{-1}\|^{d/2}_{L_\infty(\widehat{\Omega})} \|u\|_{L_2(\Omega)}$$

for all $u \in L^2(\Omega)$, where the L_∞-norm of a matrix $A(x)$ with coefficients $a_{i,j}(x)$ is given by $\sup_x \max_{i,j} |a_{i,j}(x)|$.

Proof. Let u be arbitrary but fixed and choose $\widehat{u} := u \circ F$. The substitution rule yields

$$\begin{aligned} \|u\|^2_{L_2(\Omega)} &= \int_\Omega u(x)^2 \,\mathrm{d}x = \int_{\widehat{\Omega}} \widehat{u}(\widehat{x})^2 |\det \nabla F(\widehat{x})| \,\mathrm{d}\widehat{x} \\ &\leq \left(\sup_{x \in \widehat{\Omega}} |\det \nabla F(\widehat{x})| \right) \int_{\widehat{\Omega}} \widehat{u}(\widehat{x})^2 \,\mathrm{d}\widehat{x}, \end{aligned}$$

where ∇F is the Jacobi matrix, which has size $d \times d$. As the determinant of a $d \times d$ matrix is bounded by d times its largest entry, we obtain

$$\left(\sup_{x \in \widehat{\Omega}} |\det \nabla F(\widehat{x})| \right) \leq \|\nabla F\|^d_{L_\infty(\widehat{\Omega})}$$

and, therefore, the desired result. □

Theorem 2.16. *Assume that $F : \widehat{\Omega} = (0,1)^d \to \Omega \subset \mathbb{R}^d$. Then we have*

$$|u|_{H^1(\Omega)} \leq c \|\nabla F\|^{d/2}_{L_\infty(\widehat{\Omega})} \|(\nabla F)^{-1}\|_{L_\infty(\widehat{\Omega})} |u \circ F|_{H^1(\widehat{\Omega})}$$

and

$$|u \circ F|_{H^1(\widehat{\Omega})} \leq c \|\nabla F\|_{L_\infty(\widehat{\Omega})} \|(\nabla F)^{-1}\|^{d/2}_{L_\infty(\widehat{\Omega})} |u|_{H^1(\Omega)}$$

for all $u \in H^1(\Omega)$, where the constant c only depends on d.

Proof. Let u be arbitrary but fixed and choose $\widehat{u} := u \circ F$. The substitution rule yields

$$\begin{aligned} |u|^2_{H^1(\Omega)} &= \int_\Omega (\nabla_x u)(x) \cdot (\nabla_x u)(x) \,\mathrm{d}x \\ &= \int_{\widehat{\Omega}} (\nabla_x u)(F(\widehat{x})) \cdot (\nabla_x u)(F(\widehat{x})) |\det \nabla F(\widehat{x})| \,\mathrm{d}\widehat{x}. \end{aligned}$$

Now, observe that the chain rule yields

$$(\nabla_{\widehat{x}}(u \circ F))(\widehat{x}) = \nabla_{\widehat{x}} F(\widehat{x}) \; (\nabla_x u)(F(\widehat{x}))$$

and, therefore, also

$$(\nabla_x u)(F(\widehat{x})) = (\nabla_{\widehat{x}} F(\widehat{x}))^{-1} (\nabla_{\widehat{x}}(u \circ F))(\widehat{x}).$$

By plugging this into the above equation, we obtain

$$\begin{aligned} |u|^2_{H^1(\Omega)} &= \int_{\widehat{\Omega}} [(\nabla_{\widehat{x}}(u \circ F))(\widehat{x})]^T (\nabla_{\widehat{x}} F(\widehat{x}))^{-T} (\nabla_{\widehat{x}} F(\widehat{x}))^{-1} (\nabla_{\widehat{x}}(u \circ F))(\widehat{x}) |\det \nabla F(\widehat{x})| \,\mathrm{d}\widehat{x} \\ &\leq c \|\nabla F\|^d_{L_\infty(\widehat{\Omega})} \|(\nabla F)^{-1}\|^2_{L_\infty(\widehat{\Omega})} \int_{\widehat{\Omega}} [(\nabla_{\widehat{x}}(u \circ F))(\widehat{x})]^T (\nabla_{\widehat{x}}(u \circ F))(\widehat{x}) \,\mathrm{d}\widehat{x} \\ &= c \|\nabla F\|^d_{L_\infty(\widehat{\Omega})} \|(\nabla F)^{-1}\|^2_{L_\infty(\widehat{\Omega})} |\widehat{u}|^2_{H^1(\widehat{\Omega})}, \end{aligned}$$

which was to show.

The other direction is analogous. □

Some comments:

- Results similar to Theorem 2.16 for norms with higher Sobolev indices are possible if the geometry is sufficiently smooth. So, we estimates of $|u|_{H^r(\Omega)}$, we need that that the L^∞-norm of the derivatives of the geometry function F up to order r is bounded.
- Similar results are possible for mappings $F : \widehat{\Omega} = (0,1)^d \to \Omega \subset \mathbb{R}^m$ with $m > d$.
- If the geometry transformation is not sufficiently smooth, one can work with broken Sobolev spaces or bent Sobolev spaces.

2.3 Literature

Subsection 2.1.1 is standard. For the results of Subsection 2.1.2, literature is not known. For more information on the Schumaker quasi interpolants, we refer to [5]. We moreover refer to [6], where approximation error estimates for IgA have been discussed.

Chapter 3

p-robust approximation error estimates

3.1 Estimates for polynomials

Consider the Legendre polynomials, which are given by the following formula.

$$\begin{aligned} L_0(x) &= 1 \\ L_1(x) &= x \\ L_n(x) &= \frac{2n-1}{n} x L_{n-1}(x) - \frac{n-1}{n} L_{n-2}(x) \qquad \text{for} \qquad n = 2, 3, \dots . \end{aligned}$$

One typically considers them on the interval $(-1, 1)$.

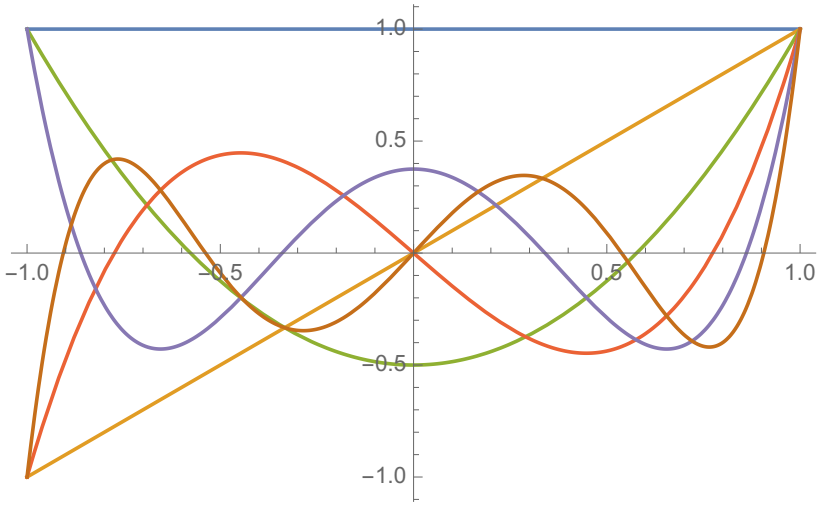


Figure 3.1: The first six Legendre-polynomials: $L_0(x) = 1, L_1(x) = x, L_2(x) = \frac{1}{2}(-1 + 3x^2), L_3(x) = \frac{1}{2}(-3x + 5x^3), L_4(x) = \frac{1}{8}(3 - 30x^2 + 35x^4), L_5(x) = \frac{1}{8}(15x - 70x^3 + 63x^5)$

Some properties:

- *Degree:* $\deg L_n = n$.

- *Boundary conditions:*
  - $L_n(1) = 1$ for all $n = 0, 1, 2, \ldots$.
  - $L_n(-1) = (-1)^n$ for all $n = 0, 1, 2, \ldots$.
- *Parity:* For even n, L_n is even, for odd n, L_n is odd.
  - $L_n(-x) = (-1)^n L_n(x)$ for all $n = 0, 1, 2, \ldots$.
- *Basis:* $\mathbb{P}_n = \operatorname{span}\{L_0, L_1, \ldots, L_n\}$.
- *Orthogonality:*
$$\int_{-1}^{1} L_i(x) L_j(x) \,\mathrm{d}x = \frac{2}{2i+1} \delta_{i,j}$$
- A simple consequence is
$$\int_{-1}^{1} L_i(x) v(x) \,\mathrm{d}x = 0 \quad \text{for all} \quad v \in \mathbb{P}_p \text{ with } p < i$$
since v can be expressed as a linear combination of the Legendre polynomials $L_0, \cdots, L_p$.
- *Orthogonality of derivatives and primes:* For $k \in \mathbb{Z}$ and $i, j \geq \max\{0, k\}$, we have
$$\int_{-1}^{1} (1-x^2)^n L_i^{(k)}(x) L_j^{(k)}(x) \,\mathrm{d}x = \frac{2}{2i+1} \frac{(i+k)!}{(i-k)!} \delta_{i,j}, \tag{3.1}$$
where, for $k \geq 0$, $u^{(k)}$ is the kth derivative, and for $k < 0$, $u^{(k)}(x) := \int_{-1}^{x} u^{(k+1)}(y) \,\mathrm{d}y$ is the prime.

  (3.1) is to be understood that the series is convergent if and only if the integral on the left-hand side is finite. A proof is for $k > 0$ is given in [7, Lemma 3.10] and for $k < 0$ is given in [8, Corollary 1].

Theorem 3.1 (Weierstrass approximation theorem)**.** *For every $\epsilon > 0$ and every $u \in C^0(-1, 1)$, there is degree p and a polynomial $u_\epsilon \in \mathbb{P}_p$ such that*
$$\|u - u_\epsilon\|_{L_\infty(-1,1)} \leq \epsilon.$$

We can relax this result to the L_2-norm and use density of L_2 in C^0 to obtain that, for every $\epsilon > 0$ and every $u \in L_2(-1, 1)$, there is degree p such that
$$\underbrace{\inf_{u_\epsilon \in \mathbb{P}_p} \|u - u_\epsilon\|_{L_2(-1,1)}}_{= \|u - \Pi_p u\|_{L_2(-1,1)}} \leq \epsilon,$$
where Π_p is the $L_2(-1, 1)$-orthogonal projection into $\mathbb{P}_p$. This yields $\lim_{p\to\infty} \|u - \Pi_p u\|_{L_2(-1,1)} = 0$ and therefore
$$u = \lim_{p\to\infty} \Pi_p u \quad \text{in the } L_2\text{-sense.} \tag{3.2}$$

Since the Legendre polynomials form an orthogonal basis, they can be used to represent the orthogonal projections into a spline space. We use the following ansatz:
$$(\Pi_p u)(x) = \sum_{i=0}^{p} u_i L_i(x).$$

As Π_p is the L_2-orthogonal projection, we have

$$0 = (u - \Pi_p u, v)_{L_2(-1,1)}$$

for all test functions $v \in \mathbb{P}_p(-1,1)$. Using the particular choice $v := L_j$ with $j = 0, \dots, p$, we obtain

$$\begin{aligned} 0 = (u - \Pi_p u, L_j)_{L_2(-1,1)} &= (u, L_j)_{L_2(-1,1)} - (\sum_{i=0}^{p} u_i L_i, L_j)_{L_2(-1,1)} \\ &= (u, L_j)_{L_2(-1,1)} - \sum_{i=0}^{p} u_i \underbrace{(L_i, L_j)_{L_2(-1,1)}}_{= \frac{2}{2j+1}\delta_{i,j}} = (u, L_j)_{L_2(-1,1)} - \frac{2}{2j+1} u_j, \end{aligned}$$

i.e., $u_j = \frac{2j+1}{2}(u, L_j)_{L_2(-1,1)}$. This yields

$$\Pi_p u = \sum_{i=0}^{p} \frac{2i+1}{2} (u, L_i)_{L_2(-1,1)} L_i.$$

Now, in combination with (3.2), we obtain

$$u = \sum_{i=0}^{\infty} \frac{2i+1}{2} (u, L_i)_{L_2(-1,1)} L_i,$$

cf. [7, (3.3.3) and (3.3.5)].
Using this result, we obtain the following approximation error estimate.

Theorem 3.2. *For any $u \in L_2(-1,1)$, we have*

$$\|u - \Pi_p u\|^2_{L_2(-1,1)} = \sum_{i=p+1}^{\infty} \frac{2i+1}{2} (u, L_i)^2_{L_2(-1,1)}.$$

Proof. We have

$$u - \Pi_p u = \sum_{i=0}^{\infty} \frac{2i+1}{2} (u, L_i)_{L_2} L_i - \sum_{i=0}^{p} \frac{2i+1}{2} (u, L_i)_{L_2} L_i = \sum_{i=p+1}^{\infty} \frac{2i+1}{2} (u, L_i)_{L_2} L_i$$

and

$$(u - \Pi_p u, u - \Pi_p u)_{L_2(-1,1)} = \sum_{i=p+1}^{\infty} \sum_{j=p+1}^{\infty} \frac{2i+1}{2} (u, L_i)_{L_2} \frac{2j+1}{2} (u, L_j)_{L_2} \underbrace{(L_i, L_j)_{L_2}}_{= \frac{2}{2i+1}\delta_{i,j}},$$

which immediately yields the desired result. □

Assuming $0 \le k \le p+1$, the combination of Theorem 3.2 and (3.1) yields:

$$\begin{aligned}
\|u - \Pi_p u\|^2_{L_2(-1,1)} &= \sum_{i=p+1}^{\infty} \frac{2i+1}{2} (u, L_i)^2_{L_2(-1,1)} \\
&\le \left(\sup_{i=p+1,\dots,\infty} \frac{(i-k)!}{(i+k)!} \right) \left(\sum_{i=p+1}^{\infty} \frac{2i+1}{2} (u, L_i)^2_{L_2(-1,1)} \frac{(i+k)!}{(i-k)!} \right) \\
&\le \left(\sup_{i=p+1,\dots,\infty} \frac{(i-k)!}{(i+k)!} \right) \left(\sum_{i=k}^{\infty} \frac{2i+1}{2} (u, L_i)^2_{L_2(-1,1)} \frac{(i+k)!}{(i-k)!} \right) \\
&= \underbrace{\left(\sup_{i=p+1,\dots,\infty} \frac{(i-k)!}{(i+k)!} \right)}_{= \frac{(p+1-k)!}{(p+1+k)!}} \underbrace{\int_{-1}^{1} |\tfrac{\partial^k}{\partial x^k} u(x)|^2 (1-x^2)\,\mathrm{d}x}_{\le |u|^2_{H^k(-1,1)}} .
\end{aligned}$$

Stirling's formula, cf. [7, p. 72], yields

$$\frac{(p+1-k)!}{(p+1+k)!} \le \left(\frac{\mathrm{e}}{2}\right)^{2k} (p+1)^{-2k}.$$

This proves the following theorem.

Theorem 3.3. *Let $0 \le k \le p+1$. Then for all $u \in H^k(-1,1)$, we have*

$$\|u - \Pi_p u\|_{L_2(-1,1)} \le \left(\frac{\mathrm{e}}{2}\right)^{k} (p+1)^{-k} |u|_{H^k(-1,1)}.$$

If we go to another interval (a,b), we obtain using a simple scaling argument

$$\begin{aligned}
\|u - \Pi_p u\|_{L_2(a,b)} &\le \left(\frac{b-a}{2}\right)^{k} \left(\frac{\mathrm{e}}{2}\right)^{k} (p+1)^{-k} |u|_{H^k(a,b)} \\
&\le (b-a)^k (p+1)^{-k} |u|_{H^k(a,b)}. \qquad (3.3)
\end{aligned}$$

Can we extend this to splines by applying this projector to all elements $I_i = (\zeta_i, \zeta_{i+1})$? If Π_p was interpolatory on the boundary, we could. This is what we discuss in the next section.

3.2 Estimates for splines with low smoothness

For simplicity, we assume $k_2 = k_3 = \cdots = k_{n-1} =: k$. Throughout this section, we assume

$$k \le \frac{p+1}{2}.$$

Under this limitation, we introduce an *Hermite type operator*

$$\Pi_{p,\Xi} : H^{k+1}(0,1) \to S_{p,\Xi}(0,1)$$

as follows. First define for each element $I_i = (\zeta_i, \zeta_{i+1})$ with $i = 1, \dots, N-1$ a polynomial $u_i \in \mathbb{P}_p$ such that

$$u_i^{(j)}(\zeta_i) = u^{(j)}(\zeta_i) \qquad \text{for all} \quad j = 0, \dots, k, \qquad (3.4)$$

$$u_i^{(j)}(\zeta_{i+1}) = u^{(j)}(\zeta_{i+1}) \qquad \text{for all} \quad j = 0, \dots, k, \qquad (3.5)$$

$$u_i^{(k+1)}|_{I_i} = P^{I_i}_{p-k-1}(u^{(k+1)}) \qquad (3.6)$$

where $u^{(i)}$ denotes the ith derivative and $P_q^{I_i}$ is the $L_2(I_i)$-orthogonal projection into $\mathbb{P}_q$. The overall function is defined via

$$(\Pi_{p,\Xi} u)|_{I_i} := u_i.$$

First we observe that

$$\Pi_{p,\Xi} u \in S_{p,\Xi}(0,1)$$

holds due to the Hermite type construction (3.4), (3.5).

The next step is to show that there is a polynomial such that (3.4), (3.5) and (3.6) are satisfied:

- (3.4) contains $k+1$ conditions,
- (3.5) contains $k+1$ conditions,
- (3.6) contains $p-k$ conditions,

while the function space $\mathbb{P}_p$ has $p+1$ dimensions. So, the problem is overdetermined.

Observe that there are polynomials in $\mathbb{P}_p$ such that both (3.4) and (3.6) are satisfied. (This is left as an exercise to the reader. Note that it is not sufficient to show that the number of conditions is $\leq$ than the number of degrees of freedom.)

The next lemma shows that the combination of (3.4) and (3.6) implies (3.5).

Lemma 3.4. *Let $u \in H^{k+1}(-1,1)$ and let $v \in \mathbb{P}_p$ such that*

$$\begin{aligned} v^{(j)}(-1) &= u^{(j)}(-1) \qquad \text{for all} \quad j = 0, \dots, k, \\ v^{(k+1)} &= P_{p-k-1}^{(-1,1)}(u^{(k+1)}), \end{aligned}$$

where $P_{p-k-1}^{(-1,1)}$ is the $L_2(-1,1)$-orthogonal projection into $\mathbb{P}_p$. Then

$$v^{(j)}(1) = u^{(j)}(1) \qquad \text{holds for all} \quad j = 0, \dots, k.$$

Proof. The Taylor expansion of any function $\varphi \in C^\ell(-1,1)$ is

$$\varphi(x) = \sum_{i=0}^{\ell} \frac{(x+1)^i}{i!} \varphi^{(i)}(-1) + \int_{-1}^{x} \frac{(x-y)^{\ell+1}}{(\ell-1)!} \varphi^{(\ell)}(y)\,\mathrm{d}y.$$

Let $j \in \{0, \dots, k\}$ be arbitrary but fixed. The Taylor expansions for $\ell := k+1-j$ are

$$\begin{aligned} u^{(j)}(x) &= \sum_{i=0}^{k+1-j} \frac{(x+1)^i}{i!} u^{(j+i)}(-1) + \int_{-1}^{x} \frac{(x-y)^{k-j+2}}{(\ell-1)!} u^{(k+1)}(y)\,\mathrm{d}y \\ &= \sum_{i=j}^{k+1} \frac{(x+1)^{i-j}}{(i-j)!} u^{(i)}(-1) + \int_{-1}^{x} \frac{(x-y)^{k-j+2}}{(\ell-1)!} u^{(k+1)}(y)\,\mathrm{d}y \end{aligned}$$

and

$$v^{(j)}(x) = \sum_{i=j}^{k+1} \frac{(x+1)^{i-j}}{(i-j)!} v^{(i)}(-1) + \int_{-1}^{x} \frac{(x-y)^{k-j+2}}{(\ell-1)!} v^{(k+1)}(y)\,\mathrm{d}y.$$

Taking $x = 1$ and subtracting the above expressions, we obtain

$$\begin{aligned} &u^{(j)}(x) - v^{(j)}(x) \\ &= \underbrace{\sum_{i=j}^{k+1} \frac{2^{i-j}}{(i-j)!}(u^{(i)} - v^{(i)})(-1)}_{=0 \text{ due to assumption}} + \underbrace{\int_{-1}^{1} \frac{(1-y)^{k-j+2}}{(\ell-1)!}(u^{(k+1)} - v^{(k+1)})(y)\,\mathrm{d}y}_{=:(*)} \end{aligned}$$

Note that $v^{(k+1)} = P^{(-1,1)}_{p-k-1}(u^{(k+1)})$, implies $u^{(k+1)} - v^{(k+1)}$ to be orthogonal to any polynomial of degree $p-k-1$, we have that also $(*) = 0$. This shows $u^{(j)}(x) = v^{(j)}(x)$. □

Using approximation results on Legendre polynomials, we obtain the following result.

Theorem 3.5. *Provided $0 \le l \le k-1 \le r \le p+1$ and $k \le \frac{p+1}{2}$, we obtain for all $u \in H^r(0,1)$*

$$|u - \Pi_{p,\Xi} u|_{H^l(0,1)} \le h^{r-l}(p-k)^{l-r}|u|_{H^r(0,1)}.$$

Proof. For $l = k+1$, this result directly follows from (3.3) and (3.6). For $l < k+1$, similar arguments are possible. The details are given in [8, Theorem 2]. □

Some comments:

- Numerical experiments indicate sharpness of the results.
- The case of highest smoothness ($k = p-1$) is *not* covered.
- The extension to the *multivariate case* and to the *physical domain* can be done as outlined in Section 2.2.

3.3 Estimates for splines with maximum smoothness

The results from the last section do not cover case $k = p-1$, which is the most interesting case for IgA. If we would plug this choice into Theorem 3.5, we *would* obtain

$$|u - \Pi_{p,\Xi} u|_{H^l(0,1)} \le h^{r-l}|u|_{H^r(0,1)}.$$

The goal of this section is to give such a statement. This does not come for free: the techniques of this section are only applicable for equidistant grids.

Throughout the next three subsections, we prove the following theorem.

Theorem 3.6. *Provided $p \in \mathbb{N}$, $h = 1/n$, $n \in \mathbb{N}$, $hp < |(0,1)| = 1$, we have*

$$\inf_{u_h \in S_{p,h}(0,1)} \|u - u_h\|_{L_2(0,1)} \le \sqrt{2}h|u|_{H^1(0,1)}$$

for all $u \in H^1(0,1)$.

Following the arguments of Theorem 2.6 and Corollary 2.7, we can extend this result to higher Sobolev indices. In this case, we obtain an estimate of the form $\le (\sqrt{2}h)^{r-l}$.

Following the arguments of Section 2.2, we can extend our results to the multivariate case. In this case, the constants depend on d. Following the arguments of Section 2.2.2, we can extend the result to the physical domain. In this case, the constants also depend on the geometry mapping.

3.3.1 A proof for the periodic case

First we have to discuss what a *periodic function* is. We say that a function $u \in H^1(0,1)$ is periodic if we can extend the function as

$$w(x) := u\big(x - \lfloor x \rfloor\big)$$

such that $w \in H^1(0,2)$. Note that w is obviously in $H^1(0,1)$ and in $H^1(1,2)$. The only question is if it is in H^1 at the point 1. From standard trace estimates, we know that H^1-functions are continuous, but do not show more continuity than that.

So, we have

$$H^{1,per}(0,1) = \{u \in H^1(0,1) \ : \ u(0) = u(1)\}.$$

Analogously, we have

$$\begin{aligned}
L_2^{per}(0,1) &= L_2(0,1),\\
H^{r,per}(0,1) &= \{u \in H^r(0,1) \ : \ \tfrac{d}{dx^s}u(0) = \tfrac{d}{dx^s}u(1) \text{ for } s = 0,\dots,r-1\},\\
C^{r,per}(0,1) &= \{u \in C^r(0,1) \ : \ \tfrac{d}{dx^s}u(0) = \tfrac{d}{dx^s}u(1) \text{ for } s = 0,\dots,r\},\\
S_{p,h}^{per}(0,1) &= \{u \in S_{p,h}(0,1) \ : \ \tfrac{d}{dx^s}u(0) = \tfrac{d}{dx^s}u(1) \text{ for } s = 0,\dots,p-1\}.
\end{aligned}$$

The desired estimate – for the periodic case – reads as follows.

$$\inf_{u_h \in S_{p,h}^{per}(0,1)} \|u - u_h\|_{L_2(0,1)} \le \sqrt{2}h|u|_{H^1(0,1)}, \quad \text{for all} \quad u \in H^{1,per}(0,1).$$

We prove this using a hierarchical argument. Let $\Pi_{p,h}^{per}$ be the H^1-orthogonal projection $H^{1,per}(0,1) \to S_{p,h}^{per}(0,1)$.

Then, we have using the triangle inequality and using $\Pi_{p,\eta}^{per}\Pi_{p,2^{-1}\eta}^{per} = \Pi_{p,\eta}^{per}$ that

$$\begin{aligned}
\|u - \Pi_{p,h}^{per}u\|_{L_2(0,1)} &\le \|u - \Pi_{p,2^{-N}h}^{per}u\|_{L_2(0,1)} + \|\Pi_{p,2^{-N}h}^{per}u - \Pi_{p,2^{-N+1}h}^{per}u\|_{L_2(0,1)}\\
&\quad + \|\Pi_{p,2^{-N+1}h}^{per}u - \Pi_{p,2^{-N+2}h}^{per}u\|_{L_2(0,1)} + \cdots + \|\Pi_{p,2^{-1}h}^{per}u - \Pi_{p,h}^{per}u\|_{L_2(0,1)}\\
&= \|u - \Pi_{p,2^{-N}h}^{per}u\|_{L_2(0,1)} + \|(I - \Pi_{p,2^{-N}h}^{per})\Pi_{p,2^{-N+1}h}^{per}u\|_{L_2(0,1)}\\
&\quad + \|(I - \Pi_{p,2^{-N+1}h}^{per})\Pi_{p,2^{-N+2}h}^{per}u\|_{L_2(0,1)} + \cdots + \|(I - \Pi_{p,h}^{per})\Pi_{p,2^{-1}h}^{per}u\|_{L_2(0,1)}.
\end{aligned}$$

Now, we use the following estimates.

- There is a non-p-robust error estimate:

  $$\|u - \Pi_{p,\eta}^{per}u\|_{L_2(0,1)} \le c(p)\eta|u|_{H^1(0,1)} \quad \text{for all} \quad u \in H^{1,per}(0,1).$$

  Note that this is just a slight extension what we know from Chapter 2. So, we assume that to be true.

- We assume that there is a p robust error estimate for two consecutive grids:

  $$\|(I - \Pi_{p,\eta}^{per})u_{\eta/2}\|_{L_2(0,1)} \le \tfrac{1}{\sqrt{2}}\eta|u_{\eta/2}|_{H^1(0,1)} \quad \text{for all} \quad u_{\eta/2} \in S_{p,\eta/2}^{per}(0,1). \tag{3.7}$$

  We will show this estimate below.

- The H^1-orthogonal projector $\Pi^{per}_{p,h}$ is obviously *stable* in H^1:

$$|\Pi^{per}_{p,h} u|_{H^1(0,1)} \leq |u|_{H^1(0,1)}.$$

Using these three estimates, we obtain

$$\begin{aligned}
\|u - \Pi^{per}_{p,h} u\|_{L_2(0,1)} &\leq c(p) 2^{-N} h |u|_{H^1(0,1)} + \tfrac{1}{\sqrt{2}} 2^{-N} h |\Pi^{per}_{p,2^{-N+1}h} u|_{H^1(0,1)} \\
&\quad + \tfrac{1}{\sqrt{2}} 2^{-N+1} h |\Pi^{per}_{p,2^{-N+2}h} u|_{H^1(0,1)} + \cdots + \tfrac{1}{\sqrt{2}} h |\Pi^{per}_{p,2^{-1}h} u|_{H^1(0,1)} \\
&\leq c(p) 2^{-N} h |u|_{H^1(0,1)} + \tfrac{1}{\sqrt{2}} 2^{-N} h |u|_{H^1(0,1)} \\
&\quad + \tfrac{1}{\sqrt{2}} 2^{-N+1} h |u|_{H^1(0,1)} + \cdots + \tfrac{1}{\sqrt{2}} h |u|_{H^1(0,1)} \\
&= \big(c(p) 2^{-N} + \tfrac{1}{\sqrt{2}} \sum_{i=0}^{N} 2^{-i}\big) h |u|_{H^1(0,1)} \\
&\leq \big(c(p) 2^{-N} + \tfrac{1}{\sqrt{2}} \sum_{i=0}^{\infty} 2^{-i}\big) h |u|_{H^1(0,1)} \\
&= \big(c(p) 2^{-N} + \sqrt{2}\big) h |u|_{H^1(0,1)}.
\end{aligned}$$

Now, we obtain the desired result for $N \to \infty$.

The next step is to show (3.7). Our idea is to rewrite that in *matrix-vector formulation*. To do so, we need a basis. We observe that

- statements, like the statement (3.7) are independent of the chosen basis,
- the B-spline basis is not the only basis of a spline space.

Note that all the B-splines splines based on p-open knot vectors on equidistant grids in the interior are just shifts of each other, but the first p functions and the last p functions look differently, cf. Figure 3.2.

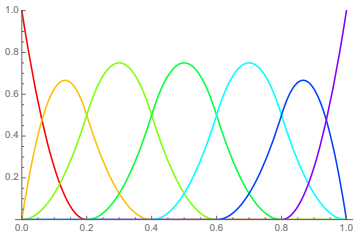


Figure 3.2: B-Splines of degree 2

An alternative construction of a spline basis for an equidistant grid is to take just shifts of these spline functions in the interior and restrict them always to $(0,1)$. Such splines are known as *cardinal splines* $(C_{i,p,h})_{i=1}^{n+p}$, cf. Figure 3.3. As for the B-splines, we have again $n+p$ basis functions and a basis of the spline space.

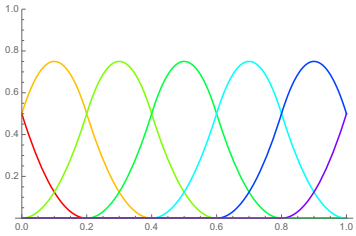


Figure 3.3: Cardinal splines of degree 2

Based on these cardinal splines, we can easily set up a basis for $S^{per}_{p,h}(0,1)$:

- The cardinal splines $C_{p+1,p,h},\dots,C_{n,p,h}$ whose support is in $[0,1]$ are already in $S^{per}_{p,h}$ and are taken unchanged into our new basis:

$$\widehat{C}_{i,p,h} := C_{i,p,h} \quad \text{for} \quad i = p+1,\dots,n.$$

- From the remaining cardinal splines we group always those two functions together such that their sum is in $S^{per}_{p,h}$. In Figure 3.3, this would be blue+red and purple+orange. We add always those sums to the new basis:

$$\widehat{C}_{i,p,h} := C_{i,p,h} + C_{i+n,p,h} \quad \text{for} \quad i = 1,\dots,p.$$

By this construction, we get a basis $(\widehat{C}_{i,p,h})_{i=1}^{n}$ with n elements, cf. Figure 3.4.

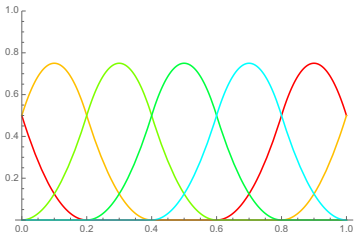


Figure 3.4: Cardinal splines of degree 2 for $S^{per}_{2,h}(0,1)$

The advantage of these splines is that the stiffness matrix (1.12) is a *circulant matrix*. We call a symmetric matrix $M = (m_{i,j})_{i,j=1}^{n}$ circulant, if there are coefficients $(\mu_l)_{l=0}^{n-1}$ such that

$$m_{i,j} = \mu_{(i-j) \bmod n},$$

where $i \bmod n := i - \lfloor i/n \rfloor n \in \{0,\dots,n-1\}$ is the remainder when differentiating. So, a 5×5 matrix looks as follows

$$M = \begin{pmatrix} \mu_0 & \mu_1 & \mu_2 & \mu_3 & \mu_4 \\ \mu_4 & \mu_0 & \mu_1 & \mu_2 & \mu_3 \\ \mu_3 & \mu_4 & \mu_0 & \mu_1 & \mu_2 \\ \mu_2 & \mu_3 & \mu_4 & \mu_0 & \mu_1 \\ \mu_1 & \mu_2 & \mu_3 & \mu_4 & \mu_0 \end{pmatrix}.$$

From the construction of the periodic cardinal splines, we immediately obtain

$$(\widehat{C}_{i,p,h}, \widehat{C}_{j,p,h})_{L_2(0,1)} = (\widehat{C}_{i+n,p,h}, \widehat{C}_{j+n,p,h})_{L_2(0,1)}$$

and

$$(\widehat{C}_{i,p,h}, \widehat{C}_{j,p,h})_{H^1(0,1)} = (\widehat{C}_{i+n,p,h}, \widehat{C}_{j+n,p,h})_{H^1(0,1)}.$$

Therefore, the *mass matrix*

$$M_h := [(\widehat{C}_{i,p,h}, \widehat{C}_{j,p,h})_{L_2(0,1)}]_{i,j=1}^n$$

and the *stiffness matrix*

$$A_h := [(\widehat{C}_{i,p,h}, \widehat{C}_{j,p,h})_{H^1(0,1)}]_{i,j=1}^n$$

are circulant matrices.

For being able to set up a matrix-vector formulation of (3.7), we need to be able to give a matrix representation of the *canonical embedding* $S_{p,\eta} \to S_{p,\eta/2}$, i.e, we need to know how coarse-grid basis functions can be represented by fine-grid basis functions.

Lemma 3.7. *For all $p \in \mathbb{N}$, all grid sizes h and all $x \in \mathbb{R}$,*

$$C_{j,p,h}(x) = 2^{-p} \sum_{l=0}^{p+1} \begin{pmatrix} p+1 \\ l \end{pmatrix} C_{2j+1,p,h/2}(x)$$

is satisfied for all $j = -p, \ldots, n-p-1$.

The lemma can be shown by induction in p; a proof can be found in [9, (4.3.4)].

This directly carries over to the periodic splines, i.e., we obtain

$$\widehat{C}_{j,p,h}(x) = \sum_{i \in \mathbb{Z}} \underbrace{2^{-p} \begin{pmatrix} p+1 \\ i-2j \end{pmatrix}}_{p_{h/2,i,j} :=} \widehat{C}_{i,p,h/2}(x). \tag{3.8}$$

Here, we use that the binomial coefficient $\begin{pmatrix} a \\ b \end{pmatrix}$ vanishes for $b \notin \{0, \ldots, a\}$. We define the matrix

$$P_{h/2} := [p_{h/2,i,j}]_{i=1,\ldots,2n}^{j=1,\ldots,n}.$$

Having the prolongation matrix $P_{h/2}$, and the stiffness matrix A_h, we can represent the projector $\Pi_{p,h}^{per}$ as follows.

Lemma 3.8. *Provided*

$$u_{h/2} := \Pi_{p,h}^{per} v_{h/2} \quad \text{and} \quad \underline{w}_{h/2} := P_{h/2} A_h^{-1} P_{h/2}^T A_{h/2} \underline{v}_{h/2}$$

for all $v_{h/2} \in S_{p,h/2}^{per}$, we have

$$u_{h/2} = w_{h/2}.$$

Proof. As $u_{h/2} = \Pi_{p,h}^{per} v_{h/2}$, Galerkin orthogonality states

$$(u_{h/2} - v_{h/2}, q_h)_{H^1(0,1)} = 0 \quad \text{for all} \quad q_h \in S_{p,h}^{per}.$$

We also obtain

$$\begin{aligned}
(w_{h/2} - v_{h/2}, q_h)_{H^1(0,1)} &= (A_{h/2}(\underline{w}_{h/2} - \underline{v}_{h/2}), P_{h/2}\underline{q}_h)_{\ell^2} \\
&= (P_{h/2}^T A_{h/2}(P_{h/2} A_h^{-1} P_{h/2}^T A_{h/2} \underline{v}_{h/2} - \underline{v}_{h/2}), \underline{q}_h)_{\ell^2} \\
&= (\underbrace{P_{h/2}^T A_{h/2} P_{h/2}}_{= A_h} A_h^{-1} P_{h/2}^T A_{h/2} \underline{v}_{h/2} - P_{h/2}^T A_{h/2} \underline{v}_{h/2}, \underline{q}_h)_{\ell^2} \\
&= (P_{h/2}^T A_{h/2} \underline{v}_{h/2} - P_{h/2}^T A_{h/2} \underline{v}_{h/2}, \underline{q}_h)_{\ell^2} = 0 \quad \text{for all} \quad q_h \in S_{p,h}^{per}.
\end{aligned}$$

The combination of the last two statements yields

$$0 = (w_{h/2} - u_{h/2}, q_h)_{H^1(0,1)} \quad \text{for all} \quad q_h \in S_{p,h}^{per}. \tag{3.9}$$

Note that we have $w_{h/2} \in S_{p,h}^{per}$ and $u_{h/2} \in S_{p,h}^{per}$ by construction. So, (3.9) implies $w_{h/2} = u_{h/2}$, which finishes the proof. □

We observe that this Lemma yields a matrix formulation of the projector $\Pi_{p,h}^{per}$.

Let $\|\cdot\|$ be the Euclidean norm and let the square root $A^{1/2}$ of a symmetric and positive definite matrix A be that symmetric and positive definite matrix that satisfies $A^{1/2}A^{1/2} = A$. Using this notation, we can rewrite (3.7) in matrix-vector notation as

$$\|M_h^{1/2}(I - P_{h/2} A_h^{-1} P_{h/2}^T A_{h/2}) \underline{v}_{h/2}\| \leq \tfrac{1}{\sqrt{2}} h \|A_h^{1/2} \underline{v}_{h/2}\| \quad \text{for all} \quad \underline{v}_h \in \mathbb{R}^{n_{h/2}}$$

and in matrix notation as

$$\|M_h^{1/2}(I - P_{h/2} A_h^{-1} P_{h/2}^T A_{h/2}) A_h^{-1/2}\| \leq \tfrac{1}{\sqrt{2}} h.$$

This is equivalent to

$$\rho\big(M_h(I - P_{h/2} A_h^{-1} P_{h/2}^T A_{h/2}) A_{h/2}^{-1} M_h\big) \leq \tfrac{h}{2}, \tag{3.10}$$

where ρ denotes the spectral radius.

3.3.2 Fourier Analysis

Fourier Analysis is a tool to analyze circulant matrices. Consider the matrix

$$A_h := \begin{pmatrix} a & b & & & c \\ c & a & b & & \\ & c & a & b & \\ & & c & a & b \\ b & & & c & a \end{pmatrix}$$

and for any $j \in \mathbb{Z}$ the vector

$$\underline{\varphi}_j := \begin{pmatrix} \mathrm{e}^{2\mathrm{i}\pi * 0 * j/5} \\ \mathrm{e}^{2\mathrm{i}\pi * 1 * j/5} \\ \mathrm{e}^{2\mathrm{i}\pi * 2 * j/5} \\ \mathrm{e}^{2\mathrm{i}\pi * 3 * j/5} \\ \mathrm{e}^{2\mathrm{i}\pi * 4 * j/5} \end{pmatrix}$$

and observe

$$\begin{aligned} A_h\underline{\varphi}_j &= a\underline{\varphi}_j + b\begin{pmatrix} \mathrm{e}^{2\mathrm{i}\pi*1*j/5} \\ \mathrm{e}^{2\mathrm{i}\pi*2*j/5} \\ \mathrm{e}^{2\mathrm{i}\pi*3*j/5} \\ \mathrm{e}^{2\mathrm{i}\pi*4*j/5} \\ \mathrm{e}^{2\mathrm{i}\pi*0*j/5} \end{pmatrix} + c\begin{pmatrix} \mathrm{e}^{2\mathrm{i}\pi*4*j/5} \\ \mathrm{e}^{2\mathrm{i}\pi*0*j/5} \\ \mathrm{e}^{2\mathrm{i}\pi*1*j/5} \\ \mathrm{e}^{2\mathrm{i}\pi*2*j/5} \\ \mathrm{e}^{2\mathrm{i}\pi*3*j/5} \end{pmatrix} \\ &= a\ \underline{\varphi}_j + b\ \mathrm{e}^{2\mathrm{i}\pi*1*j/5}\underline{\varphi}_j + c\ \mathrm{e}^{2\mathrm{i}\pi*(-1)*j/5}\underline{\varphi}_j \\ &= (a + b\ \mathrm{e}^{2\mathrm{i}\pi*1*j/5} + c\ \mathrm{e}^{2\mathrm{i}\pi*(-1)*j/5})\underline{\varphi}_j. \end{aligned}$$

We observe that $\underline{\varphi}_j$ is an eigenvector and the *symbol*

$$a + b\ \mathrm{e}^{2\mathrm{i}\pi*1*j/5} + c\ \mathrm{e}^{2\mathrm{i}\pi*(-1)*j/5}$$

is a corresponding eigenvalue. Note that the symbol is real for symmetric matrices.

We can generalize this to $n \times n$ circulant matrices. In this case, we choose

$$\varphi_j := [\mathrm{e}^{2\mathrm{i}\pi*i*j/n}]_{i=1}^n.$$

We can combine these vectors to a matrix

$$\mathbb{F}_n := [\mathrm{e}^{2\mathrm{i}\pi*i*j/n}]_{i,j=1}^n,$$

which can be proven to be non-singular. Since the vectors $\underline{\varphi}_j$ are eigenvectors, we obtain that $\mathbb{F}_n$ diagonalizes any circulant matrix: the diagonal entries are then the values of the symbol, evaluated for $j = 1, 2, \ldots$.

We can apply this technique to diagonalize the mass matrix and the stiffness matrix; so the following terms are diagonal matrices:

$$\mathbb{F}_{h^{-1}}^* A_h \mathbb{F}_{h^{-1}}, \quad \mathbb{F}_{h^{-1}}^* M_h \mathbb{F}_{h^{-1}}, \quad \mathbb{F}_{2h^{-1}}^* A_{h/2} \mathbb{F}_{2h^{-1}}, \quad \mathbb{F}_{2h^{-1}}^* M_{h/2} \mathbb{F}_{2h^{-1}}.$$

We can show moreover that

$$\mathbb{F}_{2h^{-1}}^* P_{h/2} \mathbb{F}_{h^{-1}} = \begin{pmatrix} * & & \\ & \ddots & \\ & & * \\ * & & \\ & \ddots & \\ & & * \end{pmatrix},$$

i.e., that $\mathbb{F}_{2h^{-1}}^* P_{h/2} \mathbb{F}_{h^{-1}}$ is a 2×1 block matrix, consisting of two diagonal matrices.

Based on these terms, we can compute the Fourier transform of the matrix in (3.10) and obtain that it has the following form:

$$\begin{pmatrix} \widehat{a}_1 & & & \widehat{b}_1 & & \\ & \ddots & & & \ddots & \\ & & \widehat{a}_n & & & \widehat{b}_n \\ \widehat{c}_1 & & & \widehat{d}_1 & & \\ & \ddots & & & \ddots & \\ & & \widehat{c}_n & & & \widehat{d}_n \end{pmatrix},$$

i.e., it is a 2×2 block-matrix consisting of diagonal matrices. The spectral radius of this matrix is

$$\max_{i=1,\dots,n} \rho\Big(\begin{pmatrix} \widehat{a}_i & \widehat{b}_i \\ \widehat{c}_i & \widehat{d}_i \end{pmatrix} \Big).$$

So, the problem boils down to compute the spectral radius of 2×2-matrices. Doing all the details is a lot of work. Computers might help.

3.3.3 A proof for the non-periodic case

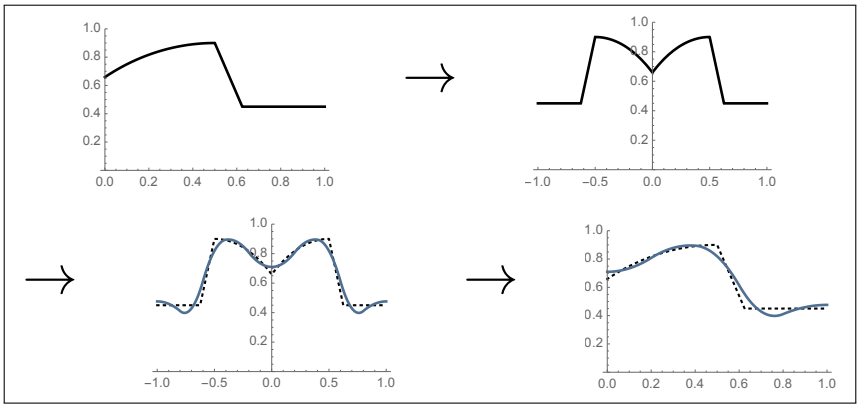


Figure 3.5: A proof for the non-periodic case

Using the definition

$$\widetilde{S}_{p,h}(0,1) := \{v \in S_{p,h}(0,1) \ : \ \tfrac{d}{dx^r}u(0) = \tfrac{d}{dx^r}u(1) = 0 \text{ for } r = 1,3,\dots,2\lfloor \tfrac{p}{2} \rfloor - 1\},$$

we obtain the following result.

Theorem 3.9. *Provided $p \in \mathbb{N}$, $h = 1/n$, $n \in \mathbb{N}$, $hp < |(0,1)| = 1$, we have*

$$\|u - \widetilde{\Pi}_{p,h}u\|_{L_2(0,1)} \leq \sqrt{2}h|u|_{H^1(0,1)}$$

for all $u \in H^1(0,1)$, where $\widetilde{\Pi}_{p,h}$ is the H^1-orthogonal projector $H^1(0,1) \to \widetilde{S}_{p,h}(0,1)$.

The proof is visualized in Figure 3.5.

Proof. Let $u \in H^1(0,1)$. Then define $w(x) := u(|x|)$ and observe $w \in H^{1,per}(-1,1)$. Then $\|w - \Pi^{per}_{p,h}w\|_{L_2(-1,1)} \leq \sqrt{2}h|w|_{H^1(-1,1)}$.
We have by construction $\Pi^{per}_{p,h}w \in S^{per}_{p,h}(-1,1)$. We can show $(\Pi^{per}_{p,h}w)(x) = (\Pi^{per}_{p,h}w)(-x)$ and we have by construction $w(x) = w(-x)$. Using these two statements, we obtain that $(\Pi^{per}_{p,h}w)|_{(0,1)} \in \widetilde{S}_{p,h}(0,1)$.
Now, we define $u_h := (\Pi^{per}_{p,h}w)|_{(0,1)}$ and observe

$$\frac{\|u - u_h\|_{L_2(0,1)}}{|u|_{H^1(0,1)}} = \frac{\frac{1}{\sqrt{2}}\|w - \widetilde{\Pi}_{p,h}w\|_{L_2(-1,1)}}{\frac{1}{\sqrt{2}}|w|_{H^1(-1,1)}} \leq \sqrt{2}h.$$

This shows the error bound $\|u - u_h\|_{L_2(0,1)} \leq \sqrt{2}h|u|_{H^1(0,1)}$. Finally, we have to show $(u_h - \widetilde{\Pi}_{p,h}u, v)_{H^1(0,1)} = 0$ for all $v \in \widetilde{S}_{p,h}$ to know that $u_h = \widetilde{\Pi}_{p,h}u$. □

We immediately obtain

- $\|u - Q_{p,h}u\|_{L_2(0,1)} = \inf_{u_h \in S_{p,h}} \|u - u_h\|_{L_2(0,1)} \leq \sqrt{2}h|u|_{H^1(0,1)}$, where $Q_{p,h}$ is the L_2-orthogonal projector into $S_{p,h}(0,1)$.
- $\|u - \Pi_{p,h}u\|_{L_2(0,1)} \leq \|u - \widetilde{\Pi}_{p,h}u\|_{L_2(0,1)} + \|\widetilde{\Pi}_{p,h} - \Pi_{p,h}u\|_{L_2(0,1)} = \|(I - \widetilde{\Pi}_{p,h})u\|_{L_2(0,1)} + \|(I - \widetilde{\Pi}_{p,h})\Pi_{p,h}u\|_{L_2(0,1)} \leq \sqrt{2}h(|u|_{H^1(0,1)} + |\Pi_{p,h}u|_{H^1(0,1)}) \leq 2\sqrt{2}h|u|_{H^1(0,1)}$.

3.4 Literature

The results of the first section follow results given in [7, Section 3.3] and [10]. The results of the second section originate in [8]. The results of the last section can be found in [11]. A completely different proof (with better constants) has been given in [12]. In [13], that analysis is extended to the non-equidistant case.

Chapter 4

Inverse inequalities

4.1 Some inverse inequalities

For the space of polynomials $\mathbb{P}_p$, the following statement holds, cf. [7, Corollary 3.94].

Theorem 4.1. *Provided $p \in \mathbb{Z}^+$, the estimate*

$$|v|_{H^1(0,1)} \leq 2\sqrt{3}p^2\|v\|_{L_2(0,1)}$$

holds for all $v \in \mathbb{P}_p$.

This result was recently improved in [14] to

$$\sqrt{\frac{p(p+1)(p+2)(p+3)}{4}} \leq \sup_{v\in\mathbb{P}_p} \frac{|v|_{H^1(0,1)}}{\|v\|_{L_2(0,1)}} \leq \sqrt{\frac{p(p+1)(p+2)(p+3)}{2}},$$

which shows sharpness.
This result can be easily carried over to splines.

Theorem 4.2. *Let $p \in \mathbb{Z}^+$ and Ξ be a p-open knot vector. Then, the estimate*

$$|v_h|_{H^1(0,1)} \leq 2\sqrt{3}p^2 h_{\min}^{-1}\|v_h\|_{L_2(0,1)}$$

holds for all $v_h \in S_{p,\Xi}(0,1)$, where

$$h_{\min} := \min_{i=1,\dots,N-1}(\zeta_{i+1} - \zeta_i)$$

is the size of the smallest interval.

Proof. Observe that Theorem 4.1 and a standard scaling argument imply

$$|v_h|_{H^1(\zeta_i,\zeta_{i+1})} \leq \frac{2\sqrt{3}p^2}{\zeta_{i+1}-\zeta_i}\|v_h\|_{L_2(\zeta_i,\zeta_{i+1})} \leq 2\sqrt{3}p^2 h_{\min}^{-1}\|v_h\|_{L_2(\zeta_i,\zeta_{i+1})}$$

for all $i = 1,\dots,N-1$. (The scaling argument is just the application of Theorems 2.15 and 2.16 to $F(x) = \zeta_i + x(\zeta_{i+1} - \zeta_i)$.) By squaring the formula, and taking the sum for $i = 1,\dots,N-1$, we obtain the desired result. □

Note that this proof does not use *any* information on the *smoothness* of the spline function. We only use that we are dealing with a piecewise polynomial function. Now the question may arise if we can do better for splines with more smoothness.

Theorem 4.3. *Let $p \in \mathbb{Z}^+$ and let Ξ be a p-open knot vector on $(-1,1)$ without repeated knots. Define*

$$S^{per}_{p,\Xi}(-1,1) := \{v_h \in S_{p,\Xi}(-1,1) \ : \ \tfrac{d^r}{dx^r} v_h(-1) = \tfrac{d^r}{dx^r} v_h(1) \text{ for } r = 0,\dots,p-1\}$$

to be the space of periodic splines (with maximum smoothness). Then, we have

$$|v_h|_{H^1(-1,1)} \leq 2\sqrt{3} h_{\min}^{-1} \|v_h\|_{L_2(-1,1)}$$

for all $v_h \in S^{per}_{p,\Xi}(-1,1)$.

Proof. This theorem is shown by induction. For $p = 1$, the result follows from Theorem 4.2. Now, consider some fixed $v_h \in S^{per}_{p,\Xi}(-1,1)$. Using integration by parts and Cauchy-Schwarz inequality, we obtain

$$|v_h|^2_{H^1(-1,1)} = (v_h, v_h)_{H^1(-1,1)} = (-v_h'', v_h)_{L_2(-1,1)} \leq |v_h'|_{H^1(-1,1)} \|v_h\|_{L_2(-1,1)}$$

Observe that $w_h := v_h' \in S^{per}_{p-1,\Xi}(-1,1)$. So, we can apply the induction hypothesis for $p-1$, i.e.,

$$|w_h|_{H^1(-1,1)} \leq 2\sqrt{3} h_{\min}^{-1} \|w_h\|_{L_2(-1,1)} \qquad \text{for all} \qquad w_h \in S^{per}_{p-1,\Xi}(-1,1),$$

and obtain

$$|v_h|^2_{H^1(-1,1)} \leq 2\sqrt{3} h_{\min}^{-1} \|u_h'\|_{L_2(-1,1)} \|v_h\|_{L_2(-1,1)} = 2\sqrt{3} p^2 h_{\min}^{-1} |u_h|_{H^1(-1,1)} \|v_h\|_{L_2(-1,1)}.$$

Now, we divide by $|u_h|_{H^1(-1,1)}$ and obtain

$$|v_h|_{H^1(-1,1)} \leq 2\sqrt{3} h_{\min}^{-1} \|v_h\|_{L_2(-1,1)},$$

i.e., the desired statement. (If $|u_h|_{H^1(-1,1)} = 0$, we cannot do the last step. But in this case, the desired statement is anyway obvious.) □

The same result is possible for the tilde-spaces.

Theorem 4.4. *Let $p \in \mathbb{Z}^+$ and let Ξ be a p-open knot vector without repeated knots. Define*

$$\widetilde{S}_{p,\Xi}(0,1) := \{v_h \in S_{p,\Xi}(0,1) \ : \ \tfrac{d^r}{dx^r} v_h(0) = \tfrac{d^r}{dx^r} v_h(1) = 0 \text{ for } r = 1,3,\dots,2\lceil \tfrac{p-1}{2} \rceil - 1\}$$

to be the space of splines (with maximum smoothness) whose odd derivatives vanish on the boundary. Then, we have

$$|v_h|_{H^1(0,1)} \leq 2\sqrt{3} h_{\min}^{-1} \|v_h\|_{L_2(0,1)}$$

for all $v_h \in \widetilde{S}_{p,\Xi}(0,1)$.

Proof. Let $v_h \in \widetilde{S}_{p,\Xi}(0,1)$ arbitrary but fixed. Define on $(-1,1)$ a function w_h by $w_h(x) := v_h(|x|)$ and observe $w_h \in S^{per}_{p,\Xi}(-1,1)$.
Observe moreover that Theorem 4.4 yields

$$\frac{|v_h|_{H^1(0,1)}}{\|v_h\|_{L_2(0,1)}} = \frac{\frac{1}{\sqrt{2}}|w_h|_{H^1(0,1)}}{\frac{1}{\sqrt{2}}\|w_h\|_{L_2(0,1)}} \leq 2\sqrt{3}h_{\min}^{-1},$$

which yields the desired result. □

The results of the last two theorems can be easily extended to splines with lower smoothness.

Theorem 4.5. *Let $p, k \in \mathbb{Z}^+$ with $k < p$ and let Ξ be a p-open knot vector with at most $p-k$ repeated knots, i.e., with smoothness of at least k, and let*

$$\widetilde{S}_{p,k,\Xi}(0,1) := \{v_h \in S_{p,\Xi}(0,1) \ : \ \tfrac{d^r}{dx^r}v_h(0) = \tfrac{d^r}{dx^r}v_h(1) = 0 \text{ for } r = 1,3,\dots,2\lceil\tfrac{k}{2}\rceil - 1\}.$$

Then, we obtain

$$|v_h|_{H^1(0,1)} \leq 2\sqrt{3}h_{\min}^{-1}(p-k)^2\|v_h\|_{L_2(0,1)}$$

for all $v_h \in \widetilde{S}_{p,k,\Xi}(0,1)$.

We obtain an analogous result for the periodic splines.

Now, the question may arise if it is possible to extend the p-robust error estimates to the whole spline space $S_{p,\Xi}(0,1)$. The following remark shows that this is impossible.

Remark 4.6. *Let Ξ be a p-open knot vector and let $\zeta_1 = 0$ and $\zeta_2 > 0$ be the first two breakpoints. Observe that the first B-spline basis function looks as follows:*

$$\widehat{B}_{1,p,\Xi}(x) = \begin{cases} (1-\frac{x}{\zeta_2})^p & \text{for } x < \zeta_2 \\ 0 & \text{otherwise.} \end{cases}$$

We can explicitly compute its L^2-norm and its H^1-seminorm:

$$\|\widehat{B}_{1,p,\Xi}\|_{L_2(0,1)} = \zeta_2^{1/2}\sqrt{\frac{1}{2p+1}}$$

$$|\widehat{B}_{1,p,\Xi}|_{H^1(0,1)} = \zeta_2^{-1/2}p\sqrt{\frac{1}{2p-1}}$$

This shows

$$\frac{\|\widehat{B}_{1,p,\Xi}\|_{L_2(0,1)}}{|\widehat{B}_{1,p,\Xi}|_{H^1(0,1)}} = \zeta_2 p^{-1}\sqrt{\frac{2p-1}{2p+1}} \leq hp^{-1},$$

which implies

$$|\widehat{B}_{1,p,\Xi}|_{H^1(0,1)} \geq ph^{-1}\|\widehat{B}_{1,p,\Xi}\|_{L_2(0,1)},$$

i.e., that a robust estimate is not possible. Note that here p only enters linearly, but in Theorem (4.3), it enters quadratically.

The results of the last two theorems can be easily extended to higher Sobolev indices.

Remark 4.7. *Provided $k \geq r$ and $u_h \in S^{per}_{p,k,\Xi}(-1,1)$, we have $\frac{d^r}{dx^r} u_h \in S^{per}_{p-1,k-1,\Xi}(-1,1)$. In this case, we have*

$$|u_h|_{H^{r+1}(-1,1)} = |\tfrac{d^r}{dx^r} u_h|_{H^1(-1,1)} \leq 2\sqrt{3} h_{\min} \|\tfrac{d^r}{dx^r} u_h\|_{L_2(-1,1)} = 2\sqrt{3} h_{\min} |u_h|_{H^r(-1,1)}.$$

Analogous results are possible for $\widetilde{S}_{p,k,\Xi}(0,1)$ and $S_{p,k,\Xi}(0,1)$.

Also the extension to the multivariate case and to the physical domain are straight-forward.

Remark 4.8. *Let $\widehat{\Omega} := (0,1)^2$ and $u_h \in \widehat{S}_{p,\boldsymbol{\Xi}}(\widehat{\Omega})$. Note that $u_h(x,\cdot)$ is a spline, so we obtain*

$$|u_h(x,\cdot)|_{H^1(0,1)} \leq 2\sqrt{3} p h_{\min} \|u_h(x,\cdot)\|_{L_2(0,1)}$$

and by integrating the square

$$\|\tfrac{\partial}{\partial x} u_h\|_{L_2(\widehat{\Omega})} \leq 2\sqrt{3} p h_{\min} \|u_h\|_{L_2(\widehat{\Omega})}.$$

Using the definition of the H^1-seminorm, we obtain

$$|u_h|_{H^1(\widehat{\Omega})} \leq 2\sqrt{3\ d} p h_{\min} \|u_h\|_{L_2(\widehat{\Omega})}.$$

where $d = 2$.
The extension to the physical domain follows using Theorems 2.15 and 2.16.
Analogous results are possible for the extension of the spaces $S^{per}_{p,k,\Xi}(-1,1)$ and $\widetilde{S}_{p,k,\Xi}(0,1)$.

4.2 The spectrum of the splines

The inverse estimate is strongly related to the spectrum of $M_h^{-1} A_h$, where M_h is the mass matrix and A_h is the stiffness matrix:

$$\frac{|u_h|_{H^1(0,1)}}{\|u_h\|_{L_2(0,1)}} = \frac{\|\underline{u}_h\|_{A_h}}{\|\underline{u}_h\|_{M_h}} \in \sigma(M_h^{-1} A_h),$$

where σ denotes the spectrum.

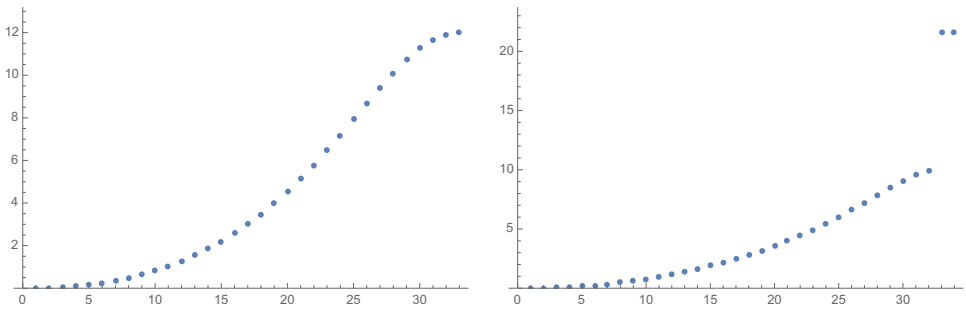


Figure 4.1: Generalized spectrum for $S_{1,h}$ (left) and $S_{2,h}$ (right)

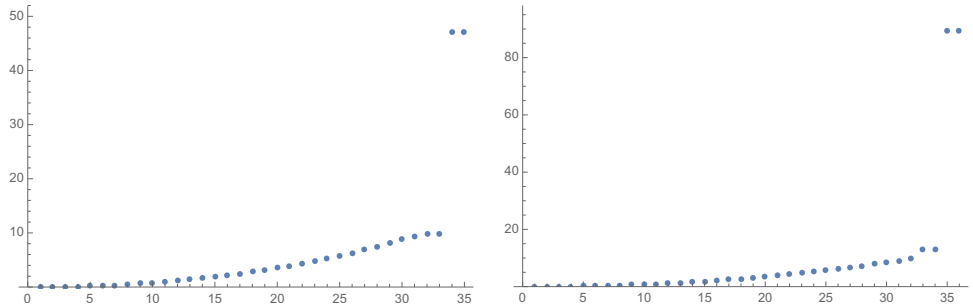


Figure 4.2: Generalized spectrum for $S_{3,h}$ (left) and $S_{4,h}$ (right)

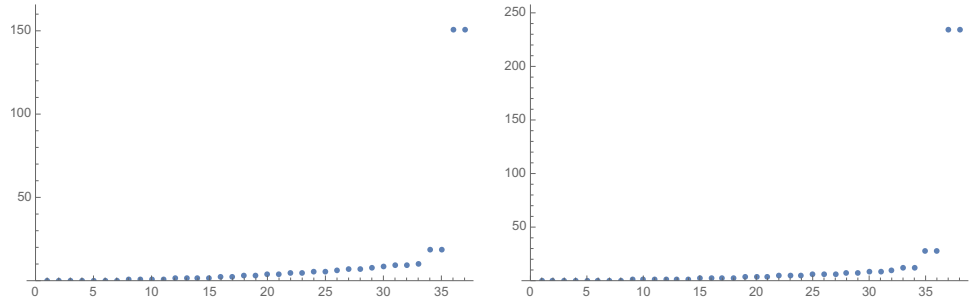


Figure 4.3: Generalized spectrum for $S_{5,h}$ (left) and $S_{6,h}$ (right)

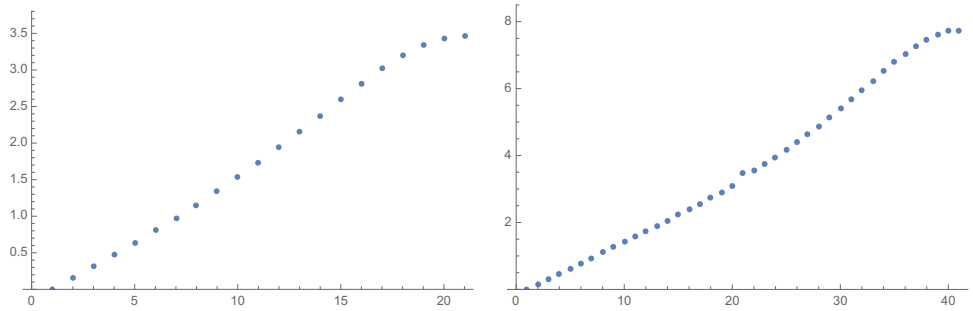


Figure 4.4: Generalized spectrum for $S_{1,0,h}$ (left) and $S_{2,0,h}$ (right)

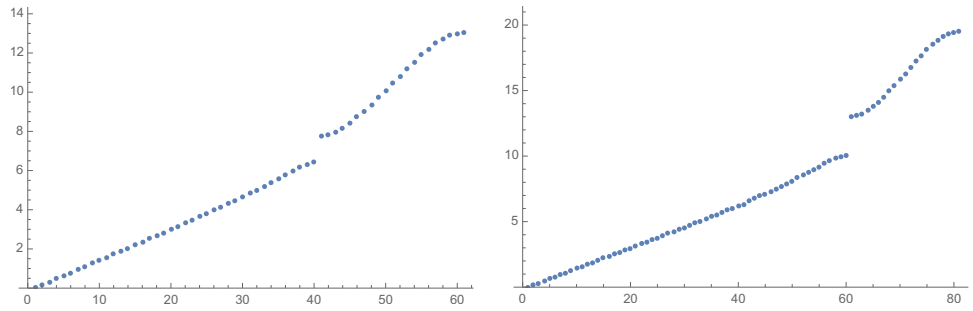


Figure 4.5: Generalized spectrum for $S_{3,0,h}$ (left) and $S_{4,0,h}$ (right)

Fig. 4.1, 4.2 and 4.3 show the spectra of the splines of maximum smoothness. We see that for $p \geq 2$, there are some outliers and that the outliers come pairwise and that the value of

the outliers is increasing. The number of outliers is bounded by

$$\dim S_{p,h} - \dim \widetilde{S}_{p,h} = 2 \left\lfloor \frac{p}{2} \right\rfloor .$$

Fig. 4.4 and 4.5 depict the spectra of the C^0-splines. Here, we observe that the spectrum is split up into p branches, where the branches on the right-hand-side again diverge.

4.3 Why inverse inequalities show sharpness of approximation error estimates (and vice versa)?

Assume that we could improve the approximation error estimate compared to Theorem 3.9 qualitatively, i.e., assume that we can show

$$\inf_{v_H \in \widetilde{S}_{p,H}(0,1)} \|u - v_H\|_{L_2(0,1)} \leq \Phi(p) H |u|_{H^1(0,1)} \quad \text{for all} \quad u \in H^1(0,1), \tag{4.1}$$

where $\Phi(p) \to 0$ as $p \to \infty$.

Let $n \in \mathbb{N}$ and let $h := 1/n$ and $H := 1/(n-1)$. Let $\Pi_{p,H}$ be the L^2-orthogonal projection into $\widetilde{S}_{p,H}(0,1)$. As $\dim \widetilde{S}_{p,h}(0,1) > \dim \widetilde{S}_{p,H}(0,1)$, there is some $u_h \in \widetilde{S}_{p,h}$ such that

$$(u_h, v_H)_{L_2(0,1)} = 0 \qquad \text{for all} \qquad v_H \in \widetilde{S}_{p,H}(0,1),$$

i.e., it is in the L_2-orthogonal complement. This means that

$$\Pi_{p,H} u_h = 0.$$

Now, we have using (4.1) and the robust inverse estimate

$$\begin{aligned} \|u_h\|_{L_2(0,1)} = \|u_h - \Pi_{p,H} u_h\|_{L_2(0,1)} &= \inf_{v_H \in \widetilde{S}_{p,H}(0,1)} \|u_h - v_H\|_{L_2(0,1)} \leq \phi(p) H |u_h|_{H^1(0,1)} \\ &\leq 2\sqrt{3}\Phi(p) \frac{n}{n+1} \|u_h\|_{L_2(0,1)} \leq 4\sqrt{3}\Phi(p) \|u_h\|_{L_2(0,1)}, \end{aligned}$$

and, therefore,

$$\Phi(p) \geq \frac{1}{4\sqrt{3}}.$$

This contradicts our assumption that $\Phi(p) \to 0$ as $p \to \infty$.

4.4 Literature

The standard results can be found in [7]. The results on the periodic splines and on the tilde splines can be found in [11].

Chapter 5

Assembling matrices in IgA

5.1 Introduction

In this chapter, we discuss the computation of the stiffness matrix

$$A_h = [a(\varphi_i, \varphi_j)]_{i,j=1}^N,$$

where the functions φ_i are the basis functions of V_h, i.e., on the physical domain and

$$a(\cdot,\cdot) = (\nabla\cdot, \nabla\cdot)_{L_2(\Omega)}.$$

Using the pull-back definition $\varphi_i = \widehat{\varphi}_i \circ F^{-1}$ and the chain-rule, we obtain

$$\begin{aligned} a(\varphi_i, \varphi_j) &= \int_\Omega \nabla_x \varphi_i(x) \cdot \nabla_x \varphi_j(x)\,\mathrm{d}x \\ &= \int_{\widehat{\Omega}} [(\nabla_\xi \widehat{\varphi}_i) \circ F^{-1}(x) * \nabla_x F^{-1}(x)] \cdot [(\nabla_\xi \widehat{\varphi}_j) \circ F^{-1}(x) * \nabla_x F^{-1}(x)]\,\mathrm{d}x. \end{aligned}$$

Using the usual rule for the derivative of an inverse function

$$\underbrace{\frac{d}{d\xi}\underbrace{g^{-1}}_{\text{inverse function}}(\xi)}_{} = \overbrace{\left(\frac{d}{dx}g(x)\right)^{-1}}^{\text{inverse matrix}}\Big|_{x=\underbrace{g^{-1}}_{\text{inverse function}}(\xi)},$$

we obtain further

$$a(\varphi_i, \varphi_j) = \int_\Omega [(\nabla_\xi \widehat{\varphi}_i) \circ F^{-1}(x)[\nabla_\xi F]^{-1} \circ F^{-1}(x)] \cdot [(\nabla_\xi \widehat{\varphi}_j) \circ F^{-1}(x)[\nabla_\xi F]^{-1} \circ F^{-1}(x)]\,\mathrm{d}x,$$

where $[\nabla_\xi F]^{-1}$ is the inverse (matrix) of the Jacobi matrix.
Now, the substitution rule yields further

$$\begin{aligned} a(\varphi_i, \varphi_j) &= \int_{\widehat{\Omega}} \left(\nabla_\xi \widehat{\varphi}_i(\xi)[\nabla_\xi F(\xi)]^{-1}\right) \cdot \left(\nabla_\xi \widehat{\varphi}_j(\xi)[\nabla_\xi F(\xi)]^{-1}\right) |\det \nabla_\xi F(\xi)|\,\mathrm{d}\xi \\ &= \int_{\widehat{\Omega}} \left(\nabla_\xi \widehat{\varphi}_i(\xi)\right)^T \left(\nabla_\xi F(\xi)\right)^{-T} \left(\nabla_\xi F(\xi)\right)^{-1} \left(\nabla_\xi \widehat{\varphi}_j(\xi)\right) |\det \nabla_\xi F(\xi)|\,\mathrm{d}\xi. \end{aligned}$$

- For the mass matrix, we only need to use the substitution rule. For *other differential operators*, we can do similar steps.
- We note that we *never* have to *evaluate* F^{-1}, the inverse (function) of the geometry mapping. This is good because evaluating the inverse function is hard.
- We have to evaluate $a(\varphi_i, \varphi_j)$ for any pair (i, j). As we know that our basis functions have local support, we know that the matrix is sparse. The sparsity pattern can be determined easily. So, we have to evaluate $a(\varphi_i, \varphi_j)$ only for the pairs (i, j), for which we do not know that the bilinear form a vanishes.
- In standard high- or low-order FEM, the stiffness matrix and the mass matrix are computed element-wise. We can do the same for IgA, see Section 5.2.
- In standard low-order FEM, the quadrature is done *exactly*. In IgA, particularly if NURBS are used, the integrals are approximated by quadrature rules.

5.2 Element-wise quadrature

In this section, we consider *element-wise quadrature*. So, we have just polynomials instead of splines.

Let φ and ψ be polynomials of degree p. For the univariate stiffness matrix, we are interested in computing

$$\int_{\widehat{I}} \varphi'(x)\psi'(x)\,\mathrm{d}x.$$

Here, both φ' and ψ' are polynomials of degree $p-1$. Their *product* is a *polynomial* of degree $2p-2$. In two dimensions, we have

$$\int_{\widehat{I}} \underbrace{\underbrace{\frac{\partial}{\partial x_1}\varphi(\mathbf{x})}_{\text{degree } 2p-2} \underbrace{\frac{\partial}{\partial x_1}\psi(\mathbf{x})}_{\text{degree } 2} + \underbrace{\frac{\partial}{\partial x_1}\varphi(\mathbf{x})}_{\text{degree } 2p} \underbrace{\frac{\partial}{\partial x_1}\psi(\mathbf{x})}_{\text{degree } 2p-2}}_{\text{degree } 2p \times 2p} \,\mathrm{d}\mathbf{x},$$

i.e., a *polynomial* of degree $2p \times 2p$.

Such integrals could be solved directly, but it is more efficient to solve them with *quadrature rules*. Quadrature rules are usually specified and discussed for the interval $(-1, 1)$. Integrals for a general interval $\widehat{I} = (a, b)$ are just computed via the substitution rule

$$\int_a^b u(x)\,\mathrm{d}x = \frac{b-a}{2}\int_{-1}^{1} u\left(\frac{b+a}{2} + \frac{b-a}{2}t\right)\,\mathrm{d}t.$$

Each quadrature rule is specified by its *quadrature weights* $\omega_1, \ldots, \omega_m$ and its *quadrature nodes* $t_1, \ldots, t_m$. Then,

$$Q_{\omega,t}(u) := \sum_{i=1}^{m} \omega_i u(t_i)$$

is an approximation (or the exact value) of the integral

$$\int_{-1}^{1} u(x)\,\mathrm{d}x.$$

How many quadrature nodes do we need? Optimal are *Gauss-quadrature* rules, which resolve polynomials of degree up to $2m-1$ exactly. Gauss-quadrature rules are set up based on *orthogonal polynomials*. We have learned in Chapter 3 that the *Legendre polynomials* are the family of orthogonal polynomials if the scalar product $(\cdot,\cdot)_{L_2(-1,1)}$ is considered. Other orthogonal polynomials are obtained if other scalar products are considered.

For simplicity, we restrict ourselves to the *Gauss-Legendre* rule. The Gauss-Legendre rule of order m uses as nodes the roots of the Legendre polynomial L_m. First we show that there are exactly m distinct roots in $(-1,1)$.

Lemma 5.1. *The Legendre polynomial L_m has m distinct roots in $(-1,1)$.*

Proof. For $m=0$, this is obvious. Now, let $m>0$. Observe that orthogonality yields

$$\int_{-1}^{1} L_m(x)\,\mathrm{d}x = \int_{-1}^{1} L_m(x)L_0(x)\,\mathrm{d}x = 0.$$

Therefore, the sign of $L_m(x)$ changes at least once in $(-1,1)$. Let $-1<\eta_1<\eta_2<\cdots<\eta_s<1$ be the locations where L_m changes its sign. Certainly $1\le s\le \deg L_m = m$.

Consider the case $s<m$ first. Using orthogonality, we obtain

$$\int_{-1}^{1} L_m(x)\underbrace{\prod_{i=1}^{s}(x-\eta_s)}_{\text{degree } s}\,\mathrm{d}x = 0.$$

This is only possible if $L_m(x)\prod_{i=1}^{s}(x-\eta_s)$ changes its sign in $(-1,1)$ at least once. This cannot be true because L_m and $\prod_{i=1}^{s}(x-\eta_s)$ change signs exactly at the same places. So, their product cannot change its sign at all.

This shows that $s<m$ was wrong, so we obtain $s=m$. So, L_m changes its sign at the points $\eta_1,\dots,\eta_m$, i.e., these points are roots of L_m. Since L_m is a polynomial of degree m, there are no other roots.

This finishes the proof. □

Theorem 5.2. *Let $t_1,\dots,t_m$ be the roots of L_m and*

$$\omega_i = \int_{-1}^{1} I_i(x)\,\mathrm{d}x, \quad \textit{where} \quad I_i(x) := \prod_{j\in\{1,\dots,m\}\setminus\{i\}} \frac{x-t_j}{t_i-t_j}. \tag{5.1}$$

Then, we have

$$Q_{\omega,t}(u) = \int_{-1}^{1} u(x)\,\mathrm{d}x \quad \textit{for all} \quad u\in\mathbb{P}_{2m-1},$$

i.e., $Q_{\omega,t}$ is a Gauss-quadrature.

Proof. Let $u\in\mathbb{P}_{2m-1}$ be arbitrary but fixed and let $v\in\mathbb{P}_{m-1}$ such that

$$u(t_i) = v(t_i) \quad \text{for } i=1,\dots,m,$$

i.e., v is the interpolation polynomial of u. By construction, we know that

$$w(x) := u(x) - v(x)$$

has roots $t_1, \dots, t_m$. Therefore, we have another polynomial $q(x)$ such that

$$w(x) = \left(\prod_{i=1}^{m} (x - t_i) \right) q(x).$$

Since $\deg w \leq 2m - 1$, we have $\deg q \leq m - 1$. We obtain

$$u(x) = v(x) + \underbrace{\left(\prod_{i=1}^{m} (x - t_i) \right)}_{r(x) :=} q(x).$$

Since both L_m and r are polynomials of degree m sharing the same m roots, we have $r = L_m$. Observe that the functions I_i *Lagrange interpolation* functions, which satisfy

$$I_i \in \mathbb{P}_{m-1} \quad \text{and} \quad I_i(t_j) = \delta_{i,j}.$$

Therefore, we have

$$v(x) = \sum_{i=1}^{m} v(t_i) I_i.$$

Concluding, we obtain

$$\begin{aligned} \int_{-1}^{1} u(x) \, \mathrm{d}x &= \int_{-1}^{1} v(x) \, \mathrm{d}x + \underbrace{\int_{-1}^{1} L_m(x) q(x) \, \mathrm{d}x}_{= 0 \text{ since } q \in \mathbb{P}_{m-1} \text{ and } L_m \text{ is orthogonal to all } q \in \mathbb{P}_{m-1}} \\ &= \sum_{i=1}^{m} \int_{-1}^{1} I_i v(t_i) \, \mathrm{d}x = \sum_{i=1}^{m} \underbrace{\int_{-1}^{1} I_i \, \mathrm{d}x}_{= \omega_i} v(t_i) = Q_{\omega,t}(u), \end{aligned}$$

i.e., that the polynomial u is integrated exactly. □

Note that (5.1) is one way of defining the weights. For practical computations, there are alternatives which are easier to compute.

The Gauss quadrature rule with m nodes does not allow to integrate all polynomials of degree $2m$ exactly.

Lemma 5.3. *Let (ω, t) be the Gauss-quadrature of order m. Then, we have $Q_{\omega,t}(L_m^2) \neq \int_{-1}^{1} L_m^2(x) \, \mathrm{d}x$.*

Proof. Recall that $t_1, \dots, t_m$ are the roots of L_m and, consequently, of L_m^2. Therefore, we have

$$Q_{\omega,t}(L_m^2) = \sum_{i=1}^{m} \omega_i L_m^2(t_i) = 0 < \|L_m\|_{L_2(-1,1)}^2 = \int_{-1}^{1} L_m^2(x) \, \mathrm{d}x,$$

which shows the desired result. Here we use that $L_m \neq 0$ implies $\|L_m\|_{L_2(-1,1)} > 0$. □

The following Lemma is of importance for numerical stability.

Lemma 5.4. *For a Gauss-quadrature, all weights are positive.*

Proof. Consider $I_i^2 \in \mathbb{P}_{2m-2}$. As we have exact quadrature for this function, we obtain

$$\omega_i = \sum_{j=1}^{m} \omega_j I_i^2(t_j) = Q(I_i^2) = \int_{-1}^{1} I_i^2(x)\,\mathrm{d}x = \|I_i\|^2_{L_2(-1,1)} > 0,$$

i.e., that the weights are positive. □

It is straight-forward to extend Gauss quadrature to the *multivariate case*. Consider for simplicity only two dimensions. Let $f \in \mathbb{P}_{2m-1} \otimes \mathbb{P}_{2m-1}$, i.e.,

$$f(x,y) = \sum_{i=0}^{2m-1} \sum_{j=0}^{2m-1} a_{i,j} x^i y^j.$$

Then, we have

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \int_{-1}^{1} \sum_{i=1}^{m} \omega_i^{(1)} f(t_i^{(1)}, y)\,\mathrm{d}y = \sum_{j=1}^{n}\sum_{i=1}^{m} \omega_i^{(1)}\omega_j^{(2)} f(t_i^{(1)}, t_j^{(2)})$$

since

$$f(\cdot, y) \in \mathbb{P}_{2m-1} \quad \text{for all} \quad y \in (-1,1)$$

and

$$f(t_i^{(1)}, \cdot) \in \mathbb{P}_{2m-1} \quad \text{for all} \quad i \in \{1,\ldots,m\}.$$

This shows that quadrature rules can be directly extended to tensor-products and that also the exactness conditions carry over to tensor-products. We have

$$\sum_{j=1}^{n}\sum_{i=1}^{m} \omega_i^{(1)}\omega_j^{(2)} f(t_i^{(1)}, t_j^{(2)}) = Q_{\omega,\mathbf{t}}(f),$$

where

$$\omega = (\omega_1^{(1)}\omega_1^{(2)},\, \omega_1^{(1)}\omega_2^{(2)}, \ldots, \omega_1^{(1)}\omega_n^{(2)},\, \omega_2^{(1)}\omega_1^{(2)},\, \omega_2^{(1)}\omega_2^{(2)}\ \ldots)$$

and

$$\mathbf{t} := t^{(1)} \times t^{(2)} = ((t_1^{(1)}, t_1^{(2)}),\, (t_1^{(1)}, t_2^{(2)}), \ldots, (t_1^{(1)}, t_n^{(2)}),\, (t_2^{(1)}, t_1^{(2)}),\, (t_2^{(1)}, t_2^{(2)}), \ldots)$$

Based on the integration rules from this section, we can evaluate integrals like

$$\int_{\widehat{\Omega}} \widehat{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}}(\mathbf{x})\widehat{B}_{\mathbf{j},\mathbf{p},\boldsymbol{\Xi}}(\mathbf{x})\,\mathrm{d}\mathbf{x} \quad \text{or} \quad \int_{\widehat{\Omega}} \nabla\widehat{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}}(\mathbf{x})\nabla\widehat{B}_{\mathbf{j},\mathbf{p},\boldsymbol{\Xi}}(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

on an element-by-element basis. The quadrature gets more involved, if also a geometry transformation is present or if NURBS are considered.

5.3 Inexact quadrature

The quadrature gets more involved if also a *geometry function* is present. Consider the mass matrix. Let F be a polynomial of degree $p \times p$. Then,

$$\det \nabla F$$

is a polynomial of degree $2p$. Thus,

$$\int_{\widehat{I}} \underbrace{\varphi(\mathbf{x})\psi(\mathbf{x})\det\, \nabla F(\mathbf{x})}_{\text{degree } 4p \times 4p} \,\mathrm{d}\mathbf{x}.$$

This means, that we need $2p+1 \times 2p+1$ quadrature nodes.

Consider now the stiffness matrix:

$$\int_{\widehat{I}} (\nabla\varphi(\mathbf{x}))^T (\nabla F(\mathbf{x}))^{-T} (\nabla F(\mathbf{x}))^{-1} \nabla\psi(\mathbf{x}) \det\, \nabla F(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

and observe that (if ∇F is not constant) the term to be integrated is *not* a polynomial. Thus, there is *no* Gauss quadrature that yields exact results.

So, the *idea* is to use *inexact quadrature*. This means that we use Gauss quadrature on terms that are not polynomials of degree $2m-1$. Certainly, in this case the question arise how many quadrature nodes should be chosen.

To answer this question, we have to do error analysis. For doing error analysis, we have always to go back to the question of our original interest: solving a PDE. So, indeed the question is: how does the usage of an inexact quadrature affect the accuracy of the solution of the PDE? Strang's lemma yields such an estimate.

Lemma 5.5 (First Lemma of Strang)**.** *Let V_h and V be Hilbert spaces such that $V_h \subset V$.*

Let a be the exact bilinear form, a_h be a perturbed bilinear form, f be the exact linear functional (for the right-hand side) and f_h be a perturbed linear functional.

Let $u \in V$ be such that

$$a(u,v) = \langle f, v\rangle \qquad \text{for all } v \in V.$$

Let $u_h \in V_h$ be such that

$$a_h(u_h, v_h) = \langle f_h, v_h\rangle \qquad \text{for all } v_h \in V_h.$$

Assume that there are constants $\underline{\mu}$ and $\overline{\mu}$ such that the assumptions of the Theorem of Lax Milgram (Theorem 1.11) hold for both a and a_h.

Then, there is a constant c that depends only on $\underline{\mu}$ and $\overline{\mu}$ such that

$$\begin{aligned}&\|u-u_h\|_V \\ &\quad \le c\Big(\underbrace{\inf_{v_h\in V_h} \Big(\|u-v_h\|_V}_{\substack{\text{discretization}\\ \text{error, like in the}\\ \text{Lemma of Ceá}}} + \underbrace{\sup_{w_h\in V_h} \frac{|a(v_h,w_h)-a_h(v_h,w_h)|}{\|w_h\|_V}\Big) + \sup_{w_h\in V_h} \frac{|\langle f, w_h\rangle - \langle f_h, w_h\rangle|}{\|w_h\|_V}}_{\text{consistency error}} \Big).\end{aligned}$$

Some remarks:

- It is important that the assumptions on a_h are satisfied, particularly the (uniform) *coercivity*.
- The quadrature rule should be chosen such that the *discretization error* and the *consistency error* are in the same size of magnitude (*discrepancy principle*).

In the following, we consider the standard discretization of the Poisson problem with Dirichlet boundary conditions, i.e., $V = H_0^1(\Omega)$, $V_h = \widehat{V}_h \circ F^{-1}$ and $\widehat{V}_h = S_{p,\boldsymbol{\Xi}}(\widehat{\Omega})$,

$$\begin{aligned} a(u,v) &= \int_\Omega \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \,\mathrm{d}\mathbf{x} \\ &= \int_{\widehat{\Omega}} \nabla^T \widehat{u}(\widehat{\mathbf{x}}) \underbrace{\big(\nabla F(\widehat{\mathbf{x}})\big)^{-T} \big(\nabla F(\widehat{\mathbf{x}})\big)^{-1} |\det \nabla F(\widehat{\mathbf{x}})|}_{G(\widehat{\mathbf{x}}) :=} \nabla \widehat{v}(\widehat{\mathbf{x}}) \,\mathrm{d}\widehat{\mathbf{x}} \\ &= \sum_{\mathbf{i}} \sum_{\alpha,\beta=1}^{d} \int_{\widehat{I}_{\mathbf{i}}} \nabla^T \widehat{u}(\widehat{\mathbf{x}})\, G(\widehat{\mathbf{x}})\, \nabla \widehat{u}(\widehat{\mathbf{x}}) \,\mathrm{d}\widehat{\mathbf{x}}, \end{aligned}$$

where $\widehat{u} = u \circ F$ and $\widehat{v} = v \circ F$. The discrete bilinear form a_h is obtained from the last representation of a, where the integrals are replaced by the quadrature rule $Q_{\omega,\mathbf{t}}$, i.e.,

$$a_h(u,v) = \sum_{\mathbf{i}} \frac{2^d}{|\widehat{I}_{\mathbf{i}}|} Q_{\omega,\mathbf{t}} \left(\nabla^T \widehat{u}(\psi_{\mathbf{i}}(\widehat{\mathbf{x}}))\, G(\psi_{\mathbf{i}}(\widehat{\mathbf{x}}))\, \nabla \widehat{u}(\psi_{\mathbf{i}}(\widehat{\mathbf{x}})) \right),$$

where $|\widehat{I}_{\mathbf{i}}| = (\overline{x}^{(1)} - \underline{x}^{(1)}) \cdots (\overline{x}^{(d)} - \underline{x}^{(d)})$ is the area of $\widehat{I}_{\mathbf{i}} = (\underline{x}^{(1)}, \overline{x}^{(1)}) \times \cdots \times (\underline{x}^{(d)}, \overline{x}^{(d)})$ and

$$\psi_{\mathbf{i}}(\widehat{\mathbf{x}}) := \big(\tfrac{1}{2}(\underline{\mathbf{x}} + \overline{\mathbf{x}}) + \widehat{\mathbf{x}} \tfrac{1}{2}(\overline{\mathbf{x}} - \underline{\mathbf{x}}) \big)$$

maps $(-1,1)$ to $\widehat{I}_{\mathbf{i}}$.

Theorem 5.6. *Provided that we have a quadrature rule with positive weights and which is is exact for polynomials of degree $2p$ and that $\lambda_{\min}(G) > 0$. Then, the bilinear form a_h is coercive and bounded with*

$$\underline{\mu} = \frac{1}{c_G} \underline{\mu}_0 \quad \text{and} \quad \overline{\mu} = c_G \overline{\mu}_0, \tag{5.2}$$

where

$$c_G = \max_{\boldsymbol{i}} \|(\lambda_{min}(G))^{-1}\|_{L_\infty(\widehat{I}_i)} \|\lambda_{max}(G)\|_{L_\infty(\widehat{I}_i)} \tag{5.3}$$

and $\underline{\mu}_0$ and $\overline{\mu}_0$ are the corresponding constants from the Lax Milgram theorem.

Note that Gauss-quadrature with $m = p+1$ is exact up to degree $2p+1$ (Theorem 5.2) and yields positive weighs (Lemma 5.4).

Also the condition $\lambda_{\min}(G) > 0$ is satisfied for our model problem. Note that the factor c_G only depends on the *local* variation of G within each element, i.e., we obtain $\underline{\mu} \to \underline{\mu}_0$ and $\overline{\mu} \to \overline{\mu}_0$ for $h \to 0$.

Proof. For any $u_h \in V_h$, we have

$$\begin{aligned} a_h(u_h,u_h) &= \sum_{\mathbf{i}} \frac{2^d}{|\widehat{I}_{\mathbf{i}}|} Q_{\omega,\mathbf{t}} \left(\nabla^T \widehat{u}_h(\psi_{\mathbf{i}}(\widehat{\mathbf{x}}))\, G(\psi_{\mathbf{i}}(\widehat{\mathbf{x}}))\, \nabla \widehat{u}_h(\psi_{\mathbf{i}}(\widehat{\mathbf{x}})) \right) \\ &= \sum_{\mathbf{i}} \frac{2^d}{|\widehat{I}_{\mathbf{i}}|} \sum_{j=1}^{m} \omega_j\, \nabla^T \widehat{u}_h(\psi_{\mathbf{i}}(\mathbf{t}_j))\, G(\psi_{\mathbf{i}}(\mathbf{t}_j))\, \nabla \widehat{u}_h(\psi_{\mathbf{i}}(\mathbf{t}_j)). \end{aligned}$$

Since all $\omega_j \geq 0$ and since $w^T G w \geq \lambda_{min}(G) w^T w$, we obtain further

$$\begin{aligned} a_h(u_h, u_h) &\geq \sum_{\mathbf{i}} \frac{2^d}{|\widehat{I}_{\mathbf{i}}|} \sum_{j=1}^{m} \omega_j \lambda_{min}(G(\psi_{\mathbf{i}}(\mathbf{t}_j))\ \nabla^T \widehat{u}_h(\psi_{\mathbf{i}}(\mathbf{t}_j))\ \nabla \widehat{u}_h(\psi_{\mathbf{i}}(\mathbf{t}_j)) \\ &\geq \sum_{\mathbf{i}} \frac{2^d}{|\widehat{I}_{\mathbf{i}}|} \|(\lambda_{min}(G))^{-1}\|^{-1}_{L_\infty(\widehat{I}_{\mathbf{i}})} \sum_{j=1}^{m} \omega_j\ \nabla^T \widehat{u}_h(\psi_{\mathbf{i}}(\mathbf{t}_j))\ \nabla \widehat{u}_h(\psi_{\mathbf{i}}(\mathbf{t}_j)). \end{aligned}$$

Since the chosen quadrature rule can resolve the integral of a polynomial of degree $2p$ exactly, we obtain

$$\begin{aligned} a_h(u_h, u_h) &\geq \sum_{\mathbf{i}} \frac{2^d}{|\widehat{I}_{\mathbf{i}}|} \|(\lambda_{min}(G))^{-1}\|^{-1}_{L_\infty(\widehat{I}_{\mathbf{i}})} \int_{(-1,1)^d} \nabla^T \widehat{u}_h(\psi_{\mathbf{i}}(\mathbf{t}))\ \nabla \widehat{u}_h(\psi_{\mathbf{i}}(\mathbf{t}))\, \mathrm{d}\mathbf{t} \\ &= \sum_{\mathbf{i}} \|(\lambda_{min}(G))^{-1}\|^{-1}_{L_\infty(\widehat{I}_{\mathbf{i}})} \int_{\widehat{I}_{\mathbf{i}}} \nabla^T \widehat{u}_h(\widehat{\mathbf{x}})\ \nabla \widehat{u}_h(\widehat{\mathbf{x}})\, \mathrm{d}\widehat{\mathbf{x}} \\ &\geq \sum_{\mathbf{i}} \|(\lambda_{min}(G))^{-1}\|^{-1}_{L_\infty(\widehat{I}_{\mathbf{i}})} \|\lambda_{max}(G)\|^{-1}_{L_\infty(\widehat{I}_{\mathbf{i}})} \int_{\widehat{I}_{\mathbf{i}}} \nabla^T \widehat{u}_h(\widehat{\mathbf{x}})\ G(x) \nabla \widehat{u}_h(\widehat{\mathbf{x}})\, \mathrm{d}\widehat{\mathbf{x}} \\ &\geq \left(\min_{\mathbf{i}} \|(\lambda_{min}(G))^{-1}\|^{-1}_{L_\infty(\widehat{I}_{\mathbf{i}})} \|\lambda_{max}(G)\|^{-1}_{L_\infty(\widehat{I}_{\mathbf{i}})} \right) |u|^2_{H_1(\Omega)} \\ &\geq \left(\max_{\mathbf{i}} \|(\lambda_{min}(G))^{-1}\|_{L_\infty(\widehat{I}_{\mathbf{i}})} \|\lambda_{max}(G)\|_{L_\infty(\widehat{I}_{\mathbf{i}})} \right)^{-1} \underline{\mu}_0 \|u\|^2_{H_1(\Omega)}. \end{aligned}$$

Using the same arguments, we also obtain

$$a_h(u_h, u_h) \leq \left(\min_{\mathbf{i}} \|(\lambda_{min}(G))^{-1}\|_{L_\infty(\widehat{I}_{\mathbf{i}})} \|\lambda_{max}(G)\|_{L_\infty(\widehat{I}_{\mathbf{i}})} \right) \overline{\mu}_0 \|u\|^2_{H_1(\Omega)}.$$

Note that we can apply Cauchy-Schwarz inequality to a_h, so we have

$$\begin{aligned} a_h(u_h, v_h) &\leq a_h(u_h, u_h)^{1/2} a_h(v_h, v_h)^{1/2} \\ &\leq \left(\min_{\mathbf{i}} \|(\lambda_{min}(G))^{-1}\|_{L_\infty(\widehat{I}_{\mathbf{i}})} \|\lambda_{max}(G)\|_{L_\infty(\widehat{I}_{\mathbf{i}})} \right) \overline{\mu}_0 \|u\|_{H_1(\Omega)} \|v\|_{H_1(\Omega)}, \end{aligned}$$

which finishes the proof. □

This shows that we can apply Strang's Lemma and that the constants $\underline{\mu}$ and $\overline{\mu}$ are well bounded. The next step is give a consistency error estimate. Note that that error estimate should be as good as the discretization error estimate.

For the discretization error estimate, we need results on the *Lagrange interpolation*. For any $u \in H^1(-1,1)$, we define the Lagrange interpolation as follows:

$$w \in \mathbb{P}_{m-1} \quad \text{such that} \quad w(t_i) = u(t_i) \text{ for all } i = 1, \ldots, m, \tag{5.4}$$

i.e., w is the Lagrange interpolation polynomial of u.

Theorem 5.7. *Let $u \in C^m(-1,1)$ and w be the Lagrange interpolation (5.4) with $-1 \leq t_0 \leq \cdots \leq t_m \leq 1$. Then,*

$$|u(x) - w(x)| \leq \frac{\prod_{i=1}^{m}(x - t_i)}{m!} \sup_{\xi \in (-1,1)} |u^{(m)}(\xi)| \quad \textit{for all} \quad x \in \mathbb{R}.$$

Proof. There is certainly some function q such that

$$u(x) - w(x) = \prod_{i=1}^{m}(x - t_i)q(x).$$

Let $x \in \mathbb{R}$ be arbitrary but fixed. Define

$$g(t) := u(t) - w(t) - \prod_{i=1}^{m}(t - t_i)\underbrace{q(x)}_{\text{this is } q(x), \text{ not } q(t).}.$$

The function g has $m+1$ roots at: $t_1, \ldots, t_m$ and x. Using the mean value theorem, there is some $\xi \in (\min\{x, t_1\}, \max\{x, t_m\}) \subseteq (-1, 1)$ such that $g^{(m)}(\xi) = 0$. For this value of ξ, we have

$$0 = g^{(m)}(\xi) = u^{(m)}(\xi) - \underbrace{w^{(m)}(\xi)}_{=0} - \underbrace{\frac{\mathrm{d}^m}{\mathrm{d}\xi^m}\prod_{i=1}^{m}(\xi - t_i)}_{=\frac{\mathrm{d}^m}{\mathrm{d}\xi^m}\xi^m = m!}\, q(x),$$

which shows $u(x) - w(x) = \prod_{i=1}^{m}(x - t_i)q(x) = \frac{\prod_{i=1}^{m}(x-t_i)}{m!}u^{(m)}(\xi)$. The desired result is obtained by taking the supremum. □

We immediately obtain that there is some constant $c(m)$ such that

$$\|u - w\|_{L_2(-1,1)} \leq \sqrt{2}\|u - w\|_{L_\infty(-1,1)} \leq c(m)\|u\|_{W_\infty^m(-1,1)}. \tag{5.5}$$

Theorem 5.8. *Let $m, k \in \mathbb{N}$ with $k \leq m$ and let $Q_{\omega,t}$ be a quadrature formula with m quadrature nodes which is exact for polynomials up to degree $2m-1$. Then,*

$$\left|\int_{-1}^{1} g(x)u(x)v(x)\,\mathrm{d}x - Q_{\omega,t}(guv)\right| \leq c(m)\|g\|_{W_\infty^m(-1,1)}\|u\|_{H^k(-1,1)}\|v\|_{L_2(-1,1)}.$$

holds for all $g \in W_\infty^m(-1,1)$, $u \in \mathbb{P}_m$ and $v \in \mathbb{P}_m$.

Proof. Let $w \in \mathbb{P}_{m-1}$ be such that

$$w(t_j) = g(t_j)u(t_j) \qquad \text{for all} \qquad j = 1, \ldots, m.$$

Using this setting, we obtain $Q_{\omega,t}(gu) = Q_{\omega,t}(w)$ and, consequently, $Q_{\omega,t}(guv) = Q_{\omega,t}(wv)$. Since $wv \in \mathbb{P}_{2m-1}$, we know that the quadrature rule is exact, i.e., we obtain using the Cauchy-Schwarz inequality

$$\begin{aligned}\left|\int_{-1}^{1} g(x)u(x)v(x)\,\mathrm{d}x - Q_{\omega,t}(guv)\right| &= \left|\int_{-1}^{1} g(x)u(x)v(x) - w(x)v(x)\,\mathrm{d}x\right| \\ &\leq \|v\|_{L_2(-1,1)}\|gu - w\|_{L_2(-1,1)}.\end{aligned}$$

Using the Lagrange interpolation error estimate (5.5), we further obtain

$$\left|\int_{-1}^{1} g(x)u(x)v(x)\,\mathrm{d}x - Q_{\omega,t}(guv)\right| \leq c_0(m)\|v\|_{L_2(-1,1)}\|gu\|_{W_\infty^m(-1,1)}.$$

Using the product rule, we further obtain

$$\left| \int_{-1}^{1} g(x)u(x)v(x)\,\mathrm{d}x - Q_{\omega,t}(guv) \right| \leq c_1(m)\|v\|_{L_2(-1,1)}\|g\|_{W_\infty^m(-1,1)}\|u\|_{W_\infty^m(-1,1)}.$$

Now, using a trace estimate, we obtain $\|q\|_{L_\infty(-1,1)} \leq c\|q\|_{H^1(-1,1)}$ and therefore,

$$\left| \int_{-1}^{1} g(x)u(x)v(x)\,\mathrm{d}x - Q_{\omega,t}(guv) \right| \leq c_2(m)\|v\|_{L_2(-1,1)}\|g\|_{W_\infty^m(-1,1)}\|u\|_{H^{m+1}(-1,1)}.$$

Since u is a polynomial of degree m, we have $\|u\|_{H^{m+1}(-1,1)} = \|u\|_{H^m(-1,1)}$. This yields the desired result for $k = m$. For $k < m$, a standard inverse estimate (Theorem 4.1) yields the desired result. □

Let $k, p, m \in \mathbb{N}_0$ be such that $0 \leq k < p + 1 \leq m$. Using standard scaling arguments, we obtain

$$\left| \int_{-1}^{1} g(x)u(x)v(x)\,\mathrm{d}x - Q_{\omega,t}(guv) \right| \leq c(m)h^k\|g\|_{W_\infty^m(0,1)}\|u\|_{H^k(0,1)}\|v\|_{L_2(0,1)}$$

for splines $u, v \in S_{p,\Xi}(0,1)$. Again, the result can be extended to tensor-product splines. We obtain

$$\left| \int_{\widehat{\Omega}} g(\mathbf{x})u(\mathbf{x})v(\mathbf{x})\,\mathrm{d}\mathbf{x} - Q_{\omega,t}(guv) \right| \leq c(m)h^k\|g\|_{W_\infty^m(\widehat{\Omega})}\|u\|_{H^k(\widehat{\Omega})}\|v\|_{L_2(\widehat{\Omega})}.$$

for splines $u \in S_{\mathbf{p},\boldsymbol{\Xi}}(\widehat{\Omega})$ and $v \in S_{\mathbf{p},\boldsymbol{\Xi}}(\widehat{\Omega})$. Using this estimate, we can show a statement as follows.

Theorem 5.9. *Let $k, p, m \in \mathbb{N}_0$ be such that $0 \leq k < p + 1 \leq m$. Assume that we have a quadrature rule with m positive weights such that the quadrature formula is exact for polynomials of degree $2m - 1$. Then,*

$$|a(u_h, v_h) - a_h(u_h, v_h)| \leq \underbrace{\max_{\boldsymbol{i}} c_{G,\boldsymbol{i}}}_{c_G :=} h^k\|u_h\|_{H^{k+1}(\Omega)}\|v_h\|_{H^1(\Omega)}$$

for all $u_h, v_h \in V_h$, where $c_{G,\boldsymbol{i}}$ only depends on $G|_{I_{\boldsymbol{i}}}$, m, p and k, but is independent of h.

Here, we obtain $\|u_h\|_{H^{k+1}(\Omega)}$ and $\|v_h\|_{H^1(\Omega)}$ instead of $\|u_h\|_{H^k(\Omega)}$ and $\|v_h\|_{L_2(\Omega)}$ form above since we get one additional derivative from the derivative in the definition of the bilinear form a.

By plugging this result into the estimate of Strang's lemma and by assuming that there was no quadrature error for f, we obtain using a standard inverse estimate

$$\begin{aligned}
\|u - u_h\|_{H^1(\Omega)} &\lesssim \inf_{v_h \in V_h} \left(\|u - v_h\|_{H^1(\Omega)} + h^k c_G \|v_h\|_{H^{k+1}(\Omega)} \right) \\
&\leq h^k c_G \|u\|_{H^{k+1}(\Omega)} + \inf_{v_h \in V_h} \left(\|u - v_h\|_{H^1(\Omega)} + h^k c_G \|u - v_h\|_{H^{k+1}(\Omega)} \right) \\
&\lesssim h^k c_G \|u\|_{H^{k+1}(\Omega)} + (1 + c_G) \inf_{v_h \in V_h} \|u - v_h\|_{H^1(\Omega)} \\
&\lesssim h^k c_G \|u\|_{H^{k+1}(\Omega)} + (1 + c_G) h^k |u|_{H^{1+k}(\Omega)} \lesssim h^k (1 + c_G) \|u\|_{H^{k+1}(\Omega)}.
\end{aligned}$$

Consistency error estimates for f can be derived analogous.

5.4 Assembling costs and sum factorization

In the following, we discuss the costs of the assembling approaches. For simplicity, we restrict ourselves to computing the mass matrix.

We assume to have a tensor-product grid with n elements in each direction, spline degree p and smoothness $k = p - 1$.

We use the following notation.

- The computational complexity is measured in floating point operations (flops), i.e., additions, multiplications, etc.
- The computational complexity is expressed in terms of the spline degree p and $n \eqsim h^{-1}$, the number of elements per direction. The spacial dimension d is only written in the exponent, not as multiplicative factor.

  More precisely, we write that the number of flops $\mathfrak{C}_{n,p}$ satisfies

  $$\mathfrak{C}_{n,p} = \mathcal{O}(n^\alpha p^\beta)$$

  if there are a constants $c > 0$, $c_n > 0$ and $c_p > 0$, independent of h and p such that

  $$\mathfrak{C}_{n,p} \leq c n^\alpha p^\beta \quad \text{for all } n \geq c_n \text{ and } p \geq c_p.$$

- The *evaluation* of the B-spline basis functions *cannot* be done in $\mathcal{O}(1)$ flops. However, it is possible to evaluate the basis functions at all quadrature nodes of interest in a pre-processing step.

  If the basis functions and the quadrature nodes follow a tensor-product structure, we simply use

  $$\widehat{B}_{\mathbf{i},\mathbf{p},\mathbf{h}}(\mathbf{t_k}) = \underbrace{\widehat{B}_{i_1,p_1,h_1}(t_{k_1})}_{\phi_{i_1}(t_{k_1}) :=} \cdots \underbrace{\widehat{B}_{i_d,p_d,h_d}(t_{k_d})}_{\phi_{i_d}(t_{k_d}) :=}$$

  and pre-compute the values of the univariate splines $\phi_{i_j}(t_{k_j})$.

  As we assume that the splines have been pre-evaluated, we assume the costs to access that value of a basis function to be $\mathcal{O}(1)$.

- We moreover assume that the geometry function can be evaluated in $\mathcal{O}(1)$ flops. This is possible for simple geometry functions. For more complicated geometry mappings, like B-spline surfaces, this is not possible, cf. Remark 5.10.

Before we proceed, we remember that the number of non-zero entries of the mass matrix is

$$\mathcal{O}(p^d n^d).$$

The most naive approach to assemble the mass matrix using Gauss quadrature is to compute each matrix entry separately. So for each pair $(\mathbf{i}, \mathbf{j})$, the quadrature is done as follows. We first determine the intersection of the supports of the basis functions $\widehat{B}_{\mathbf{i},\mathbf{p},\mathbf{h}}$ and $\widehat{B}_{\mathbf{j},\mathbf{p},\mathbf{h}}$. The intersection of the supports is at least 1 element $I_{\mathbf{k}}$ and at most $\mathcal{O}(p^d)$ elements.[1]

[1] Note that the overall costs are dominated by the case with $\mathcal{O}(p^d)$ elements.

Within each element, we have $\mathcal{O}(p^d)$ quadrature nodes. Therefore, we have $\mathcal{O}(p^{2d})$ quadrature nodes in the intersection of the supports. The overall costs are

$$\#\text{ non-zeros } * \#\text{ quadrature points in intersection of support},$$

i.e.,

$$\mathfrak{C}_{n,p} = \mathcal{O}(p^{3d} n^d).$$

Numerical *experiments* show that the *assembling costs are a big issue in IgA*. Particularly, this approach of naive Gauss quadrature is a way to slow. So, better approaches are required.

The first approach which we consider, is *Sum factorization*. This approach is already known from FEM, where it is typically only applied element-wise. Consider for simplicity only the two-dimensional case. Let entries of the mass matrix M_h be denoted by

$$m_{i,j} = m_{i_1+n(i_2-1),i_1+n(i_2-1)} = c_{(i_1,i_2),(j_1,j_2)} = c_{\mathbf{i},\mathbf{j}}.$$

Assume that the mass matrix is computed with a Gauss quadrature rule. In the last section, we have written down the quadrature rules only element-wise but the idea of quadrature can be applied also to the whole spline space.

Consider the the univariate case first. Provided, we choose to have $p+1$ quadrature nodes per element, we obtain $(p+1)n$ quadrature nodes in total. We denote the nodes by $t_1, \dots, t_{(p+1)n}$ and the corresponding weights by $\omega_1, \dots, \omega_{(p+1)n}$.

For the multivariate case, we use again standard tensor-product quadrature:

$$\begin{aligned}
m_{i,j} &= \sum_{k_1=1}^{(p+1)n} \sum_{k_2=1}^{(p+1)n} \omega_{k_1}\omega_{k_2}\, g(t_{k_1}, t_{k_2})\, \widehat{B}_{\mathbf{i},\mathbf{p},\mathbf{h}}(\mathbf{t_k}) \widehat{B}_{\mathbf{j},\mathbf{p},\mathbf{h}}(\mathbf{t_k}) \\
&= \sum_{k_1=1}^{(p+1)n} \sum_{k_2=1}^{(p+1)n} \omega_{k_1}\omega_{k_2} g(t_{k_1}, t_{k_2}) \phi_{i_1}(t_{k_1}) \phi_{i_2}(t_{k_2}) \phi_{j_1}(t_{k_1}) \phi_{j_2}(t_{k_2}) \\
&= \underbrace{\sum_{k_1=1}^{(p+1)n} \omega_{k_1} \phi_{i_1}(t_{k_1}) \phi_{j_1}(t_{k_1}) \underbrace{\sum_{k_2=1}^{(p+1)n} \omega_{k_2} \phi_{i_2}(t_{k_2}) \phi_{j_2}(t_{k_2}) \underbrace{g(t_{k_1}, t_{k_2})}_{a_{k_1,k_2} :=}}_{b_{k_1,(i_2,j_2)} :=}}_{c_{(i_1,j_1),(i_2,j_2)} :=}
\end{aligned}$$

For computing the mass matrix, we perform the following steps.

- Compute a_{k_1,k_2}: We have $(p+1)n$ quadrature nodes per direction, so we have to consider $(p+1)^2 n^2$ quadrature nodes in total. For simplicity, we have assumed that the evaluation of the geometry function can be done on $\mathcal{O}(1)$ flops. So, the costs are

$$\mathcal{O}(p^2 n^2).$$

- Compute $b_{k_1,(i_2,j_2)}$: Again, we have $(p+1)n$ quadrature nodes (choices of k_1). Moreover, we have $n+p$ basis functions i_2. For each fixed i_2, there are not more than $2p+1$ basis functions j_2 such that the supports of the basis functions are not disjoint. So, we

have $\mathcal{O}(p^2 n(n+p))$ coefficients b to compute. For each particular coefficient, we have to take the sum over p^2 quadrature nodes (choices of k_1). So, the overall costs are

$$\mathcal{O}(p^4 n(n+p))$$

- Compute $c_{(i_1,j_1),(i_2,j_2)}$: Here, we have $(n+p)(2p+1)$ pairs (i_1, i_2) and as many pairs (j_1, j_2). So, we have $\mathcal{O}(p^2(n+p)^2)$ coefficients c to compute. For each particular coefficient, we have to take the sum over p^2 quadrature nodes (choices of k_2). So, the overall costs are
$$\mathcal{O}(p^4(n+p)^2).$$

The overall costs follow the dominant summand. Thus, we obtain

$$\mathfrak{C}_{n,p} = \mathcal{O}(p^4(n+p)^2).$$

If we consider the d-dimensional case, we obtain

$$\mathfrak{C}_{n,p} = \mathcal{O}(p^{d+2}(n+p)^d).$$

Provided the usual case $n \geq p$, we have

$$\mathfrak{C}_{n,p} = \mathcal{O}(p^{d+2} n^d).$$

Sometimes, sum factorization is provided *element-wise*. Here, we perform for each of the n^d elements the following steps.

- Compute a_{k_1,k_2}: We have $p+1$ quadrature nodes per direction, so we have to consider $(p+1)^2$ quadrature nodes in total. For simplicity, we have assumed that the evaluation of the geometry function can be done on $\mathcal{O}(1)$ flops. So, the costs are
$$\mathcal{O}(p^2).$$

- Compute $b_{k_1,(i_2,j_2)}$: Again, we have $p+1$ quadrature nodes (choices of k_1). Moreover, we have $2p+1$ basis functions i_2 and j_2. So, we have $\mathcal{O}(p^3)$ coefficients b to compute. For each particular coefficient, we have to take the sum over p quadrature nodes (choices of k_1). So, the overall costs are
$$\mathcal{O}(p^4)$$

- Compute $c_{(i_1,j_1),(i_2,j_2)}$: Here, we have $(2p+1)^2$ pairs (i_1, i_2) and as many pairs (j_1, j_2). So, we have $\mathcal{O}(p^4)$ coefficients c to compute. For each particular coefficient, we have to take the sum over p quadrature nodes (choices of k_2). So, the overall costs are
$$\mathcal{O}(p^5).$$

Thus, the costs per element are $\mathcal{O}(p^5)$ or, for the d-dimensional case

$$\mathcal{O}(p^{2d+1}).$$

Since we have to apply quadrature for each element, the overall costs are obtained if the per-element costs are multiplied with the number of elements:

$$\mathfrak{C}_{n,p} = \mathcal{O}(p^{2d+1}n^d).$$

This is the result that has been obtained in [15]. We observe that the per-element sum factorization is more expensive than the global sum factorization. A careful analysis shows that one can do a box-wise quadrature. Here we always combine $p \times \cdots \times p$ elements to a box and apply our sum-factorization algorithm box-wise. The costs per box are

$$\mathcal{O}(p^{2d+2}).$$

The number of boxes is $\mathcal{O}((n/p)^d)$. Thus, the overall costs are

$$\mathfrak{C}_{n,p} = \mathcal{O}(p^{d+2}n^d),$$

which coincides with the case of global sum factorization.

Remark 5.10. *If the geometry function is a B-spline surface (or volume), the evaluation of the geometry function on all quadrature nodes is as expensive as the assembling procedure itself. Similar to assembling, also the evaluation of the geometry function can be done using element-wise, box-wise or global sum factorization, see [16] for details.*

5.5 Weighted quadrature

For simplicity, consider the one dimensional case first. Consider again the problem of deriving the coefficients $m_{i,j}$ of the mass matrix, representing

$$\int_0^1 g(x)\phi_i(x)\phi_j(x)\,\mathrm{d}x,$$

where g is the Jacobi determinant of the geometry function and ϕ_i and ϕ_j are the basis function of $\widehat{V}_h = S_{p,\Xi}(0,1)$. For the computation of the computational complexity, we assume to have n elements per direction and no repeated knots.

Now, we introduce for each row i of the mass matrix a separate quadrature rule:

$$Q_i(\psi) := \sum_{l=1}^{N} \omega_{i,l}\psi(t_i),$$

which aims to resolve the integral

$$\int_0^1 \phi_i(x)\psi(x)\,\mathrm{d}x.$$

We choose $\psi := g\phi_j$ as the function to be integrated. The function ϕ_i is part of the quadrature weights. (The quadrature nodes for all quadrature rules Q_i are the same.)

We require the following exactness condition:

$$Q_i(v_h) = \int_{\widehat{\Omega}} \phi_i(x)v_h(x)\,\mathrm{d}x \qquad \text{for all} \qquad v_h \in \widehat{V}_h. \tag{5.6}$$

This is similar to the Gauss quadrature, where we had also required exactness for the case that $g = 1$.

Moreover, we require the following locality condition:

$$\phi_i(t_l) = 0 \quad \Rightarrow \quad \omega_{i,l} = 0. \tag{5.7}$$

The exactness condition (5.6) contains $n + p$ conditions. This means, that the work could be done theoretically with $n + p$ quadrature nodes. (The Gauss quadrature approach uses pn quadrature nodes.)

To have some more freedom, in weighted quadrature, one uses 2 quadrature per direction. Since the nodes are also located on the element-boundary, we have basically 3^d quadrature nodes that belong to the closure of the element. For the elements on the boundary, $p + 1$ quadrature nodes per direction are used. In total, we obtain $2n + 2p + 1$ quadrature nodes, cf. Figure 5.1.

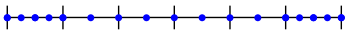


Figure 5.1: Quadrature nodes of weighted quadrature

It is outlined in [17], that it is possible to find a quadrature rule Q_i such that (5.6) and (5.7) hold. Since these conditions do not yield exactness, a quadrature rule with minimum norm is chosen.

The quadrature is now done using the quadrature rules Q_i:

$$m_{i,j} = Q_i(g(x)\phi_j(x)) = \sum_{l=1}^{2n+2p+1} \omega_{i,l}\phi_j(t_l).$$

The extension to tensor-product discretizations is straight forward. First, a tensor-product grid is introduced, cf. Figure 5.2.

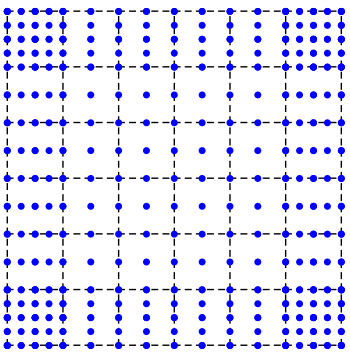


Figure 5.2: Quadrature nodes of weighted quadrature

The quadrature is now rather straight forward:

$$\begin{aligned} m_{i_1+n(i_2-1),j_1+n(j_2-1)} &= \sum_{l_1=1}^{2n+2p+1} \sum_{l_2=1}^{2n+2p+1} \omega_{i_1,l_1}\omega_{i_2,l_2} g(t_{l_1}, t_{l_2}) \phi_{j_1}(t_{l_1}) \phi_{j_2}(t_{l_2}) \\ &\eqsim \int_{\widehat{\Omega}} g(x_1,x_2) \underbrace{\phi_{i_1}(x_1)\phi_{i_2}(x_2)}\underbrace{\phi_{j_1}(x_1)\phi_{j_2}(x_2)} \,\mathrm{d}(x_1,x_2). \end{aligned}$$

This integral is again resolved using sum factorization. A careful analysis shows that the overall computational complexity is

$$\mathcal{O}(p^{d+1}n)$$

flops.

Some remarks:

- The *stiffness matrix* can be handled analogously, e.g., by computing the integrals $\int_{\widehat{\Omega}} (\frac{\partial}{\partial x}u)(\frac{\partial}{\partial x}v)\,\mathrm{d}\mathbf{x}$ and $\int_{\widehat{\Omega}} (\frac{\partial}{\partial y}u)(\frac{\partial}{\partial y}v)\,\mathrm{d}\mathbf{x}$ separately.
- The method treats the ansatz functions different to the test functions. This approach yields non-symmetric matrices. (If the g was constant, all integrals were exact. In this case, the overall matrix was certainly symmetric.)
- There is a detailed error analysis (based on Strang's lemma) which also takes into account that the matrices are non-symmetric, cf. [17].
- The resulting matrix could be symmetrized, i.e., it could be replaced by $\frac{1}{2}(M_h + M_h^T)$. In this case, the error analysis from [17] breaks apart. So, one has to stick to the original, non-symmetric matrix.

5.6 Low-tensor-rank quadrature

Consider once more the mass matrix $M_h = [m_{i,j}]_{i,j=1}^{n^2}$ first. Here, we first discuss the case that $g = 1$. For $d = 2$, we have

$$\begin{aligned} m_{i_1+(i_2-1)n,j_1+(j_2-1)n} &= \int_{\widehat{\Omega}} \overbrace{\phi_{i_1}(x_1)\phi_{i_2}(x_2)}\overbrace{\phi_{j_1}(x_1)\phi_{j_2}(x_2)} \,\mathrm{d}(x_1,x_2) \\ &= \underbrace{\int_0^1 \phi_{i_1}(x_1)\phi_{j_1}(x_1)\,\mathrm{d}x_1}_{m^{(1)}_{i_1,j_1} :=} \underbrace{\int_0^1 \phi_{i_2}(x_2)\phi_{j_2}(x_2)\,\mathrm{d}x_2}_{m^{(2)}_{i_2,j_2} :=}. \end{aligned}$$

The coefficients $m^{(1)}_{i_1,j_1}$ and $m^{(2)}_{i_2,j_2}$ can be combined to *univariate mass matrices*:

$$M_h^{(1)} = [m^{(1)}_{i,j}]_{i,j=1}^N \qquad \text{and} \qquad M_h^{(2)} = [m^{(2)}_{i,j}]_{i,j=1}^N.$$

The overall mass matrix M_h is just the Kronecker product:

$$M_h = M_h^{(1)} \otimes M_h^{(2)} \tag{5.8}$$

or

$$M_h = [m_{i,j}]_{i,j=1}^{N^2} \qquad \text{with} \qquad m_{i_1+(i_2-1)n,j_1+(j_2-1)n} = m^{(1)}_{i_1,j_1} m^{(2)}_{i_2,j_2}.$$

Such a Kronecker product structure has many advantages. Instead of assembling the mass matrix M_h as a whole, we can assemble the mass matrices $M_h^{(1)}$ and $M_h^{(2)}$.

Each of those matrices has

$$\mathcal{O}(np)$$

non-zero entries. The costs for assembling each of those matrices (using Gauss quadrature) are

$$\mathcal{O}(np^3)$$

flops, which is very small compared to the assembling costs for assembling the matrix M_h as a whole. After assembling the univariate mass matrices, we can derive the Kronecker product to obtain M_h. The computational complexity of derivation of the Kronecker product is

$$\mathcal{O}(n^2p^2),$$

i.e., as large as the number of non-zero entries of M_h.

In many contexts, we can work with *matrix-free approaches*. Often, we do not need the matrix M_h at all, but only the possibility to compute matrix-vector products $M_h\underline{u}_h$. The computation of these matrix-vector products can be evaluated using the formula

$$M_h\underline{u}_h = (M_h^{(1)} \otimes M_h^{(2)})\underline{u}_h = \underbrace{(M_h^{(1)} \otimes I)\underbrace{(I \otimes M_h^{(2)})\underline{u}_h}_{\underline{v}_h :=}}_{\underline{w}_h :=}.$$

Here, the computation of $\underline{v}_h$ can be realized by n matrix-vector products of the matrix $M_h^{(2)}$ and appropriate blocks of $\underline{u}_h$. The vector $\underline{w}_h$ can be computed analogously from $\underline{v}_h$. Using this approach, the computational complexity is only

$$\mathcal{O}(pn^d),$$

compared to

$$\mathcal{O}(p^dn^d)$$

for the straight-forward approach.

The idea of low-tensor rank assembling easily extends to the *stiffness matrix*. Here, we obtain

$$A_h = A_h^{(1)} \otimes M_h^{(2)} + M_h^{(1)} \otimes A_h^{(2)}, \tag{5.9}$$

where $A_h^{(1)}$ and $A_h^{(2)}$ are the univariate stiffness matrices and $M_h^{(1)}$ and $M_h^{(2)}$ are the univariate mass matrices.

For general geometry mappings, we can observe that the geometry function typically has a low-tensor-rank structure:

$$F(x,y) \approx \sum_{l=1}^{r} f_l^{(1)}(x) f_l^{(2)}(y)$$

for some (not too large) value of r. Since we are only interested in the values at the quadrature nodes (t_{i_1}, t_{i_2}), we can formalize this in a discrete setting. Consider the vector $\underline{F}_h$, given by

$$\underline{F}_h = (f_i)_{i=1}^N \quad \text{with} \quad f_{i_1+n(i_2-1)} = F(t_{i_1}, t_{i_2}).$$

A low-tensor-rank representation has the form

$$\underline{F}_h \approx \sum_{l=1}^{r} \underline{f}_l^{(1)} \otimes \underline{f}_l^{(2)},$$

where $\underline{f}_l^{(1)}, \underline{f}_l^{(2)} \in \mathbb{R}^n$. In the discrete setting, we know that such a decomposition exists certainly for $r = n$. A low-tensor-rank formulation has a smaller tensor rank.

Provided to have a low-tensor-rank representation of F, we also have a low-tensor-rank approximation of the Jacobi matrix (or its individual entries). For assembling the mass matrix, we are interested in a low-tensor-rank approximation of the determinant of the Jacobi matrix. Having a representation of the Jacobi matrix with tensor rank r, we can show that there is a representation of its determinant that is not larger than $d\,r^2$. Numerical experiments show that this, however, is too pessimistic. Since the transformation required for the stiffness matrix also contains the inverse of the Jacobi matrix, we have no guarantee on the rank. Also in this case, experiments show that these typically have a low rank.

So, provided we have a function g with low tensor rank, how can we evaluate the integral

$$m_{i,j} = \int_{\widehat{\Omega}} g(\mathbf{x})\varphi_i(\mathbf{x})\varphi_j(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

efficiently?

We use the tensor-product structure

$$\begin{aligned} \varphi_{i_1+n(i_2-1)}(x,y) &= \phi_{i_1}(x)\phi_{i_2}(y) \\ g(x,y) &= \sum_{l=1}^{r} g_l^{(1)}(x) g_l^{(2)}(x) \end{aligned}$$

and obtain

$$\begin{aligned} m_{i_1+n(i_2-1),j_1+n(j_2-1)} &= \sum_{l=1}^{r} \int_{\widehat{\Omega}} g_l^{(1)}(x) g_l^{(2)}(x) \phi_{i_1}(x)\phi_{i_2}(y)\phi_{j_1}(x)\phi_{j_2}(y)\,\mathrm{d}(x,y) \\ &= \sum_{l=1}^{r} \underbrace{\int_0^1 g_l^{(1)}(x)\phi_{i_1}(x)\phi_{j_1}(x)\,\mathrm{d}x}_{m_{l,i_1,j_1}^{(1)} :=} \underbrace{\int_0^1 g_l^{(2)}(y)\phi_{i_2}(y)\phi_{j_2}(y)\,\mathrm{d}y}_{m_{l,i_2,j_2}^{(2)} :=}. \end{aligned}$$

In matrix notation, this yields

$$M = \sum_{l=1}^{r} M_l^{(1)} \otimes M_l^{(2)}.$$

This also shows the assembling procedure:

- Assume to have a low-tensor-representation of g.

- Assemble the univariate matrices $M_l^{(1)}$ and $M_l^{(2)}$.
- Compute the Kronecker product (if desired).

The assembling of the univariate matrices can be done, e.g., with Gauss quadrature. Here, the computational complexity is

$$\mathcal{O}(rp^3n),$$

which is (for n large) much smaller than n^d, the number of degrees of freedom. The computation of the Kronecker product can be done with

$$\mathcal{O}(rp^dn^d)$$

flops, which would be the dominant part. Thus, whenever low-tensor-rank representations are available, it makes much sense to use matrix-free solvers.

5.7 An algebraic low-tensor-rank quadrature

Assume to have a low-tensor-rank matrix

$$M = \sum_{l=1}^{r} A_l \otimes B_l \tag{5.10}$$

with

$$M = (m_{i,j})_{i,j=1}^{n^2} \quad \text{with} \quad m_{i_1+n(i_2-1),j_1+n(j_2-1)} = \mathfrak{m}_{(i_1,i_2),(j_1,j_2)},$$

and $A_l = (a_{i,j}^{(l)})_{i,j=1}^{n}$ and $B_l = (b_{i,j}^{(l)})_{i,j=1}^{n}$. Consider the following reordered matrix

$$\widehat{M} = (\widehat{m}_{i,j})_{i,j=1}^{n^2} \quad \text{with} \quad \widehat{m}_{i,j} = \mathfrak{m}_{i_1+n(j_1-1),i_2+n(j_2-1)},$$

cf. Figure 5.3.

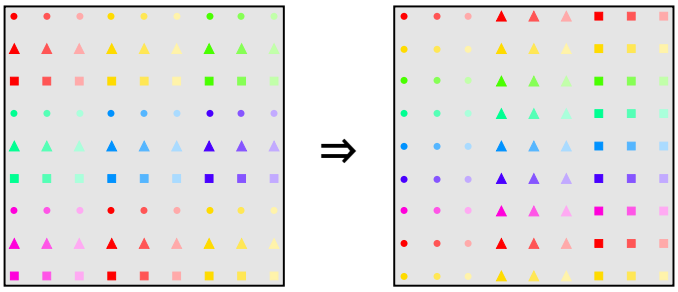


Figure 5.3: Reordering

Similarly, we reorder the entries of A_l and B_l such that they get vectors:

$$\widehat{A}_l = (\widehat{a}_i^{(l)})_{i=1}^{n^2} \quad \text{with} \quad \widehat{a}_{i+n(j-1)}^{(l)} = a_{i,j}^{(l)},$$

$$\widehat{B}_l = (\widehat{b}_i^{(l)})_{i=1}^{n^2} \quad \text{with} \quad \widehat{b}_{i+n(j-1)}^{(l)} = b_{i,j}^{(l)}.$$

Now the low-tensor-rank representation (5.10) reads as follows:

$$\widehat{M} = \sum_{l=1}^{r} \widehat{A}_l^T \widehat{B}_l,$$

i.e., $\widehat{M}$ is a *low-rank* matrix.

There are many (black-box) algorithms around that allow the treatment of low-rank matrices, like

- the singular value decomposition (SVD) or
- the adaptive cross approximation (ACA) algorithm.

The SVD algorithm can be applied to any known (i.e., already assembled) matrix to a representation of the form

$$\widehat{M} = \sum_{l=1}^{n} \sigma_l \widehat{A}_l^T \widehat{B}_l.$$

The best possible low-rank approximation is obtained by only taking the contributions with the largest singular values σ.

The ACA algorithm can be used to get a low-rank representation

$$\widehat{M} = \sum_{l=1}^{r} \widehat{A}_l^T \widehat{B}_l,$$

without the need of knowing the matrix $\widehat{M}$ in advance. So, it is suitable for assembling. The only ingredient is that one has to be able to compute individual matrix entries of the matrix $\widehat{M}$. This can be done, e.g., with Gauss quadrature. The ACA algorithm is beneficial if not too many matrix entries have to be computed.

In numerical experiments, it was shown that this approach works well, cf. [18].

5.8 Literature

For the orthogonal polynomials and Gauss quadrature, see [10]. Sum factorization (Section 5.4) is a known technique in FEM, cf. [19] and others. Sum factorization in the IgA context was first discussed in [15]. Global and box-wise sum factorization for IgA was proposed in [16]. Weighted quadrature (Section 5.5) was proposed and analyzed in [17].

For low-tensor-rank approaches (Section 5.7), see [20]. The corresponding algebraic approaches (Section 5.7) have been proposed in [18].

Chapter 6

Adaptive discretizations in Isogeometric Analysis

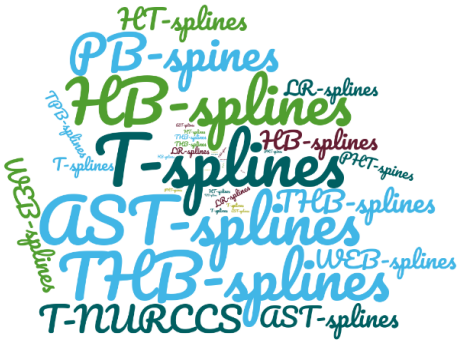


Figure 6.1: Spline zoo

Splines that allow local refinement *include*, but are *not* restricted to:

- **HB-splines:** hierarchical B-splines
- **THB-splines:** truncated hierarchical B-splines
- **T-splines:** splines over meshes with T-junctions
- **AST-splines:** analysis suitable T-splines
- **DCT-splines:** dual compatible T-splines
- **LR-splines:** locally refined splines
- **WEB-splines:** weighted extended B-splines
- **PB-spines:** patchwork B-splines
- **TPB-splines:** truncated patchwork B-splines
- **T-NURCCS:** non-uniform rational Catmull-Clark surfaces with T-junctions
- **PHT-spines:** polynomial splines over hierarchical T-meshes
- **G-spines:** splines with geometric continuity
- **TURBS:** topologically unrestricted rational B-splines
- **MPBES:** multi-patch B-splines with enhanced smoothness

Figure 6.1 visualizes a part of the spline zoo. We will restrict ourselves to THB-splines and T-splines.

6.1 Error estimates for locally refined spaces

We assume to have given some basis $\mathcal{B}$ of piecewise polynomial functions and some dual basis Λ, i.e.,

$$\lambda_{\mathbf{j}}(B_{\mathbf{i}}) = \delta_{\mathbf{i},\mathbf{j}} \text{ for all } B_{\mathbf{i}} \in \mathcal{B} \text{ and } \lambda_{\mathbf{j}} \in \Lambda.$$

Let $\widehat{\Omega}$ be the parameter domain. For simplicity, we only consider that space. We assume that

$$\widehat{\Omega} = \bigcup_k Q_k,$$

where $Q_k = I_k^1 \times \ldots \times I_k^d$ are axis aligned boxes and $Q_k^\circ \cap Q_\ell^\circ = \emptyset$ for $k \neq \ell$.
The basis and dual basis need to satisfy some reasonable properties.

- The basis satisfies
$$\sum_{\mathbf{i}} B_{\mathbf{i}} \leq 1 \qquad \text{everywhere on} \qquad \widehat{\Omega}.$$
- The space spanned by the basis contains all tensor-product polynomials of degree p on $\widehat{\Omega}$, i.e., $\mathbb{P}^p \subseteq \text{span}(\mathcal{B})$.
- The dual functionals have support inside the support of the basis functions, i.e.,
$$\lambda_{\mathbf{j}}(f) = 0 \text{ if } \text{supp}(f) \cap \text{supp}(B_{\mathbf{j}}) = \emptyset.$$
- The dual functionals are stable in L^∞, i.e.,
$$|\lambda_{\mathbf{j}}(f)| \leq C_\infty \|f\|_{L^\infty(\text{supp}(B_{\mathbf{j}}))}.$$

Standard tensor-product B-splines together with the Schumaker dual basis satisfy all properties.
We directly obtain the following.

Lemma 6.1. *The projector* $\Pi : L^2(\widehat{\Omega}) \to \text{span}(\mathcal{B})$ *with*

$$\Pi(f) = \sum_{\mathbf{i}} \lambda_{\mathbf{i}}(f) B_{\mathbf{i}} \tag{6.1}$$

satisfies

$$\|\Pi(f)\|_{L^\infty(Q)} \leq C_\infty \|f\|_{L^\infty(\widetilde{Q})}, \tag{6.2}$$

where Q *is some box and* $\widetilde{Q}$ *is its support extension, i.e.,*

$$\widetilde{Q} = \bigcup_{\mathbf{i} \in I(Q)} \text{supp}(B_{\mathbf{i}})$$

where $I(Q) = \{\mathbf{i} : \text{supp}(B_{\mathbf{i}}) \cap Q \neq \emptyset\}$.

Proof. We have

$$\begin{aligned}\|\Pi(f)\|_{L^\infty(Q)} &= \left\|\sum_{\mathbf{i}} \lambda_{\mathbf{i}}(f) B_{\mathbf{i}}\right\|_{L^\infty(Q)} \\ &\leq \left\|\sum_{\mathbf{i}\in I(Q)} \max_{\mathbf{j}\in I(Q)} |\lambda_{\mathbf{j}}(f)| B_{\mathbf{i}}\right\|_{L^\infty(Q)} \\ &\leq \max_{\mathbf{j}\in I(Q)} (|\lambda_{\mathbf{j}}(f)|) \leq C_\infty \|f\|_{L^\infty(\widetilde{Q})},\end{aligned}$$

which concludes the proof. □

We can now combine standard polynomial approximation results with the local boundedness of the operator Π to obtain the following.

Theorem 6.2. *Let $Q \subset \widehat{\Omega}$ be some element of the mesh. If $\Pi(g)|_{\widetilde{Q}} = g|_{\widetilde{Q}}$ for all $g \in \mathbb{P}^p$, then*

$$|(I-\Pi)u|_{H^r(Q)} \leq K \left(1 + \frac{h_{\widetilde{Q},\max}}{h_{\widetilde{Q},\min}}\right)^d \frac{(h_{\widetilde{Q},\max})^s}{(h_{Q,\min})^r} |u|_{H^s(\widetilde{Q})}, \tag{6.3}$$

for all $u \in H^s(\widetilde{Q})$, with $0 \leq r < s \leq p+1$.

Proof. Given a polynomial $g \in \mathbb{P}^p$, we have

$$\begin{aligned}|(I-\Pi)u|_{H^r(Q)} &= |u - g - \Pi(u-g)|_{H^r(Q)} &(6.4)\\ &\leq |u-g|_{H^r(Q)} + |\Pi(u-g)|_{H^r(Q)}\,. &(6.5)\end{aligned}$$

For suitable g (being an averaged Taylor polynomial), the first part satisfies

$$|u-g|_{H^r(Q)} \leq |u-g|_{H^r(\widetilde{Q})} \leq C_1 \frac{(h_{\widetilde{Q},\max})^s}{(h_{\widetilde{Q},\min})^r} |u|_{H^s(Q)}, \tag{6.6}$$

where C_1 is a constant independent of the mesh size. The second part in (6.5) is a Sobolev semi-norm of a polynomial, so we can apply a standard inverse estimate

$$|\Pi(u-g)|_{H^r(Q)} \leq C_{inv} (h_{Q,\min})^{-r} \|\Pi(u-g)\|_{L^2(Q)}. \tag{6.7}$$

We have

$$\|\Pi(u-g)\|_{L^2(Q)} \leq (h_{Q,\max})^{d/2} \|\Pi(u-g)\|_{L^\infty(Q)} \leq (h_{Q,\max})^{d/2} C_\infty \|u-g\|_{L^\infty(\widetilde{Q})}, \tag{6.8}$$

due to Lemma 6.1. Again, for g being an averaged Taylor polynomial, we obtain

$$\|u-g\|_{L^\infty(\widetilde{Q})} \leq C_2 \left(1 + \frac{h_{\widetilde{Q},\max}}{h_{\widetilde{Q},\min}}\right)^d (h_{\widetilde{Q},\max})^{s-d/2} |u|_{H^s(\widetilde{Q})}.$$

So, in total, we obtain

$$\begin{aligned}|(I-\Pi)u|_{H^r(Q)} &\leq |u-g|_{H^r(Q)} + |\Pi(u-g)|_{H^r(Q)} \\ &\leq C_1 \frac{(h_{\widetilde{Q},\max})^s}{(h_{\widetilde{Q},\min})^r} |u|_{H^s(Q)} + K' \left(1 + \frac{h_{\widetilde{Q},\max}}{h_{\widetilde{Q},\min}}\right)^d \frac{(h_{\widetilde{Q},\max})^s}{(h_{Q,\min})^r} |u|_{H^s(\widetilde{Q})},\end{aligned}$$

where $K' = C_{inv} C_\infty C_2$. □

In this estimate, the support extension should be shape regular, that is, $h_{\widetilde{Q},\max}/h_{\widetilde{Q},\min}$ should be bounded. It is reasonable to assume that all elements are shape regular. Moreover for $r > 0$, the sizes of the element $h_{Q,\min}$ and support extension $h_{\widetilde{Q},\max}$ should be of similar order. Otherwise, the estimate might blow up.

We will now discuss two different constructions to obtain suitable locally refineable bases and dual bases. One can distinguish two different approaches:

- Domain based refinement (THB-splines)
- Function based refinement (T-splines)

6.2 THB-splines

Let $\Omega^* \supset \widehat{\Omega}$ be a box containing our domain of interest. Let moreover

$$\mathcal{S}^0 \subset \mathcal{S}^1 \subset \mathcal{S}^2 \subset \ldots \subset \mathcal{S}^n$$

be a nested sequence of tensor-product B-spline spaces on Ω^*, where $\mathcal{S}^\ell$ has the basis $\mathcal{B}^\ell = \{B_{\mathbf{i}}^\ell\}$.

In addition, we have given a nested sequence of subdomains

$$\widehat{\Omega} = \Omega^0 \supseteq \Omega^1 \supseteq \Omega^2 \supseteq \ldots \supseteq \Omega^n,$$

where each Ω^ℓ is the union of elements of level ℓ.

We can now define the hierarchical basis corresponding to the spaces $\mathcal{S}^\ell$ and domains Ω^ℓ.

Definition 6.3. *The hierarchical bases $\mathcal{H}^\ell$ are defined recursively with*

1. $\mathcal{H}^0 = \{B_{\mathbf{i}}^0 \in \mathcal{B}^0 : \operatorname{supp}(B_{\mathbf{i}}^0) \cap \Omega^0 \neq \emptyset\}$.
2. *For $\ell = 0, \ldots, n-1$:*

$$\mathcal{H}_C^{\ell+1} = \{B_{\mathbf{i}}^k \in \mathcal{H}^\ell : \operatorname{supp}(B_{\mathbf{i}}^k) \not\subseteq \Omega^{\ell+1}\}, \tag{6.9}$$

$$\mathcal{H}_F^{\ell+1} = \{B_{\mathbf{i}}^{\ell+1} \in \mathcal{B}^{\ell+1} : \operatorname{supp}(B_{\mathbf{i}}^{\ell+1}) \subseteq \Omega^{\ell+1}\}, \tag{6.10}$$

$$\mathcal{H}^{\ell+1} = \mathcal{H}_C^{\ell+1} \cup \mathcal{H}_F^{\ell+1}. \tag{6.11}$$

We call the functions in $\mathcal{H}^n$ the hierarchical B-splines.

Obviously, the functions in $\mathcal{H}^n$ are linearly independent, so they form a basis. The space spanned by the basis contains all polynomials. However, they do not form a partition of unity.

To obtain a partition of unity, we need to define the so-called truncation of a basis function. Let $s \in \mathcal{S}^{\ell+1}$ have the form

$$s = \sum_{\mathbf{i}} c_{\mathbf{i}}^{\ell+1} B_{\mathbf{i}}^{\ell+1},$$

then the truncation of s is defined as

$$\operatorname{trunc}^{\ell+1}(s) = \sum_{\mathbf{i} \,:\, \operatorname{supp}(\mathbf{B}_{\mathbf{i}}^{\ell+1}) \not\subseteq \boldsymbol{\Omega}^{\ell+1}} c_{\mathbf{i}}^{\ell+1} B_{\mathbf{i}}^{\ell+1}.$$

Now we can define truncated bases as follows.

Definition 6.4. *The truncated hierarchical bases $\mathcal{T}^\ell$ are defined recursively with*

1. $\mathcal{T}^0 = \mathcal{H}^0$.
2. *For $\ell = 0, \dots, n-1$:*

$$\mathcal{T}_C^{\ell+1} = \{\hat{B}_{\mathbf{i}}^k = \text{trunc}^{\ell+1}(B_{\mathbf{i}}^k) : B_{\mathbf{i}}^k \in \mathcal{T}^\ell, \text{ supp}(B_{\mathbf{i}}^k) \not\subseteq \Omega^{\ell+1}\}, \tag{6.12}$$

$$\mathcal{T}_F^{\ell+1} = \{\hat{B}_{\mathbf{i}}^{\ell+1} = B_{\mathbf{i}}^{\ell+1} \in \mathcal{B}^{\ell+1} : \text{supp}(B_{\mathbf{i}}^{\ell+1}) \subseteq \Omega^{\ell+1}\}, \tag{6.13}$$

$$\mathcal{T}^{\ell+1} = \mathcal{T}_C^{\ell+1} \cup \mathcal{T}_F^{\ell+1}. \tag{6.14}$$

We call the functions in $\mathcal{T}^n$ the truncated hierarchical B-splines (THB-splines).

The THB-splines satisfy the following.

Theorem 6.5. *The THB-splines $\mathcal{T}^n$ form a partition of unity on $\widehat{\Omega}$. Moreover*

$$\text{span}(\mathcal{T}^n) = \text{span}(\mathcal{H}^n).$$

What remains to be shown is the existence of a suitable dual basis. Let therefore Λ^ℓ be a dual basis for $\mathcal{B}^\ell$.

We can now define the hierarchical projector

$$\Pi(f) = \sum_{\ell=0}^{n} \sum_{\mathbf{i} \in I_n^\ell} \lambda_{\mathbf{i}}^\ell(f) \hat{B}_{\mathbf{i}}^\ell,$$

where $\hat{B}_{\mathbf{i}}^\ell \in \mathcal{T}^\ell$ and $\lambda_{\mathbf{i}}^\ell \in \Lambda^\ell$. The operator Π reproduces polynomials.

Theorem 6.6. *If*

$$\sum_{\mathbf{i}} \lambda_{\mathbf{i}}^\ell(g) B_{\mathbf{i}}^\ell = g$$

for all $g \in \mathbb{P}^p$, then

$$\Pi(g) = g$$

for all $g \in \mathbb{P}^p$.

Moreover, we have the following.

Theorem 6.7. *We assume that each $\lambda_{\mathbf{i}}^\ell$, for $\mathbf{i} \in I_n^\ell$, used in the projector satisfies*

$$f|_{\Omega^\ell \setminus \Omega^{\ell+1}} = 0 \Rightarrow \lambda_{\mathbf{i}}^\ell(f) = 0.$$

Then, if

$$\sum_{\mathbf{i}} \lambda_{\mathbf{i}}^\ell(s) B_{\mathbf{i}}^\ell = s$$

for all $s \in \mathcal{S}^\ell$, we have

$$\Pi(s) = s$$

for all $s \in \text{span}(\mathcal{T}^n)$.

The condition on the support is not satisfied for the Schumaker dual basis, but one can show that such a dual basis exists.

We now have all the ingredients to apply the results from Subsection 6.1, as the dual basis can be chosen such that all functionals are suitably bounded and the operator Π reproduces polynomials.

To be able to bound $h_{Q,\min} \geq \underline{c}\, h_{\widetilde{Q},\min}$, we use the notion of mesh level disparity.

Definition 6.8. *The mesh level disparity δ is the largest difference of levels of THB-splines supported on any element Q.*

If the mesh on level 0 is regular of size H_0 and the refinement is diadic, then we have

$$h_{Q,\min} = h_{Q,\max} = h_\ell = H_0 2^{-\ell}$$

and

$$h_{\widetilde{Q},\min} \leq h_{\widetilde{Q},\max} \leq C\, p\, H_0 2^{\delta-\ell} = C\, p\, 2^\delta h_\ell,$$

for any Q being an element of level ℓ. From this we obtain $\underline{c} = (C\, p\, 2^\delta)^{-1}$.

6.3 T-splines

In this section we consider T-splines for local refinement. T-spline basis functions are standard tensor-product B-splines over a non-tensor-product mesh. The definition of T-splines is based on the fact that B-spline basis functions only depend on their local knot vectors.

We assume to have given a planar box mesh over which we define T-splines. Note that T-splines generalize also to arbitrary dimensions.

Definition 6.9. *A box mesh over a two dimensional domain $\widehat{\Omega}$ is a collection of axis-aligned boxes Q_k, such that*

$$\widehat{\Omega} = \bigcup_k Q_k,$$

vertices $\mathbf{x}_\ell$ and horizontal and vertical edge segments $[\mathbf{x}_j, \mathbf{x}_{j'}]$.

6.3.1 Index T-mesh

An index T-mesh is defined to be a box mesh over an integer grid.

Definition 6.10. *Let $(\underline{m}, \overline{m}), (\underline{n}, \overline{n}) \in \mathbb{Z} \times \mathbb{Z}$ be the minimal and maximal indices in the first and second direction, respectively. An index T-mesh $\mathcal{M}$ is a box mesh over $[\underline{m}, \underline{n}] \times [\overline{m}, \overline{n}]$, where all vertices are integer indices $\mathbf{x}_\ell \in \mathbb{Z} \times \mathbb{Z}$. All vertices in the interior have valency 3 or 4.*

For given polynomial degrees in the two dimensions we can define an admissible T-mesh.

Definition 6.11. *Let $p_1, p_2 \in \mathbb{Z}_0^+$ be the degrees in the first and second direction, respectively. An index T-mesh is called admissible, if*

$$\{\ell\} \times [\underline{n}, \overline{n}] \text{ for } \ell = \underline{m}, \ldots, \underline{m} + \left\lfloor \frac{p_1+1}{2} \right\rfloor \text{ and } \ell = \overline{m} - \left\lfloor \frac{p_1+1}{2} \right\rfloor, \ldots, \overline{m},$$

$$[\underline{m}, \overline{m}] \times \{\ell\} \text{ for } \ell = \underline{n}, \ldots, \underline{n} + \left\lfloor \frac{p_2+1}{2} \right\rfloor \text{ and } \ell = \overline{n} - \left\lfloor \frac{p_2+1}{2} \right\rfloor, \ldots, \overline{n}$$

are contained as edges in the mesh and all vertices in

$$]\underline{m}, \underline{n}[\times]\overline{m}, \overline{n}[\setminus \mathrm{AR}$$

have valency four. Here

$$\mathrm{AR} = \left[\underline{m} + \left\lfloor \frac{p_1+1}{2} \right\rfloor, \underline{n} + \left\lfloor \frac{p_2+1}{2} \right\rfloor\right] \times \left[\overline{m} - \left\lfloor \frac{p_1+1}{2} \right\rfloor, \overline{n} - \left\lfloor \frac{p_2+1}{2} \right\rfloor\right]$$

is the active region.

We can now define the degrees of freedom within our index T-mesh. Then

- for odd p_1, p_2, all vertices in AR,
- for odd p_1 and even p_2, all vertical edge segments in AR,
- for even p_1 and odd p_2, all horizontal edge segments in AR,
- for even p_1, p_2, all faces in AR,

are called anchors, denoted by $A \in \mathcal{A}(\mathcal{M})$.

6.3.2 T-mesh and T-splines

We can now define one T-spline for every anchor. But before we need to define a T-mesh.

Definition 6.12. *We assign a knot value for every index of the index T-mesh, with*

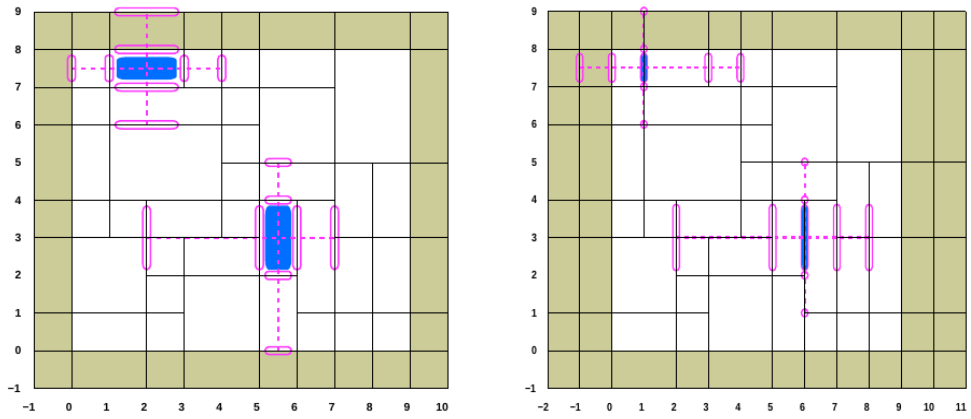
$$0 = t^1_{\underline{m}} = \ldots = t^1_{\underline{m}+p_1} < t^1_{\underline{m}+p_1+1} \leq \ldots \leq t^1_{\overline{m}-p_1-1} < t^1_{\overline{m}-p_1} = \ldots = t^1_{\overline{m}} = 1$$

as well as

$$0 = t^2_{\underline{n}} = \ldots = t^2_{\underline{n}+p_2} < t^2_{\underline{n}+p_2+1} \leq \ldots \leq t^2_{\overline{n}-p_2-1} < t^2_{\overline{n}-p_2} = \ldots = t^2_{\overline{n}} = 1.$$

The mesh introduced by these knots is called the T-mesh of $\mathcal{M}$, in short $\mathcal{T}(\mathcal{M})$. For every anchor $A \in \mathcal{A}(\mathcal{M})$ we can now extract local knot vectors in horizontal and vertical direction, $\mathrm{hkv}(A)$ *and* $\mathrm{vkv}(A)$*, respectively.*

We describe the extraction for odd degree $p = 2k+1$ in horizontal direction: Let the anchor A be the vertex (m', n') in index space. Then the extraction for the horizontal knot vector takes the knot $t^1_{m'}$ as the central knot. Then, moving left, the first $k+1$ knots are collected, where the horizontal line through A intersects vertical knot lines. Similarly, the next $k+1$ knots to the right are collected, see Figure 6.2.



(a) $(p, q) = (2, 2)$ (b) $(p, q) = (3, 2)$

Figure 6.2: Knot vector extraction for different degrees.

Definition 6.13. *The T-splines of degree (p_1, p_2) over the admissible T-mesh $\mathcal{T}(\mathcal{M})$ are given by*

$$\mathcal{B}(\mathcal{M}) = \{B^{p_1}[\mathrm{hkv}(A)](t_1)\, B^{p_2}[\mathrm{vkv}(A)](t_2) \ : \ \text{for } A \in \mathcal{A}(\mathcal{M})\}.$$

Here, $B^p[\Xi](t)$ is the B-spline of degree p having the local knot vector Ξ of length $p+2$.

6.3.3 Dual-compatible T-splines

In the following we define locally dual functionals for the T-splines. It turns out that under certain geometric conditions the locally dual functionals actually form a dual basis.

Before we can define the dual basis, we need some additional notation.

Definition 6.14. *We say that two local knot vectors Ξ and Ξ' of length $p+2$ overlap, if for every knot $t \in \Xi$ with $\min(\Xi') \le t \le \max(\Xi')$ we have $t \in \Xi'$; as well as for every knot $t' \in \Xi'$ with $\min(\Xi) \le t' \le \max(\Xi)$ we have $t' \in \Xi$, see Figure 6.3.*

Figure 6.3: Overlapping (left) and non-overlapping (right) knot vectors.

We say that two anchors A and A' partially overlap, if they overlap either horizontally (with $\mathrm{hkv}(A)$ and $\mathrm{hkv}(A')$ overlapping) or vertically (with $\mathrm{vkv}(A)$ and $\mathrm{vkv}(A')$ overlapping).

Definition 6.15. *Given a local knot vector Ξ of length $p+2$. Let $\lambda^p[\Xi](f)$ be a functional, such that $\lambda^p[\Xi](B^p[\Xi]) = 1$ and $\lambda^p[\Xi](B^p[\Xi']) = 0$ for all knot vectors Ξ' that are overlapping with Ξ.*

Theorem 6.16. *Let $\mathcal{B}(\mathcal{M})$ be T-splines of degree (p_1, p_2). We assume that all pairs of anchors in $\mathcal{A}(\mathcal{M})$ partially overlap. Then, the functionals*

$$\Lambda(\mathcal{M}) = \{\lambda^{p_1}[\mathrm{hkv}(A)](t_1) \otimes \lambda^{p_2}[\mathrm{vkv}(A)](t_2) \ : \ \text{for } A \in \mathcal{A}(\mathcal{M})\}$$

form a dual basis for $\mathcal{B}(\mathcal{M})$.

A proof of this theorem is straight-forward. We call T-splines where all anchors partially overlap dual-compatible T-splines.

Dual-compatible T-splines have a nice geometric interpretation.

Definition 6.17. *A T-node extension is a line segment that extends any T-node (a vertex of valency 3)* $\lfloor \frac{p+1}{2} \rfloor$ *forward and* $\lceil \frac{p-1}{2} \rceil$ *elements back, where* $p = p_1$ *for horizontal and* $p = p_2$ *for vertical T-node extensions. See Figure 6.4.*

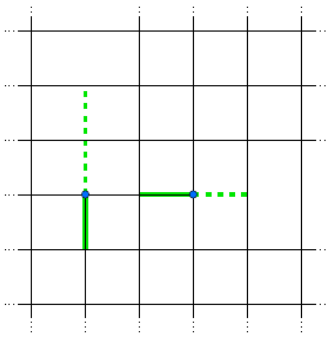


Figure 6.4: T-node extensions of degree 2 (horizontally) and 3 (vertically).

Theorem 6.18. *An index T-mesh* $\mathcal{M}$ *is called analysis-suitable, if no vertical and horizontal T-node extensions intersect. If* $\mathcal{M}$ *is analysis-suitable, then* $\mathcal{B}(\mathcal{M})$ *are dual-compatible.*

6.4 Literature

Section 6.1 follows mainly [21, 22]. Local error bounds for polynomials and splines can be found in [5]. Section 6.2 is based on [23, 21]. T-Splines were first described in [24]. Section 6.3 is based on [25, 26].

Chapter 7

Linear solvers for Isogeometric Analysis

7.1 Introduction

After assembling, we end up with the linear system

$$A_h \underline{u}_h = \underline{f}_h$$

to be solved. The stiffness matrix A_h has the following properties.

- A_h *is large-scale:* The number of degrees of freedom (dofs) is $\mathcal{O}((n+p)^d) \eqsim \mathcal{O}(n^d)$.
- A_h *is sparse:* The number of non-zero entries per row is $\mathcal{O}(p^d)$.
- The *condition number* of A_h is as follows:

$$\kappa(A_h) = \mathcal{O}(\underbrace{h^{-2}}_{\text{like in FEM}} \overbrace{2^p}^{\text{very bad}})$$

  (If we consider the pure Neumann problem and do not take the averaging condition $\int_\Omega u(x)\,\mathrm{d}x = 0$ into account, we certainly have $\lambda_{\min}(A_h) = 0$ and thus also $\kappa(A_h) = \infty$.)

- The *condition number* of the mass matrix M_h is as follows:

$$\kappa(M_h) = \mathcal{O}(2^p).$$

  Here, as for standard FEM, the condition number is independent of h.

- The *condition number* of $K_h^{-1} M_h$: Using $\|u\|_{H^1} \geq \|u\|_{L_2}$ (in combination with coercivity of $a(\cdot,\cdot)$), we obtain

$$A_h \gtrsim M_h$$

  and using a standard inverse estimate (Theorem 4.2), we obtain

$$A_h \lesssim p^4 h^{-2} M_h.$$

Here and in what follows within this chapter, we assume for simplicity that $h = h_{\max} \eqsim h_{\min}$. Thus,

$$\kappa(M_h^{-1} A_h) \leq \mathcal{O}(p^4 h^{-2}),$$

which is much better than an exponential dependence on p. If we restrict ourselves to the space $\widetilde{S}_{p,\Xi}$, we have even

$$\kappa(\widetilde{M}_h^{-1} \widetilde{A}_h) = \mathcal{O}(h^{-2}),$$

which is exactly the same as we know from standard FEM.

We call a solver *optimal* if it can solve the linear system (up to a fixed precision) in

$$\mathcal{O}(n^d) = \mathcal{O}(\#\text{ dofs})$$

flops. We call a solver *quasi optimal* if it can solve the linear system (up to a fixed precision) in

$$\mathcal{O}(p^d n^d) = \mathcal{O}(\#\text{ non-zeros}(A_h))$$

flops. Note that this definitions are *not* accepted by everybody.

In general, we can distinguish between *direct solvers* and *iterative solvers*. In both classes, we can consider methods that are available for all matrices which satisfy some properties (like being symmetric and/or positive definite). We call these solvers *generic* solvers. Besides them, there are solvers which are only applicable to matrices that have a very particular structure (like a Kronecker structure) or which represent the discretization of a PDE.

In the following list, we only consider solvers that are of particular interest for us.

1. *Direct solvers*
   (a) *Generic direct solvers*
      i. Standard algorithms: Gaussian elimination, Cholesky decomposition, LU decomposition, ...
      ii. Non-standard algorithms: matlab backslash \, PARDISO, ...
   (b) *Direct solvers for special cases*
      i. Direct solvers for Kronecker products
2. *Iterative solvers*
   (a) *Generic iterative solvers*
      i. Linear iterations: Richardson iteration, Jacobi iteration, Gauss-Seidel iteration, ...
      ii. Krylov-space methods: conjugate gradient, (generalized) minimum residual method, ...
   (b) *Iterative solvers for special cases*
      i. Single-patch case: multigrid, ...
      ii. Multi-patch case: multigrid, IETI, ...

Standard Generic direct solvers (1.a.ii) usually do not show optimal complexity. Some of those algorithms are efficient for matrices with bounded bandwidth, cf. Section 7.2. We will see that those algorithms are efficient only for $d = 1$.

In the last decades, much effort was put on developing fast direct solvers (1.a.ii). In general, one can say that there are good direct solvers that yield very good results for up to $d = 2$. In IgA, such direct solvers typically suffer from the fact that the number of non-zeros per row, grows with the spline degree. Also the exponential dependence of the condition number on the spline degree might cause numerical instability. For further information, see [1, Table 8.1].

We focus to iterative solvers (2.a). Krylov space methods (2.a.ii) are in general good methods to solve linear systems. Since we only consider elliptic problems, we have symmetric and positive definite matrices. Thus, the conjugate gradient method is our choice. Its convergence rates depend on the spectrum of the matrix of interest. The classical estimate depends on the condition number. Since we know that the condition number grows if we refine the problem, we need to do something about that: *preconditioning*, see Section 7.3.

Direct solvers for Kronecker products (1.b.i) are very fast. However, typically, the mass matrix and the stiffness matrix are not Kronecker products. In Chapter 8, we discuss how to apply them to realize a preconditioner.

In the finite element context, multigrid solvers (2.b) are known to be very efficient. We can extend these solvers to the context of Isogeometric Analysis. Since multigrid solvers are linear iterations, we can again use them as a preconditioner.

The IsogEometric Tearing and Interconnecting (IETI) method is a variant of the Finite Element Tearing and Interconnection (FETI) method that realizes the coupling of patch-local solvers. This means, by choosing IETI as solver, we still have the freedom to choose a patch-local solver.

7.2 Cholesky factorization

One possible direct solver is the Cholesky factorization, cf. [27]: Each *symmetric and positive definite* (or Hermitian positive-definite) matrix A can be represented as

$$A = L\, L^*,$$

where L is a lower triangular matrix with positive diagonal entries and L^* denotes the (conjugate) transpose of L. The algorithm is shown in Fig. 7.1. One can easily observe that the algorithm only reads every entry of the matrix A once: The entry $A_{i,j}$ is read before one writes the entry $L_{i,j}$. Thus, the algorithm can be applied in-place, i.e., in lines 08 and 10, one could write to $A_{i,j}$ and the final result would be the lower-triangular part of A.

After computing the Cholesky factorization, a linear system

$$A^{-1}\underline{f} = L^{-*}L^{-1}\underline{f}$$

can be realized by back-substitution.

In general, the computational complexity grows like n^3, where n is the number of unknowns. A better result can be obtained if the bandwidth of the matrix of interest is appropriately

```
01  function cholesky(A)
02      for i = 1, ..., n
03          for j = 1, ..., i
04              ξ ← A_{i,j}
05              for k = 1, ..., j − 1
06                  ξ ← ξ − L_{i,k} L_{j,k}
07              if i > j
08                  L_{i,j} ← ξ/L_{j,j}
09              else    // here i = j
10                  L_{i,j} ← √ξ
11      return L   // undefined values are 0
```

Figure 7.1: Cholesky decomposition

bounded. The *bandwidth* of a matrix A is the smallest non-negative integer q such that

$$A_{i,j} = 0 \quad \text{for all indices } i \text{ and } j \text{ with } |i-j| > q.$$

Note that in this case, we do not have a fill-in. Let q be the bandwidth. Consider that we have chosen $j < i - q$ in line 03. Thus, we have $A_{i,j} = 0$ in line 04. All choices k in line 05 satisfy $k < j < i - q$. Thus, $A_{i,k} = 0$ in line 06. Thus, $\xi = 0$ and nothing is changed by line 08. Concluding, we can replace line 03 by "`for` $j = \max\{1, i-q\}, \ldots, i$" and line 05 by "`for` $k = \max\{1, i-q\}, \ldots, j-1$".

Thus, the computational costs of a Cholesky solver are as follows:

- for computing the factorization: $\mathcal{O}(nq^2)$ flops,
- for the back-substitution: $\mathcal{O}(nq)$ flops,

where n is the number of unknowns and q is the bandwidth, cf. [27, p. 180].

In one dimension, the mass matrix M_h and the stiffness matrix A_h are band matrices with bandwidth p. Therefore, the application of a Cholesky solver is feasible. We will make use of that.

This does not carry over to two and more dimensions. Here, the bandwidth of a standard tensor-product discretization is $q = n^{d-1} + p$. Provided $p \lesssim n$, we have

- for computing the factorization: $\mathcal{O}(n^{3d-2})$ flops,
- for the back-substitution: $\mathcal{O}(n^{2d-1})$ flops,

where the number of unknowns is n^d, i.e., n per direction.

7.3 (Preconditioned) conjugate gradient method

The conjugate gradient method allows to solve linear systems with symmetric and positive definite matrix iteratively. For a given matrix A_h, right-hand-side $\underline{f}_h$, initial guess $\underline{x}_h$ and error tolerance ϵ, the conjugate gradient method is as given in Fig. 7.2, if we choose $P_h^{-1} := I$.

```
01  function pcg(A_h, P_h^{-1}, f_h, x_h, ε)
02      r_h ← f_h − A_h x_h
03      if ‖r_h‖ < ε‖f_h‖
04          return x_h
05      p_h ← P_h^{-1} r_h
06      γ ← r_h · p_h
07      do
08          z_h ← A_h p_h
09          α ← γ/(p_h · z_h)
10          x_h ← x_h + α p_h
11          r_h ← r_h − α z_h
12          if ‖r_h‖ < ε‖f_h‖
13              return x_h
14          z_h ← P_h^{-1} r_h
15          β ← 1/γ
16          γ ← r_h · z_h
17          β ← β γ
18          p_h ← z_h + β p_h
```

Figure 7.2: Preconditioned conjugate gradient method

Note that for the realization of the algorithm, we do not need to know the matrix A_h itself, we only need to be able to evaluate $A_h \, \underline{q}_h$ for any given vector $\underline{q}_h$.

The convergence rate of the *conjugate gradient* iteration is bounded by

$$\frac{\sqrt{\kappa(A_h)} - 1}{\sqrt{\kappa(A_h)} + 1}.$$

Note that this is only one possible bound; better results can be obtained if the whole spectrum of A_h is taken into account.

Since $\kappa(A_h)$ can be very large, we need a preconditioner. Let P_h be a SPD matrix. Consider the problem

$$P_h^{-1} A_h \underline{x}_h = P_h^{-1} \underline{f}_h.$$

Note that $P_h^{-1} A_h$ is self-adjoined in the scalar product $(\cdot, \cdot)_{P_h}$. Thus, we can apply a variant of the conjugate gradient method, the *preconditioned conjugate gradient* (PCG) method. Here, the pseudo code is given in Fig. 7.2. The standard convergence analysis shows that the PCG method converges with a rate that is bounded by

$$\frac{\sqrt{\kappa(P_h^{-1} A_h)} - 1}{\sqrt{\kappa(P_h^{-1} A_h)} + 1}.$$

So, we are interested in choosing a preconditioner P_h. As for the matrix A_h, we do not need the matrix P_h itself. We only need an algorithm that computes $P_h^{-1} \underline{q}_h$ for a given vector $\underline{q}_h$. Among others, we are interested in the following approaches.

- One possibility for the single-patch case is to choose the stiffness matrix for the pa-

rameter domain P_h as preconditioner, where we define

$$P_h := ((\varphi_i \circ F^{-1}, \varphi_j \circ F^{-1})_{H^1(\widehat{\Omega})})_{i,j=1,\dots,n},$$

where the functions φ_i are the basis functions in the space V_h (on the physical domain). Using Theorem 2.16, we obtain

$$\kappa(P_h^{-1} A_h) \leq \|\nabla F\|_{L_\infty(\widehat{\Omega})}^{1+d/2} \|(\nabla F)^{-1}\|_{L_\infty(\widehat{\Omega})}^{1+d/2},$$

which is obviously independent of h and p. Since this heavily depends on the geometry function, we can apply this approach only if the geometry function is not too distorted.

In the next section, we discuss how to solve linear systems of the form

$$P_h \underline{w}_h = \underline{g}_h$$

using a direct solver cheaply.

- An alternative approach for preconditioning is to take one or a few steps of a linear iterative solver. (Note that the conjugate gradient method is not linear.) Linear iterative solvers are, e.g., the Gauss-Seidel method, the Jacobi method and (typically) also multigrid solvers.

  So, consider one step of a multigrid iteration. Let $\underline{x}_h^{(0)}$ be the initial guess and $\underline{x}_h^{(1)}$ be the first iterate. As for any other linear iteration scheme, there is a *matrix* MG_h such that

$$\underbrace{\underline{x}_h^{(1)} - A_h^{-1} \underline{f}_h}_{\text{error after 1 iteration}} = \underbrace{(I - \mathrm{MG}_h\, A_h)}_{\text{iteration matrix}} \underbrace{(\underline{x}_h^{(0)} - A_h^{-1} \underline{f}_h)}_{\text{initial error}}.$$

  We will see that the multigrid method converges. Thus

$$\rho(I - \mathrm{MG}_h\, A_h) \leq q_0 < 1. \tag{7.1}$$

  Now consider a preconditioner P_h^{-1}, which realizes $P_h^{-1} \underline{q}_h$ by one step of the multigrid method with initial guess $\underline{x}_h^{(0)} := 0$ and $\underline{f}_h := \underline{q}_h$:

$$\underline{x}_h^{(1)} = \mathrm{MG}_h \underline{f}_h.$$

  From (7.1), we obtain

$$1 - q_0 \leq \mathrm{MG}_h\, A_h \leq 1 + q_0$$

  and thus

$$\kappa(\mathrm{MG}_h\, A_h) \leq \frac{1 + q_0}{1 - q_0}.$$

  This yields a convergence result for the PCG method:

$$q \leq \frac{\sqrt{1 + q_0} - \sqrt{1 - q_0}}{\sqrt{1 + q_0} + \sqrt{1 - q_0}} < q_0$$

  for $q_0 \in (0, 1)$. The statement $q < q_0$ means that the preconditioned conjugate gradient method where one multigrid cycle is applied as preconditioner is always faster than the multigrid method itself. (Note that this is only true if the multigrid method is set up such that it is symmetric.)

Chapter 8

Low-tensor-rank solvers

8.1 Introduction

Tensor methods can be used to efficiently solve linear systems

$$A_h \underline{u}_h = \underline{f}_h$$

provided that A_h is known to have a small tensor-rank r. For two dimensions, this would read as follows:

$$A_h = \sum_{i=1}^{r} A_{i,1} \otimes A_{i,2}$$

for suitable matrices $A_{i,j}$. For more than two dimensions, there is not a unified definition of a *rank*. Certainly, one possibility (for $d = 3$) is that A_h can be expressed as

$$A_h = \sum_{i=1}^{r} A_{i,1} \otimes A_{i,2} \otimes A_{i,3}$$

for suitable matrices $A_{i,j}$. Note that for $r \geq 2$ and $d \geq 3$, the space of matrices with tensor-rank r is not closed.

We will make use of low-tensor-rank constructions. First, in Section 8.2, we make use of the fact that for $\Omega = (0,1)^d$, the mass matrix and the stiffness matrix are particularly nice. In Section 8.3, we comment on low-tensor-rank solvers for the general case.

8.2 Parameter domain preconditioners

Consider the case of Section 5.6, i.e., we have a tensor-product discretization and a mass matrix $\widehat{M}_h$ and a stiffness matrix $\widehat{A}_h$ that represent the L_2 or the H^1 scalar product on the parameter domain, respectively.

In this case, these matrices have a Kronecker-product form. In the two-dimensional case, we have

$$\widehat{M}_h = M_h^{(1)} \otimes M_h^{(2)} \qquad \text{and} \qquad \widehat{A}_h = A_h^{(1)} \otimes M_h^{(2)} + M_h^{(1)} \otimes A_h^{(2)},$$

where $M_h^{(\delta)}$ are univariate mass matrices and $A_h^{(\delta)}$ are univariate stiffness matrices.

The univariate matrices are simple band matrices. For those matrices, we can solve linear systems

$$M_h^{(\delta)} \underline{w}_h = \underline{g}_h \qquad \text{and} \qquad A_h^{(\delta)} \underline{w}_h = \underline{g}_h$$

easily, e.g., using the Cholesky factorization (Section 7.2).

Since $\widehat{M}_h$ is a Kronecker-product of two matrices, we can easily set up a direct solver as follows. Note that $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$. Thus, we have

$$\widehat{M}_h^{-1} = \left(M_h^{(1)}\right)^{-1} \otimes \left(M_h^{(2)}\right)^{-1} = \left(I \otimes \left(M_h^{(2)}\right)^{-1}\right) \left(\left(M_h^{(1)}\right)^{-1} \otimes I\right).$$

Note that we also have

$$\left(\left(M_h^{(1)}\right)^{-1} \otimes I\right) = \begin{pmatrix} \left(M_h^{(1)}\right)^{-1} & & & & \\ & \left(M_h^{(1)}\right)^{-1} & & & \\ & & \left(M_h^{(1)}\right)^{-1} & & \\ & & & \ddots & \\ & & & & \left(M_h^{(1)}\right)^{-1} \end{pmatrix}.$$

This means that

$$\left(\left(M_h^{(1)}\right)^{-1} \otimes I\right) \underline{g}_h$$

can be computed by applying $\left(M_h^{(1)}\right)^{-1}$ to the corresponding blocks of $\underline{g}_h$. The term "applying $\left(M_h^{(1)}\right)^{-1}$ to some vector $\underline{f}_h$" has to be understood as "solving the linear system

$$M_h^{(1)} \underline{v}_h = \underline{f}_h$$

with some appropriate direct solver". Thus, we can realize

$$\left(M_h^{(1)}\right)^{-1} \otimes I$$

by solving n linear systems. In the same way, we can realize

$$I \otimes \left(M_h^{(2)}\right)^{-1}.$$

So, we can apply $\widehat{M}_h^{-1}$ to any vector by solving n linear systems with matrix $M_h^{(1)}$ and n linear systems with matrix $M_h^{(2)}$. The costs for realizing $\widehat{M}_h^{-1}$ that way using the Cholesky factorization are $\mathcal{O}(pn^d + p^2 n)$ flops, cf. Section 7.2. Here we make use of the fact that the factorization itself is only computed once. Provided $p \leq n$, we obtain that the costs are bounded by

$$\mathcal{O}(pn^d)$$

flops. This is quasi-optimal since the number of non-zero entries of $\widehat{M}_h$ is $\mathcal{O}(p^d n^d)$.

We *cannot* extend the same trick to the stiffness matrix

$$\widehat{A}_h = A_h^{(1)} \otimes M_h^{(2)} + M_h^{(1)} \otimes A_h^{(2)}$$

since that matrix is a sum of Kronecker products.

To construct a direct solver for $\widehat{A}_h$, we can use the *fast diagonalization approach*. Here, we first solve the generalized eigenvalue problems

$$A_h^{(\delta)} \underline{v}_h = \lambda M_h^{(\delta)} \underline{v}_h,$$

which yield each n eigenvalues and corresponding eigenvectors. We set up the vectors to be orthonormal in the scalar product $(\cdot,\cdot)_{M_h}$ and collect the eigenvectors in a matrix $Q^{(\delta)}$. Then, we obtain

$$\left(Q^{(\delta)}\right)^T M^{(\delta)} Q^{(\delta)} = I$$

and

$$\left(Q^{(\delta)}\right)^T A^{(\delta)} Q^{(\delta)} = D^{(\delta)},$$

where D is a diagonal matrix containing all the eigenvalues.

We immediately obtain also

$$A^{(\delta)} = \left(Q^{(\delta)}\right)^{-T} D^{(\delta)} \left(Q^{(\delta)}\right)^{-1} \qquad \text{and} \qquad M^{(\delta)} = \left(Q^{(\delta)}\right)^{-T} I \left(Q^{(\delta)}\right)^{-1}.$$

Using the Kronecker-product structure, we obtain

$$\begin{aligned}\widehat{A}_h &= A_h^{(1)} \otimes M_h^{(2)} + M_h^{(1)} \otimes A_h^{(2)} \\ &= \left(\left(Q^{(1)}\right)^{-T} \otimes \left(Q^{(2)}\right)^{-T}\right) \underbrace{\left(D^{(1)} \otimes I + I \otimes D^{(2)}\right)}_{n^2 \times n^2 \text{ diagonal matrix}} \left(\left(Q^{(1)}\right)^{-1} \otimes \left(Q^{(2)}\right)^{-1}\right)\end{aligned}$$

and

$$\widehat{A}_h^{-1} = \left(Q^{(1)} \otimes Q^{(2)}\right) \underbrace{\left(D^{(1)} \otimes I + I \otimes D^{(2)}\right)^{-1}}_{n^2 \times n^2 \text{ diagonal matrix}} \left(\left(Q^{(1)}\right)^T \otimes \left(Q^{(2)}\right)^T\right).$$

Using the latter, we can $\widehat{A}_h^{-1} \underline{w}_h$ for any given vector $\underline{w}_h$ by applying the following steps.

- Compute the matrices $Q^{(\delta)}$, which can be done with $\mathcal{O}(n^3)$ flops each.
- Pre-multiply $\underline{w}_h$ with $\left(Q^{(1)}\right)^T \otimes I$. Since $Q^{(1)}$ is a dense $n \times n$ matrix, this requires n^3 flops.
- Pre-multiply the result with $I \otimes \left(Q^{(2)}\right)^T$, which again requires n^3 flops.
- Pre-multiply the result with the $n^2 \times n^2$ *diagonal* matrix $\left(D^{(1)} \otimes I + I \otimes D^{(2)}\right)^{-1}$, which requires n^2 flops.
- Pre-multiply the result with $I \otimes Q^{(2)}$, which again requires n^3 flops.
- Pre-multiply $\underline{w}_h$ with $Q^{(1)} \otimes I$, which again requires n^3 flops.

The overall costs are $\mathcal{O}(n^3)$ flops. The extension to more dimensions is straight-forward. For any $d \geq 2$, we obtain overall costs of

$$\mathcal{O}(n^{d+1}).$$

Obviously, this is not optimal costs but it might be smaller than $\mathcal{O}(p^d n^d)$, which is the number of non-zero entries of A_h (or of $\widehat{A}_h$). The numerical experiments have shown that this approach is very fast in practice particularly because this method only uses standard algorithms (which are well developed in numerical linear algebra toolboxes). Another reason is that the costly parts are the multiplications with the matrices $Q^{(\delta)}$; here the costs are

$\mathcal{O}(n^{d+1})$, however the corresponding matrices have only dimension $n \times n$ any may fit well into the cash, therefore.

Since

$$A_h \eqsim \widehat{A}_h \qquad \text{and} \qquad M_h \eqsim \widehat{M}_h$$

with constants robust in h and p, we can apply the fast diagonalization approach to obtain good preconditioner also for the problem of interest (which of course lives on the physical domain).

8.3 Preconditioners for the physical domain

Low-tensor rank techniques also allow the construction of preconditioners that live directly for the physical domain. Here, usually also the solution $\underline{u}_h$ is expressed as a tensor-product. Having both the stiffness matrix A_h and the solution vector $\underline{u}_h$ is expressed as a tensor-product, the total number of degrees of freedom might be much smaller than n^d, the usual number of degrees of freedom.

In the paper [28], several approaches are discussed, which can be applied:

- the Alternating Least Squares (ALS) algorithm,
- the Greedy Rank One Update (GROU) algorithm, and
- the Greedy Tucker Approximation (GTA) algorithm.

The numerical experiments show that these approaches are very efficient.

8.4 Literature

The fast diagonalization approach follows [29]. For more information on tensor methods in the IgA context, see [28].

Chapter 9

Multigrid for Isogeometric Analysis

9.1 What is multigrid?

Consider the easiest case first. Consider the Poisson problem with Dirichlet boundary conditions for $d = 1$ and $p = 1$. Then we have

$$A_h = \frac{1}{h} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

Consider the *Jacobi iteration*:

$$\underline{u}_h^{(i+1)} := \underline{u}_h^{(i)} + \tau(\text{diag } A_h)^{-1}(\underline{f}_h - A_h \underline{u}_h^{(i)}). \tag{9.1}$$

Using the exact solution $\underline{u}_h^* := A_h^{-1}\underline{f}_h$ and the error $\underline{e}_h^{(i)} := \underline{u}_h^{(i)} - \underline{u}_h^*$, we obtain

$$\underline{e}_h^{(i+1)} = \underbrace{(I - \tau(\text{diag } A_h)^{-1}A_h)}_{\text{iteration matrix}} \underline{e}_h^{(i)}.$$

We have convergence if

$$\|I - \tau(\text{diag } A_h)^{-1}A_h\| < 1$$

for some norm $\|\cdot\|$. Since A_h is symmetric and positive definite, the iteration (9.1) converges for

$$0 < \tau < \frac{2}{\|(\text{diag } A_h)^{-1}A_h\|}.$$

However, the convergence is *very* slow.

Consider a (rough) initial error, as seen on Figure 9.1.

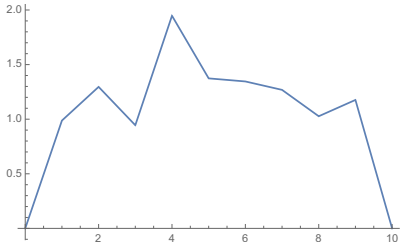


Figure 9.1: Rough initial error

We apply the damped Jacobi iteration to this solve the corresponding problem.

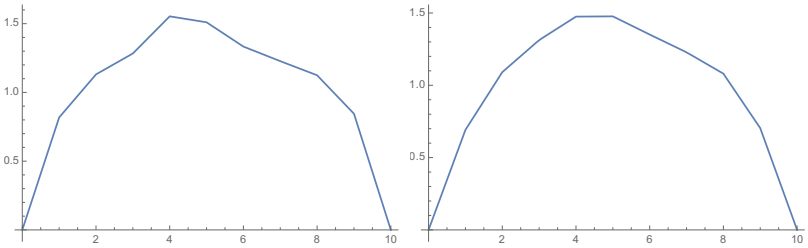


Figure 9.2: Error after one and after two steps of damped Jacobi iteration

We observe on Figure 9.2 that the Jacobi iteration does not really decrease the error. However, it makes the error much smoother. Even if the concept of smoothness might be clear when "looking at the pictures", we need a mathematical definition to understand it.

The smoothed errors can be much better approximated on coarse grids than the original (coarse) problem, cf. Figure 9.3 (left). The obtain this approximation, we need to solve the problem on the coarser grid level as a subproblem. This is feasible because the problem on the coarser grid level has by a factor of 2^d fewer degrees of freedom. If we think of solving the problem on the next coarser grid level exactly, we obtain a *two-grid method*. In practice, we use our algorithm recursively also to solve the problem for the coarser grid level. (Only on a very coarse level, we need to solve the problem with a direct solver.) This yields the *multigrid method*. So, it is easier to solve. When subtracting the approximation on the coarse grid from the original problem, we obtain a very small, but again rough error, see Figure 9.3 (right).

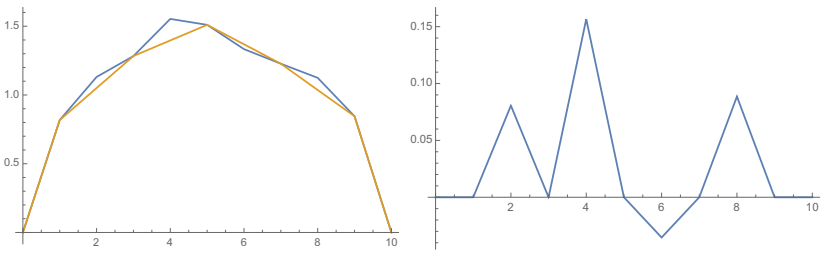


Figure 9.3: Error after one step of damped Jacobi iteration with coarse-grid approximation (left); Error after coarse grid correction (right)

So, what is smoothness?

- It can be characterized by considering different Sobolev norms

$$\|\cdot\|_{L_2}, \quad \|\cdot\|_{H^1}, \quad \|\cdot\|_{H^2}, \quad \cdots$$

  The intuition is that for a smooth function, their value is about the same. For a rough function, the H^1-norm is much larger than the L_2-norm and again the the H^2-norm is much larger than the H^1-norm, etc.

This is not a definition, but it should give some intuition. The definition comes in the next section.

9.2 An abstract multigrid framework

Assume to have a sequence of grids and, therefore, a sequence of nested function spaces

$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_{\mathcal{L}-1} \subset V_{\mathcal{L}} \subset V,$$

where V is the function space of the continuous formulation, like $V = H_0^1(\Omega)$. The function space $V_{\mathcal{L}}$ is the (Isogeometric) function space where the solution of interest lives in. The spaces $V_0, \ldots, V_{\mathcal{L}-1}$ are subspaces.

On each of the spaces V_ℓ, we assume to have a basis $\Phi_\ell := (\phi_{\ell,i})_{i=1,\ldots,N_\ell}$ such that

$$u_\ell = \sum_{i=1}^{N_\ell} u_{\ell,i}\phi_{\ell,i} \in V_\ell \qquad \leftrightarrow \qquad \underline{u}_\ell = (u_{\ell,1}, \ldots, u_{\ell,N_\ell})^T \in \mathbb{R}^{N_\ell}$$

are associated.

Since we have a sequence of nested spaces, each function $u_\ell \in V_\ell$ satisfies $u_\ell \in V_{\ell+1}$. The identity operator

$$\begin{aligned} I \ &: V_\ell \to V_{\ell+1} \\ &v_\ell \to v_{\ell+1} := v_\ell \end{aligned}$$

can be represented using the *prolongation matrix* $P_\ell = [P_{\ell,i,j}]_{i=1,\ldots,N_\ell}^{j=1,\ldots,N_{\ell+1}} \in \mathbb{R}^{N_{\ell+1}\times N_\ell}$ as follows:

$$P_\ell \underline{u}_\ell = \underline{u}_{\ell+1} \qquad \Leftrightarrow \qquad u_\ell = u_{\ell+1}.$$

In other words, we have

$$\phi_{\ell,i} = \sum_{j=1}^{N_{\ell+1}} P_{\ell,i,j}\phi_{\ell+1,j}. \tag{9.2}$$

For the prolongation matrix, see also Sec. 1.2.3.1.

On each grid level, we have a stiffness matrix

$$A_\ell = [A_{\ell,i,j}]_{i,j=1,\ldots,N_\ell} = [a(\phi_{\ell,i}, \phi_{\ell,j})]_{i,j=1,\ldots,N_\ell}.$$

Using (9.2), we obtain

$$A_{\ell,i,j} = a\left(\sum_{\hat{i}=1}^{N_{\ell+1}} P_{\ell,i,\hat{i}}\phi_{\ell+1,\hat{i}}, \sum_{\hat{j}=1}^{N_{\ell+1}} P_{\ell,i,\hat{j}}\phi_{\ell+1,\hat{j}}\right) = \sum_{\hat{i}=1}^{N_{\ell+1}}\sum_{\hat{j}=1}^{N_{\ell+1}} P_{\ell,i,\hat{i}}P_{\ell,i,\hat{j}} \underbrace{a(\phi_{\ell+1,\hat{i}}, \phi_{\ell+1,\hat{j}})}_{= A_{\ell+1,\hat{i},\hat{j}}}$$

This shows

$$A_\ell = P_\ell^T A_{\ell+1} P_\ell,$$

i.e., that the coarse-grid matrices are the *Galerkin projections* of the fine-grid matrix.

Now, we can introduce an *abstract* formulation of the multigrid solver. The multigrid solver on grid level ℓ consists of two steps:

- *Given:* Last iterate (or initial guess) $\underline{u}_\ell^{(i)}$, right-hand side $\underline{f}_\ell$, stiffness matrix A_ℓ, smoother B_ℓ.
- (A) *(Pre-)Smoothing.* Apply $\nu > 0$ smoothing steps:

  $$\underline{u}_\ell^{(i,m+1)} := \underline{u}_\ell^{(i,m)} + B_\ell^{-1}(\underline{f}_\ell - A_\ell \underline{u}_\ell^{(i,m)}),$$

  for $m = 0, \ldots, \nu - 1$, where $\underline{u}_\ell^{(i,0)} := \underline{u}_\ell^{(i)}$.
- (B) *Coarse-grid correction.* Realize the following steps:

  1. Compute defect

     $$\underline{r}_\ell^{(i)} = \underline{f}_\ell - A_\ell \underline{u}_\ell^{(i,\nu)}.$$

  2. Restrict defect to coarser grid level

     $$\underline{r}_{\ell-1}^{(i)} = P_\ell^T \underline{r}_\ell^{(i)}.$$

  3. Solve the problem on the next coarser grid level:
     - If $\ell = 0$ or if we choose to apply the *two-grid method*, we set

       $$\underline{p}_{\ell-1} := A_{\ell-1}^{-1} \underline{r}_{\ell-1}^{(i)}.$$

     - If $\ell > 0$ and we choose to apply the *V-cycle multigrid method* ($\mu = 1$) or *the W-cycle multigrid method* ($\mu = 2$), we invoke the multigrid algorithm with initial guess $\underline{u}_{\ell-1}^{(0)} := 0$ and right-hand-side $\underline{f}_{\ell-1} := r_{\ell-1}^{(i)}$. We compute μ steps of the algorithm and set

       $$\underline{p}_{\ell-1} := \underline{u}_{\ell-1}^{(\mu)},$$

       i.e., to the result of the algorithm after applying μ steps.

  4. Prolongate solution and update:

     $$\underline{u}_\ell^{(i+1)} := \underline{u}_\ell^{(i)} + P_\ell \underline{p}_{\ell-1}.$$

- *Output:* next iterate $\underline{u}_\ell^{(i+1)}$.

As a third step (C), we can do *Post-Smoothing*. Here, we apply the smoother again ν times. The advantage of pre-smoothing is that we obtain a symmetric method. This is of importance if we want to use multigrid as a preconditioner in a PCG iteration.

The two-grid method is of theoretical interest. Here, we have

$$\underline{u}_\ell^{(i+1)} := \underline{u}_\ell^{(i,\nu)} + P_\ell A_{\ell-1}^{-1} P_\ell^T (\underline{f}_\ell - A_\ell \underline{u}_\ell^{(i,\nu)}),$$

For $\mu = 1$, we obtain the most efficient V-cycle approach, cf. Figure 9.4.

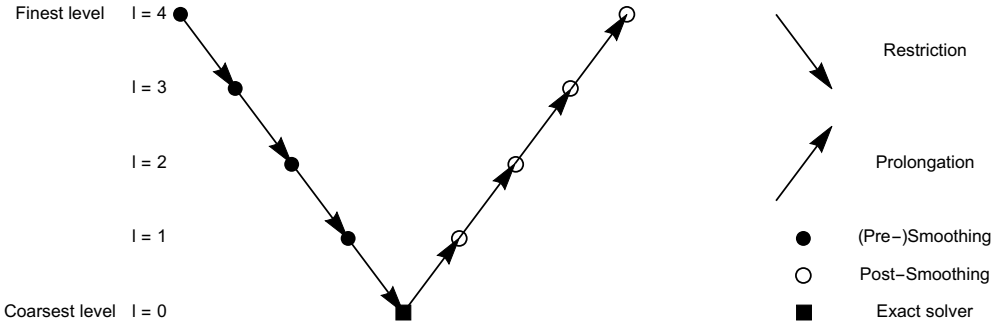


Figure 9.4: V-cycle multigrid

For $\mu = 2$, we obtain the less efficient, but easier to analyze W-cycle approach, cf. Figure 9.5.

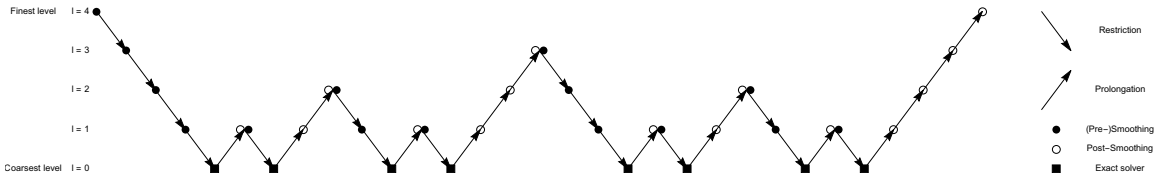


Figure 9.5: W-cycle multigrid

Before we proceed, we discuss the computational complexity of the multigrid solver. Here, we assume that the smoother is a simple iteration scheme, whose costs are *linear* in the number of unknowns, i.e., the computational complexity on each level ℓ is $\mathcal{O}(N_\ell)$.

Observe that for any fixed grid level ℓ, also the computational complexity of

- computing the residual (B.1)
- restricting the residual to the next coarser grid level (B.2)
- prolongating the solution and update (B.4)

is $\mathcal{O}(N_\ell)$. Since we have $N_0 = \mathcal{O}(1)$, the costs of the coarse-grid solver are $\mathcal{O}(1)$. In practice, the costs are negligible.

Since $N_{\ell-1} \approx 2^{-d} N_\ell$, we obtain that the costs of *one* V-cycle are

$$\mathcal{O}(N_\mathcal{L} + N_{\mathcal{L}-1} + \cdots + N_0) = \mathcal{O}(N_\mathcal{L} + 2^{-d} N_\mathcal{L} + 2^{-2d} N_\mathcal{L} + \cdots + 2^{-(\mathcal{L}-1)d} N_L) = \mathcal{O}(N_\mathcal{L}),$$

i.e., we have optimal complexity. For one W-cycle, the costs are

$$\mathcal{O}(N_\mathcal{L} + 2N_{\mathcal{L}-1} + \cdots + 2^{\mathcal{L}-1} N_0) = \mathcal{O}(N_\mathcal{L} + 2^{-(d-1)} N_\mathcal{L} + 2^{-2(d-1)} N_\mathcal{L} + \cdots + 2^{-(\mathcal{L}-1)(d-1)} N_0),$$

which again yields

$$\mathcal{O}(N_\mathcal{L})$$

for $d \geq 2$, but $\mathcal{O}(\mathcal{L} N_\mathcal{L}) = \mathcal{O}(N_\mathcal{L} \log N_\mathcal{L})$ for the (unimportant) case $d = 1$.

Remark 9.1. *This analysis is the standard analysis and does not take the spline degree into account.*

9.3 Hackbusch like convergence analysis

There are several techniques to derive a convergence analysis for multigrid solvers. We use Hackbusch's analysis, cf. [30], since it is one of the simplest methods, which is still quite general.

We first discuss the two-grid method. The overall iteration matrix for the two-grid method is

$$(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell)(I - B_\ell^{-1} A_\ell)^\nu.$$

Thus, its norm is for any symmetric and positive definite matrix L_ℓ given by

$$\begin{aligned} q &= \|(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell)(I - B_\ell^{-1} A_\ell)^\nu\|_{L_\ell} \\ &= \|L_\ell^{1/2}(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell) A_\ell L_\ell^{-1/2} L_\ell^{1/2} A_\ell - 1 (I - B_\ell^{-1} A_\ell)^\nu L_\ell^{-1/2}\|_{\ell^2} \\ &\leq \underbrace{\|L_\ell^{1/2}(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell) A_\ell L_\ell^{-1/2}\|_{\ell^2}}_{C_A :=} \underbrace{\|L_\ell^{1/2} A_\ell - 1 (I - B_\ell^{-1} A_\ell)^\nu L_\ell^{-1/2}\|_{\ell^2}}_{\Psi_S(\nu) :=}. \end{aligned}$$

We are interested in a proof that there is some constant $q_0 < 1$, independent of h, such that

$$q \leq q_0 < 1.$$

We call

- the uniform boundedness of C_A *approximation property* and
- the uniform convergence $\Psi_S(\nu) \to 0$ for $\nu \to \infty$ *smoothing property.*

Observe that C_A is the smallest constant such that

$$\underbrace{\|(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell)\underline{u}_\ell\|_{L_\ell}}_{\text{error after coarse-grid correction}} \leq C_A \underbrace{\|\underline{u}_\ell\|_{A_\ell L_\ell^{-1} A_\ell}}_{\text{error before coarse-grid correction}}$$

holds for all $\underline{u}_\ell$. Note that $P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell$ is the A_ℓ-orthogonal projector into $V_{\ell-1}$, so the approximation property is something like a discretization error estimate.

Observe that $\Psi_S(\nu)$ is the smallest value such that

$$\underbrace{\| (I - B_\ell^{-1} A_\ell)^\nu \underline{u}_\ell \|_{A_\ell L_\ell^{-1} A_\ell}}_{\text{error after smoothing}} \quad \leq \quad \Psi_S(\nu) \quad \underbrace{\| \underline{u}_\ell \|_{L_\ell}}_{\text{error before smoothing}}$$

holds for all $\underline{u}_\ell$.

We can show the smoothing property using the following theorem provided that the smoother can be represented as a symmetric and positive definite matrix.

Theorem 9.2. *Let A_ℓ, B_ℓ and L_ℓ be symmetric and positive definite matrices with*

$$A_\ell \leq B_\ell \leq C_S L_\ell,$$

where $C_S > 0$ is constant. Then, we have

$$\| L_\ell^{-1/2} A_\ell (I - B_\ell^{-1} A_\ell)^\nu L_\ell^{-1/2} \|_{\ell^2} \leq \frac{C_S}{\nu + 1}$$

for all $\nu \in \mathbb{N}$.

Proof. First, we observe

$$\begin{aligned}
\Psi_S(\nu) &:= \| L_\ell^{-1/2} A_\ell (I - B_\ell^{-1} A_\ell)^\nu L_\ell^{-1/2} \|_{\ell^2} \leq \| B_\ell^{-1/2} L_\ell^{1/2} \|_{\ell^2}^2 \| B_\ell^{-1/2} A_\ell (I - B_\ell^{-1} A_\ell)^\nu B_\ell^{-1/2} \|_{\ell^2} \\
&\leq C_S \| B_\ell^{-1/2} A_\ell (I - B_\ell^{-1} A_\ell)^\nu B_\ell^{-1/2} \|_{\ell^2} = C_S \| B_\ell^{-1/2} A_\ell B_\ell^{-1/2} (I - B_\ell^{-1/2} A_\ell B_\ell^{-1/2})^\nu \|_{\ell^2} \\
&= C_S \| W (I - W)^\nu \|_{\ell^2}
\end{aligned}$$

using $W := B_\ell^{-1/2} A_\ell B_\ell^{-1/2}$. Since W is symmetric and positive definite, we immediately obtain that also

$$W(I - W)^\nu = -\sum_{i=0}^{\nu} \begin{pmatrix} \nu \\ i \end{pmatrix} (-W)^{i+1}$$

is symmetric. Thus, we obtain

$$\Psi_S(\nu) = C_S \rho(W(I - W)^\nu).$$

There is a singular value decomposition of W:

$$W = QDQ^{-1},$$

where $D = (\lambda_1, \ldots, \lambda_N)$ with $\lambda_1 \leq \cdots \leq \lambda_N$ is diagonal. Thus,

$$\begin{aligned}
\Psi_S(\nu) &= C_S \rho(QD(I - D)^\nu Q^{-1}) = C_S \rho(D(I - D)^\nu) = C_S \max_{i=1,\ldots,N} \lambda_i (1 - \lambda_i)^\nu \\
&\leq C_S \sup_{\lambda \in [\lambda_1, \lambda_n]} \lambda (1 - \lambda)^\nu.
\end{aligned}$$

Since W is symmetric and positive definite, we have $\lambda > 0$. Since $\rho(W) = \rho(\widehat{A}_\ell^{-1} A_\ell) \leq 1$ by assumption, we have $\lambda \leq 1$. Thus, we obtain

$$\Psi_S(\nu) \leq C_S \sup_{\lambda \in [0,1]} \underbrace{\lambda (1 - \lambda)^\nu}_{f(\lambda) :=}.$$

We have $f(0) = f(1) = 0$ and $f(x) \geq 0$ for $x \in [0,1]$. So, the maximum must be taken in the interior. Observe $f'(\lambda) = (1-\lambda)^\nu - \lambda\nu(1-\lambda)^{\nu-1} = (1-(\nu+1)\lambda)(1-\lambda)^{\nu-1}$. Its only root in $(0,1)$ is $\lambda = (\nu+1)^{-1}$. Thus,

$$\Psi_S(\nu) \leq C_S f\left(\frac{1}{\nu+1}\right) = C_S \frac{1}{\nu+1}\left(\frac{\nu}{\nu+1}\right)^\nu \leq \frac{C_S}{\nu+1}.$$

□

Remark 9.3. *This analysis does not cover the Gauss-Seidel smoother, since it is represented by*

$$B_\ell = D_\ell + R_\ell,$$

where D_ℓ is a diagonal matrix and R_ℓ a (strict) lower triangular matrix such that $A_\ell = D_\ell + R_\ell + R_\ell^T$.
The symmetric Gauss-Seidel smoother, which consists of one forward Gauss-Seidel sweep and one backward Gauss-Seidel sweep, is represented by the matrix

$$B_\ell = A_\ell + R_\ell D_\ell^{-1} R_\ell^T.$$

Thus, the results of the last Theorem are applicable (provided that we find a good estimate $A_\ell + R_\ell D_\ell^{-1} R_\ell^T \leq C_S A_\ell$).

Before, we apply our theory to IgA in the next section, we show that the convergence of the two-grid method implies the convergence of the W-cycle multigrid method. (For a convergence analysis of the V-cycle multigrid method, other tools are required.)
Let (for $\ell = 1, \ldots, \mathcal{L}$)

$$T_\ell := (I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell)(I - \widehat{A}_\ell^{-1} A_\ell)^\nu$$

be the iteration matrix of the two-grid method. Now, we assume that we already know the convergence of the two-grid method, i.e.,

$$\|T_\ell\|_{L_\ell} \leq q < 1$$

for all ℓ. Since T_ℓ is self-adjoined in $(\cdot,\cdot)_{A_\ell}$, we have

$$\|T_\ell\|_{A_\ell} = \rho(T_\ell) \leq \|T_\ell\|_{L_\ell} \leq q < 1,$$

i.e., convergence in the *energy norm* $\|\cdot\|_{A_\ell}$.
Now, let W_ℓ be the iteration matrix of the W-cycle ($\mu = 2$) or the V-cycle ($\mu = 1$) multigrid method. We have for all $\ell = 1, \ldots, \mathcal{L}$

$$\begin{aligned} W_\ell &= (I - P_\ell (I - W_{\ell-1}^\mu) A_{\ell-1}^{-1} P_\ell^T A_\ell)(I - \widehat{A}_\ell^{-1} A_\ell)^\nu \\ &= T_\ell + P_\ell W_{\ell-1}^\mu A_{\ell-1}^{-1} P_\ell^T A_\ell (I - \widehat{A}_\ell^{-1} A_\ell)^\nu \end{aligned}$$

where we use $W_0 := 0$. Now, we show that the multigrid method is only a *small* perturbation of the two-grid method. We obtain using the triangle inequality and semi-multiplicativity of

the norms that

$$\begin{aligned}\widetilde{q}_\ell := \|W_\ell\|_{A_\ell} &\le \|T_\ell\|_{A_\ell} + \|P_\ell(I - W_{\ell-1})^\mu A_{\ell-1}^{-1} P_\ell^T A_\ell (I - \widehat{A}_\ell^{-1} A_\ell)^\nu\|_{A_\ell} \\ &\le q + \|P_\ell W_{\ell-1}^\mu A_{\ell-1}^{-1} P_\ell^T A_\ell\|_{A_\ell} \|I - \widehat{A}_\ell^{-1} A_\ell\|_{A_\ell}^\nu \\ &\le q + \|A_\ell^{1/2} P_\ell A_{\ell-1}^{-1/2}\|_{\ell^2} \|A_{\ell-1}^{1/2} W_{\ell-1}^\mu A_{\ell-1}^{-1/2}\|_{\ell^2} \|A_{\ell-1}^{-1/2} P_\ell^T A_\ell^{1/2}\|_{\ell^2} \|I - \widehat{A}_\ell^{-1} A_\ell\|_{A_\ell}^\nu \\ &= q + \underbrace{\|A_\ell^{1/2} P_\ell A_{\ell-1}^{-1/2}\|_{\ell^2}^2}_{= \rho(P_\ell^T A_\ell P_\ell A_{\ell-1}^{-1}) = \rho(I) = 1} \underbrace{\|W_{\ell-1}\|_{A_{\ell-1}}^\mu}_{= \widetilde{q}_{\ell-1}^2} \|I - \widehat{A}_\ell^{-1} A_\ell\|_{A_\ell}^\nu.\end{aligned}$$

Using $\rho(\widehat{A}_\ell^{-1} A_\ell) \le 1$, we obtain further $\widetilde{q}_\ell \le q + \widetilde{q}_{\ell-1}^\mu$ and using $W_0 = 0$ the initial condition $\widetilde{q}_0 = 0$. Thus, we have

$$\widetilde{q}_{\mathcal{L}} \le \underbrace{q + (q + (q + \cdots)^\mu)^\mu}_{\mathcal{L} \text{ times}}.$$

For $\mu = 1$, we just obtain

$$\widetilde{q}_{\mathcal{L}} \le \mathcal{L} q,$$

which would be only uniformly bounded away from 1 if $q = \mathcal{O}(\mathcal{L}^{-1})$. We cannot assume this in general.

For $\mu = 2$, we obtain

$$\widetilde{q}_{\mathcal{L}} \le q + (q + (q + \cdots)^2)^2.$$

For $0 \le q \le \frac{1}{4}$, we obtain by induction that

$$\widetilde{q}_{\mathcal{L}} \le 1 - \sqrt{1 - 4q} \le 4q.$$

Provided $q \le \frac{C_A C_S}{1+\nu}$, we obtain that the W-cycle convergence rate is bounded by

$$\widetilde{q}_{\mathcal{L}} \le \frac{4 C_A C_S}{1 + \nu},$$

i.e., the multigrid method converges if sufficiently many smoothing steps are applied.

9.4 Multigrid solvers for Isogeometric Analysis

To progress further, we have to discuss a particular model problem. Consider a standard Poisson problem with Neumann boundary conditions.

$$-\Delta u = f \quad \text{in} \quad \Omega, \qquad \int_\Omega u \,\mathrm{d}x = 0, \qquad \frac{\partial}{\partial n} u = 0 \quad \text{on} \quad \partial\Omega.$$

Its variational formulation is as follows. Find $u \in V := H^1(\Omega)$ such that

$$(u, v)_{H^{1,\circ}(\Omega)} = (f, v)_{L_2(\Omega)} \quad \text{for all} \quad v \in V_\ell, \tag{9.3}$$

where

$$(u, v)_{H^{1,\circ}(\Omega)} := (\nabla u, \nabla v)_{L_2(\Omega)} + (u, 1)_{L_2(\Omega)} (v, 1)_{L_2(\Omega)}.$$

We now consider the *approximation property*

$$\|L_\ell^{1/2} (I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell) A_\ell^{-1} L_\ell^{1/2}\|_{\ell^2} \le C_A.$$

Note that $P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell$ represents the A_ℓ-orthogonal projection into the space $V_{\ell-1}$, embedded in A_ℓ.

Thus, we have $(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell)^2 = (I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell)$ and further

$$\begin{aligned}
&\|L_\ell^{1/2}(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell) A_\ell^{-1} L_\ell^{1/2}\|_{\ell^2} \\
&\quad \le \|L_\ell^{1/2}(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell) A_\ell^{-1/2}\|_{\ell^2} \|A_\ell^{1/2}(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell) A_\ell^{-1} L_\ell^{1/2}\|_{\ell^2} \\
&\quad \le \|L_\ell^{1/2}(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell) A_\ell^{-1/2}\|_{\ell^2}^2.
\end{aligned}$$

Thus, the approximation property can be equivalently rewritten as

$$\|(I - P_\ell A_{\ell-1}^{-1} P_\ell^T A_\ell)\underline{u}_\ell\|_{L_\ell} \le C_A^{1/2} \|\underline{u}_\ell\|_{A_\ell} \quad \text{for all} \quad \underline{u}_\ell \in \mathbb{R}^{N_\ell}. \tag{9.4}$$

In a Hackbusch-like analysis, convergence is typically shown in the L_2 norm. If we would follow that approach, we would define $L_\ell := h_\ell^{-2} M_\ell$, where M_ℓ is the mass matrix and h_ℓ the grid size. To be able to give robust estimates in h and p, we define

$$L_\ell := A_\ell + h_\ell^{-2} M_\ell.$$

Observe that using a standard inverse estimate, we have

$$h_\ell^{-2} M_\ell \le L_\ell \lesssim h^{-2} p^4 M_\ell.$$

This means that if we are only interested in h-robustness, the terms L_ℓ and $h_\ell^{-2} M_\ell$ are equivalent. So, in this case we could follow the standard definition.

Using the choice of L_ℓ, the approximation property reads as follows.

$$\|(I - \Pi_{\ell-1})u_\ell\|_{H^{1,\circ}(\Omega)}^2 + h^{-2}\|(I - \Pi_{\ell-1})u_\ell\|_{L_2(\Omega)}^2 \le C_A \|u_\ell\|_{H^{1,\circ}(\Omega)}^2 \quad \text{for all} \quad u_\ell \in V_\ell, \tag{9.5}$$

where $\Pi_{\ell-1}$ is the $H^{1,\circ}(\Omega)$-orthogonal projection into $V_{\ell-1}$.

The estimate (9.5) is satisfied if both

$$\|(I - \Pi_{\ell-1})u_\ell\|_{H^{1,\circ}(\Omega)} \le \|u_\ell\|_{H^{1,\circ}(\Omega)} \quad \text{for all} \quad u_\ell \in V_\ell, \tag{9.6}$$

and

$$\|(I - \Pi_{\ell-1})u_\ell\|_{L_2(\Omega)} \lesssim h\|u_\ell\|_{H^{1,\circ}(\Omega)} \quad \text{for all} \quad u_\ell \in V_\ell, \tag{9.7}$$

are satisfied. The estimate (9.6) is obviously true since the $H^{1,\circ}(\Omega)$-orthogonal projector Π_ℓ does not increase the $H^{1,\circ}(\Omega)$-norm.

So, it remains to show (9.7). By transposing, we obtain that this equation reads as

$$\|(I - \Pi_{\ell-1})u_\ell\|_{H^{1,\circ}(\Omega)} \lesssim h \sup_{w_\ell \in V_\ell} \frac{(u_\ell, w_\ell)_{H^{1,\circ}(\Omega)}}{\|w_\ell\|_{L_2(\Omega)}} \quad \text{for all} \quad u_\ell \in V_\ell, \tag{9.8}$$

This statement can be shown if the following regularity assumption is satisfied.

Assumption 9.4. *For all $f \in L_2(\Omega)$, the solution $u \in H^1(\Omega)$ of the problem (9.3) satisfies*

$$u \in H^2(\Omega) \quad \textit{and} \quad |u|_{H^2(\Omega)} \lesssim \|f\|_{L_2(\Omega)}.$$

For this regularity assumption, cf. Section 1.5.1.7.

Now, we proceed as follows. Let $u_\ell \in V_\ell$ be arbitrary but fixed. Define $f_\ell \in V_\ell$ such that

$$(u_\ell, q_\ell)_{H^{1,\circ}(\Omega)} = (f_\ell, q_\ell)_{L_2(\Omega)} \quad \text{for all} \quad q_\ell \in V_\ell$$

Now, define $u \in V$ to be such that

$$(u, q)_{H^{1,\circ}(\Omega)} = (f_\ell, q)_{L_2(\Omega)} \quad \text{for all} \quad q \in V$$

and observe that Assumption 9.4 yields $u \in H^2(\Omega)$ and

$$|u|_{H^2(\Omega)} \lesssim \|f\|_{L_2(\Omega)} = \sup_{w_\ell \in V_\ell} \frac{(f_\ell, w_\ell)_{L_2(\Omega)}}{\|w_\ell\|_{L_2(\Omega)}} = \sup_{w_\ell \in V_\ell} \frac{(u_\ell, w_\ell)_{H^{1,\circ}(\Omega)}}{\|w_\ell\|_{L_2(\Omega)}}. \tag{9.9}$$

Now, observe that by construction $u_\ell = \Pi_\ell u$. Thus, we also have $\Pi_{\ell-1} u_\ell = \Pi_{\ell-1} u$. Using the triangle inequality, we obtain

$$\|(I - \Pi_{\ell-1})u_\ell\|_{H^{1,\circ}(\Omega)} \leq \|(I - \Pi_\ell)u\|_{H^{1,\circ}(\Omega)} + \|(I - \Pi_{\ell-1})u\|_{H^{1,\circ}(\Omega)}.$$

The only remaining piece of the puzzle is an approximation error estimate. For equidistant grids, we might use the h- and p- robust estimates. For other grids, we might use an estimate which is only robust in h. In any case, we obtain

$$\|(I - \Pi_{\ell-1})u_\ell\|_{H^{1,\circ}(\Omega)} \lesssim \underbrace{(h_\ell + h_{\ell-1})}_{\eqsim h_\ell} |u|_{H^2(\Omega)},$$

where the constants in $\lesssim$ and $\eqsim$ are independent of the grid size and might also be independent of the spline degree. Using (9.9), we obtain

$$\|(I - \Pi_{\ell-1})u_\ell\|_{H^{1,\circ}(\Omega)} \lesssim h_\ell \sup_{w_\ell \in V_\ell} \frac{(u_\ell, w_\ell)_{H^{1,\circ}(\Omega)}}{\|w_\ell\|_{L_2(\Omega)}},$$

i.e., (9.8) and, therefore, the approximation property.

The last remaining step is to find a smoother B_ℓ that satisfies

$$A_\ell \leq B_\ell \lesssim L_\ell = A_\ell + h_\ell^{-2} M_\ell.$$

Here, we first consider standard smoothers (Jacobi, Gauss-Seidel), where the constant hidden in the symbol $\lesssim$ is independent of the grid size, but depends on the spline degree, see Section 9.4.1. Then, we discuss a smoother where this hidden constant is independent of both, the grid size and the spline degree, see Section 9.4.2.

9.4.1 Jacobi and (symmetric) Gauss-Seidel smoothers

The *Jacobi smoother* is given by

$$B_\ell := \tau^{-1} \operatorname{diag} A_\ell,$$

where $\tau > 0$ is chosen such that $B_\ell \geq A_\ell$.

Lemma 9.5. *$A_\ell \leq (p+1)^d \operatorname{diag} A_\ell$ holds.*

Proof. Let $a_{i,j}$ be the coefficients of A_ℓ. We have using Cauchy-Schwarz inequality

$$a_{i,j} = a(\phi_i, \phi_j) \leq a(\phi_i, \phi_i)^{1/2} a(\phi_j, \phi_j)^{1/2} = a_{i,i}^{1/2} a_{j,j}^{1/2}.$$

Now, observe for all $\underline{u}_\ell = (u_1, \dots, u_{N_\ell})$ that

$$(A_\ell u_\ell, u_\ell)_{\ell^2} = \sum_{i,j} a_{i,j} u_i u_j \leq \sum_{i,j} (a_{i,i}^{1/2} u_i a_{j,j}^{1/2} u_j) \leq \frac{1}{2} \sum_{i,j} (a_{i,i} u_i^2 + a_{j,j} u_j^2).$$

As we have not more than $(p+1)^d$ non-zero entries per row, we further obtain

$$\begin{aligned}(A_\ell u_\ell, u_\ell)_{\ell^2} &\leq \frac{(p+1)^d}{2} \left(\sum_i a_{i,i} u_i^2 + \sum_j a_{j,j} u_j^2 \right) = (p+1)^d \left(\sum_i a_{i,i} u_i^2 \right) \\ &= (p+1)^d (\operatorname{diag} A_\ell u_\ell, u_\ell)_{\ell^2}.\end{aligned}$$

□

This shows that we can choose $\tau \eqsim (p+1)^{-d}$ independently of the grid size to obtain

$$A_\ell \leq B_\ell.$$

For showing the other direction, we use the following theorem.

Theorem 9.6. *Let M_h be the mass matrix for the space $S_{p,p-1,h}(0,1)$. Then, $\kappa(M_h) < p\, 2^p$.*

For a proof, see [31]. The theorem is similar to the deBoor conjecture itself. The conjecture is not proven; it states $\kappa(M_h) < 2^p$. The extension to tensor-product splines is straight-forward and immediately yields

$$\kappa(M_\ell) \lesssim p^d\, 2^{pd},$$

where $\lesssim$ depends also on the geometry function.
Using Theorem 9.6, we immediately obtain

$$\operatorname{diag} M_\ell \leq \lambda_{\max}(M_\ell) I \lesssim p^d\, 2^{pd} \lambda_{\min}(M_\ell) I \leq p^d\, 2^{pd} M_\ell.$$

Using a simple scaling argument, we obtain

$$\operatorname{diag} M_\ell \eqsim \operatorname{diag} A_\ell.$$

By combining these estimates, we obtain

$$A_\ell \leq B_\ell \lesssim p^d\, 2^{pd} (A_\ell + h_\ell^{-2} M_\ell).$$

Thus, we obtain using Theorem 9.2 the smoothing property and using the approximation property the convergence of the two-grid method. Using the proof outlined in the end of Section 9.3, we obtain the convergence of the W-cycle multigrid method.

Theorem 9.7. *The W-cycle multigrid method with Jacobi smoother with $\tau \eqsim (p+1)^{-d}$ small enough converges robustly in the grid size h_ℓ if sufficiently many smoothing steps are applied. The required number of smoothing steps and the convergence rate are independent of the grid size.*

The *symmetric Gauss-Seidel smoother* is given by

$$B_\ell := A_\ell + R_\ell D_\ell^{-1} R_\ell^T,$$

where $B_\ell \geq A_\ell$ is obviously true. Using arguments that are similar to those of the proof of Lemma 9.5, we obtain

$$R_\ell D_\ell^{-1} R_\ell^T \lesssim D_\ell = \operatorname{diag} A_\ell.$$

Thus, using the arguments derived for the Jacobi smoother, we again obtain

$$A_\ell \leq B_\ell \lesssim p^d \, 2^{pd} (A_\ell + h_\ell^{-2} M_\ell)$$

and the following result.

Theorem 9.8. *The W-cycle multigrid method with symmetric Gauss-Seidel smoother converges robustly in the grid size h_ℓ if sufficiently many smoothing steps are applied. The required number of smoothing steps and the convergence rate are independent of the grid size.*

Numerical experiments show that these multigrid method work very well for small spline degrees. As the theory predicts, the convergence rates are robust in the grid size. However, if the spline degree is increased, one sees that the convergence rates deteriorate significantly.

9.4.2 Subspace corrected mass smoother

In this subsection, we derive a smoother $B_\ell := \tau^{-1} \widehat{B}_\ell$ that satisfies

$$A_\ell \lesssim \widehat{B}_\ell \lesssim A_\ell + h_\ell^{-2} M_\ell$$

robustly in the grid size h and the spline degree p.

Let $\widehat{A}_\ell$ and $\widehat{M}_\ell$ are stiffness matrix and mass matrix corresponding to the parameter domain, cf. Section 8.2. For this case we have $A_\ell \eqsim \widehat{A}_\ell$ and $M_\ell \eqsim \widehat{M}_\ell$.

To do so, we first remember the space $\widetilde{S}_{p,h}(0,1)$, as defined in Theorem 4.4:

$$\widetilde{S}_{p,h}(0,1) := \{v_h \in S_{p,h}(0,1) \ : \ \tfrac{d^r}{dx^r} v_h(0) = \tfrac{d^r}{dx^r} v_h(1) = 0 \text{ for } r = 1, 3, \ldots, 2\lceil \tfrac{p-1}{2} \rceil - 1\}.$$

Now, we write

$$W := S_{p,h}(0,1), \qquad W_0 := \widetilde{S}_{p,h}(0,1)$$

and

$$W_1 := \{w \in W \ : \ (w,q)_{L_2(0,1)} = 0 \text{ for all } q \in W_0\}$$

for the L_2-orthogonal complement. Let Q_0 be the L_2-orthogonal projector $W \to W_0$ and $Q_1 = I - Q_0$ be the the L_2-orthogonal projector $W \to W_1$.

Since the projector is L_2-orthogonal, we immediately obtain

$$u = Q_0 u + Q_1 u \qquad \text{and} \qquad \|u\|^2_{L_2(0,1)} = \|Q_0 u\|^2_{L_2(0,1)} + \|Q_1 u\|^2_{L_2(0,1)}.$$

Using the robust inverse estimate Theorem 4.4 and the robust approximation error estimate Theorem 3.9, we obtain

$$|u|^2_{H^1(0,1)} \eqsim |Q_0 u|^2_{H^1(0,1)} + |Q_1 u|^2_{H^1(0,1)}$$

for all $u \in W$, where the constant hidden in $\eqsim$ is independent of the grid size and the spline degree. A proof for this statement is given in [32, Theorem 4].

This result can be directly carried over to the multi-dimensional case. We restrict ourselves for simplicity to $d = 2$ with $\widehat{\Omega} = (0,1)^2$. Here, we have the spaces $W_{\alpha,\beta} := W_\alpha \otimes W_\beta$ and the corresponding L_2-orthogonal projectors $Q_{\alpha,\beta} = Q_\alpha \times Q_\beta$. Again, analogous stability results, cf. [32, Theorem 5], hold:

$$\|u\|^2_{L_2(\widehat{\Omega})} = \sum_{(\alpha,\beta)\in\{0,1\}^2} \|Q_{\alpha,\beta} u\|^2_{L_2(\widehat{\Omega})} \quad \text{and} \quad |u|^2_{H^1(\widehat{\Omega})} \eqsim \sum_{(\alpha,\beta)\in\{0,1\}^2} |Q_{\alpha,\beta} u|^2_{H^1(\widehat{\Omega})}$$

for all $u \in W \otimes W = S_{p,h}(\widehat{\Omega})$. By adding up (and using the matrix-vector notation), we obtain

$$\|\underline{u}_\ell\|^2_{A_\ell + h_\ell^{-2} M_\ell} \eqsim \sum_{(\alpha,\beta)\in\{0,1\}^2} \|Q_{\alpha,\beta}\underline{u}_\ell\|^2_{A_\ell + h_\ell^{-2} M_\ell}.$$

Here, $Q_{\alpha,\beta} = Q_\alpha \otimes Q_\beta = (P_\alpha M_\alpha^{-1} P_\alpha^T M) \otimes (P_\beta M_\beta^{-1} P_\beta^T M)$ are the matrix representations of the L_2-orthogonal projectors, where $M_\alpha := P_\alpha^T M P_\alpha$ are the *univariate* mass matrices in W_α and P_α represents the canonical embedding $W_\alpha \to W$. Analogously, we define *univariate* stiffness matrices $A_\alpha := P_\alpha^T A P_\alpha$. Based on these definitions, we define $P_{\alpha,\beta} := P_\alpha \otimes P_\beta$ and $\widehat{M}_{\alpha,\beta} := P_{\alpha,\beta}^T \widehat{M}_\ell P_{\alpha,\beta} = M_\alpha \otimes M_\beta$ and $\widehat{A}_{\alpha,\beta} := P_{\alpha,\beta}^T \widehat{A}_\ell P_{\alpha,\beta} = A_\alpha \otimes M_\beta + M_\alpha \otimes A_\beta$. Using these definitions, we obtain

$$\begin{aligned} &A_\ell + h_\ell^{-2} M_\ell \eqsim \widehat{A}_\ell + h_\ell^{-2}\widehat{M}_\ell \\ &\eqsim \sum_{(\alpha,\beta)\in\{0,1\}^2} M_\ell (P_\alpha M_\alpha^{-1} P_\alpha^T \otimes P_\beta M_\beta^{-1} P_\beta^T)(\widehat{A}_{\alpha,\beta} + h_\ell^{-2}\widehat{M}_{\alpha,\beta})(P_\alpha M_\alpha^{-1} P_\alpha^T \otimes P_\beta M_\beta^{-1} P_\beta^T) M_\ell. \end{aligned}$$

and

$$(A_\ell + h_\ell^{-2} M_\ell)^{-1} \eqsim \sum_{(\alpha,\beta)\in\{0,1\}^2} (P_\alpha \otimes P_\beta)(\widehat{A}_{\alpha,\beta} + h_\ell^{-2}\widehat{M}_{\alpha,\beta})^{-1}(P_\alpha^T \otimes P_\beta^T). \tag{9.10}$$

Theorem 4.4 yields $A_0 \lesssim h_\ell^{-2} M_0$. Thus, we have

$$\begin{aligned} \widehat{A}_{0,0} + h_\ell^{-2}\widehat{M}_{0,0} &= A_0 \otimes M_0 + M_0 \otimes A_0 + h_\ell^{-2} M_0 \otimes M_0 && \lesssim h_\ell^{-2} M_0 \otimes M_0 =: \widehat{B}_{0,0} \\ \widehat{A}_{1,0} + h_\ell^{-2}\widehat{M}_{1,0} &= A_1 \otimes M_0 + M_1 \otimes A_0 + h_\ell^{-2} M_1 \otimes M_0 && \lesssim (A_1 + h_\ell^{-2} M_1) \otimes M_0 =: \widehat{B}_{1,0} \\ \widehat{A}_{0,1} + h_\ell^{-2}\widehat{M}_{0,1} &= A_0 \otimes M_1 + M_0 \otimes A_1 + h_\ell^{-2} M_0 \otimes M_1 && \lesssim M_0 \otimes (A_1 + h_\ell^{-2} M_1) =: \widehat{B}_{0,1} \\ \widehat{A}_{1,1} + h_\ell^{-2}\widehat{M}_{1,1} &= A_1 \otimes M_1 + M_1 \otimes A_1 + h_\ell^{-2} M_1 \otimes M_1 && =: \widehat{B}_{1,1}. \end{aligned} \tag{9.11}$$

The matrices $\widehat{B}_{0,0}$, $\widehat{B}_{1,0}$ and $\widehat{B}_{0,1}$ are tensor-products and thus easy to invert using the techniques discussed in Section 8.2. The matrix $\widehat{B}_{1,1}$ has a small dimension; so the corresponding systems can be solved using a direct solver.

Based on this observation, we define

$$\widehat{B}_\ell^{-1} := \sum_{(\alpha,\beta)\in\{0,1\}^2} (P_\alpha \otimes P_\beta)\widehat{B}_{\alpha,\beta}^{-1}(P_\alpha^T \otimes P_\beta^T).$$

The combination of this definition, (9.11) and (9.10) yields

$$\widehat{B}_\ell^{-1} \eqsim (A_\ell + h_\ell^{-2} M_\ell)^{-1}.$$

Thus, we have

$$\widehat{B}_\ell \eqsim A_\ell + h_\ell^{-2} M_\ell$$

and using an appropriate choice (independent of the grid size and the spline degree) of τ also

$$A_\ell \leq \underbrace{\tau^{-1}\widehat{B}_\ell}_{B_\ell =} \lesssim A_\ell + h_\ell^{-2} M_\ell.$$

Thus, we obtain using Theorem 9.2 the smoothing property and using the approximation property the convergence of the two-grid method. Using the proof outlined in the end of Section 9.3, we obtain the convergence of the W-cycle multigrid method.

Theorem 9.9. *The W-cycle multigrid method with subspace corrected mass smoother with an appropriate choice of τ robustly in the grid size h_ℓ* and *the spline degree p if sufficiently many smoothing steps are applied. The required number of smoothing steps and the convergence rate are independent of the grid size and the spline degree.*

This multigrid method works very well in numerical experiments. As the theory predicts, the convergence rates are robust in the grid size and the spline degree. The method suffers if the geometry function gets too distorted.

Numerical experiments have shown that a hybrid method which combines the strength of the Gauss-Seidel smoother and the mass smoother works very well. This hybrid approach consists of the following steps:

- One forward Gauss-Seidel sweep
- ν steps of the subspace corrected mass smoother
- Coarse-grid correction
- ν steps of the subspace corrected mass smoother
- One backward Gauss-Seidel sweep,

where we typically choose $\nu = 1$.

Note that a proof indicating that this method is robust in the geometry function is not known. (Such a proof is also not known if only the Gauss-Seidel smoother is used.) However, one can easily extend the proof of the p and h robust convergence of the W-cycle multigrid method with subspace corrected mass smoother to this hybrid smoother.

9.5 Literature

The classical multigrid method follows the ideas presented in [30] The results on the approximation property and on the W-cycle convergence follow [33]. Finally, the last subsection on the subspace corrected mass smoother follows [32].

Chapter 10

Multi-patch Isogeometric Analysis

10.1 Motivation

So far, we have considered only computational domains that are parameterized by one (global) geometry function. We call this the *single-patch* case. Having only one global geometry function restricts the set of possible computational domains to domains that are topologically equivalent to the unit square or the unit cube. This means, for example, that computational domains with holes cannot be considered.

Since this is too restrictive, we need a generalization of the classical isogeometric approach. The idea of *multi-patch* isogeometric analysis is to decompose the whole computational domain Ω into patches Ω_k, where each of the patches is decomposed with its own geometry function.

In general, we can distinguish between two cases:

- *Non-overlapping decompositions:* Here, we assume that the patches Ω_k are simply connected open sets, which are disjoint:

  $$\Omega_k \cap \Omega_l = \emptyset \quad \text{for all} \quad k \neq l. \tag{10.1}$$

  Moreover, we assume that the closures of the patches cover the closure of the whole domain:

  $$\overline{\Omega} = \bigcup_{k=1}^{K} \overline{\Omega_k}. \tag{10.2}$$

  Now, the basic idea is to set up isogeometric function spaces on each of the patches. These patch-local function spaces are then combined to a global function space.

  One possibility is to do this in a *conforming* way (Section 10.2), which means that the resulting discrete function space is a subspace of the continuous function space V. Here and in what follows, we assume to consider the standard Poisson problem with homogenous Neumann boundary conditions. So, $V = H^1(\Omega)$.

  Alternatively, the global space can be defined in a *non-conforming* way (Section 10.3), which means that the resulting discrete function space is not a subspace of the continuous function space V. In this case, the setup of the variational formulation (integration by parts) cannot be done in the usual way. In Section 10.3, we will discuss alternative approaches.

- *Overlapping decompositions:* Here, we again assume that the patches Ω_k are simply connected open sets such that (10.2) holds.

  The condition (10.1) is not satisfied for such decompositions. Thus, on the overlaps, we have several function values: consider the case of 2 patches Ω_1 and Ω_2. On each of the patches, we have a discrete solution: $u_h^{(1)}$ and $u_h^{(2)}$. Thus, on the overlap $\Omega_1 \cap \Omega_2$, both solution functions $u_h^{(1)}$ and $u_h^{(2)}$ are defined.

  Here, it is not clear how the solution on the overlap should be interpreted. It is possible that the discretization is set up in a way that the overall discrete solution u_h (which aims to approximate the solution u of the original problem) is just the sum:

$$u_h = \begin{cases} u_h^{(1)} + u_h^{(2)} & \text{on } \Omega_1 \cap \Omega_2 \\ u_h^{(1)} & \text{on } \Omega_1 \backslash \Omega_2 \\ u_h^{(2)} & \text{on } \Omega_2 \backslash \Omega_1 \end{cases}.$$

  An alternative is that both $u_h^{(1)}$ and $u_h^{(2)}$ aim to approximate the original problem. In this case, one can define the overall discrete solution u_h on the overlap to be the average:

$$u_h = \begin{cases} \frac{1}{2}(u_h^{(1)} + u_h^{(2)}) & \text{on } \Omega_1 \cap \Omega_2 \\ u_h^{(1)} & \text{on } \Omega_1 \backslash \Omega_2 \\ u_h^{(2)} & \text{on } \Omega_2 \backslash \Omega_1 \end{cases}.$$

  If one is interested in a continuous version of that approach, one can introduce distance functions $d_i : \Omega \to \mathbb{R}$, which could look like

$$d_i(x) := \begin{cases} \operatorname{dist}(x, \partial\Omega_i) & \text{on } \Omega_i \\ 0 & \text{otherwise} \end{cases}$$

  for $i \in \{1, 2\}$, where $\operatorname{dist}(x, \partial\Omega_i) = \inf_{y \in \partial\Omega_i} \|x - y\|_{\ell^2}$, and define

$$u_h = \tfrac{1}{d_1 + d_2}(d_1 u_h^{(1)} + d_2 u_h^{(2)}).$$

  We will not go too much into the details of these approaches.

10.2 Conforming discretizations

For simplicity, we assume $d = 2$. We assume that the open domain Ω consists of the patches $\Omega_1, \ldots, \Omega_k$:

$$\overline{\Omega} = \bigcup_{k=1}^{K} \overline{\Omega_k},$$

where each patch Ω_k is simply connected and open. Moreover, we assume that the patches are disjoint:

$$\Omega_k \cap \Omega_l = \emptyset \quad \text{for all} \quad k \neq l.$$

We assume (as for the single patch case) that each of the patches is represented by a bijective geometry function

$$G_k : \widehat{\Omega} := (0,1)^2 \to \Omega_k := G_k(\widehat{\Omega}) \subset \mathbb{R}^2,$$

which can be continuously extended to the closure of $\widehat{\Omega}$.

Conforming discretizations are typically only set up on discretizations without T-junctions.

Assumption 10.1. *The intersection of $\overline{\Omega_k}$ and $\overline{\Omega_l}$ for $k \neq l$ is either (a) empty, (b) one common vertex or (c) the union of one common edge and two common vertices.*

Now, having a representation of the domain, we introduce the isogeometric function space. First, we define spline spaces $\widehat{V}_k$ (possibly different for each of the patches) for the parameter domain. On the physical patch Ω_k, we define the ansatz functions using the pull-back principle

$$V_k := \{u \in H^1(\Omega_k) \ : \ u \circ G_k \in \widehat{V}_k\}. \tag{10.3}$$

The *multi-patch function space* V_h is given by

$$V_h := \{u \in H^1(\Omega) \ : \ u|_{\Omega_k} \in V_k \text{ for } k = 1, \dots, K\}. \tag{10.4}$$

This definition states that the restriction of u to any patch is an isogeometric function and moreover that u is continuous (this is a sufficient and necessary for $u \in H^1(\Omega)$).

This means that we impose certain conditions. If we do so, we have to make sure that the conditions do not contradict each other. If they would contradict, the space V_h would be not rich enough. This can be clarified by an example. Let $\Omega_1 := (0,1)^2$ and $\Omega_2 := (1,2) \times (0,1)$. Ω_1 is discretized with

$$\widehat{V}_1 = S_{p,h} \otimes S_{p,h}, \qquad G_1(x,y) = (x,y)$$

and Ω_2 is discretized with

$$\widehat{V}_2 = S_{q,h} \otimes S_{q,h}, \qquad G_2(x,y) = (1+x,y)$$

and $q < p$.

When restricting V_1 and V_2 to the interface $T = \{1\} \otimes (0,1)$, we obtain the spline spaces

$$S_{p,h} \qquad \text{and} \qquad S_{q,h}.$$

In (10.9), we have assumed the spline spaces to agree on the interface. Thus, all functions in V_h are, when restricted to T, are in $S_{p,h}$ and in $S_{q,h}$:

$$u|_T \in C^{p-1}(0,1), \quad (u|_T)_{(ih,(i+1)h]} \in \mathbb{P}_p, \quad u|_T \in C^{q-1}(0,1), \quad (u|_T)_{(ih,(i+1)h]} \in \mathbb{P}_q.$$

This yields

$$u|_T \in C^{p-1}(0,1) \cap C^{q-1}(0,1) = C^{p-1}(0,1), \quad (u|_T)_{(ih,(i+1)h]} \in \mathbb{P}_p \cap \mathbb{P}_q = \mathbb{P}_q,$$

so, we have splines of degree q with smoothness $p - 1 \geq q$. Thus, we directly obtain

$$u|_T \in \mathbb{P}_q,$$

i.e., the function space on the interface *only consists of polynomials*. This is completely independent of the chosen grid size h. It is not hard to imagine that the function space V_h is not a good space for approximating the solution.

Similar effects are possible if the spline degrees agree, but the knot vectors are different.

The following assumption avoids such phenomena.

Assumption 10.2. *Let $(B_k^{(i)})_{i=1}^{N_k}$ be the basis of V_k for any $k = 1, \dots, K$. For any being a common edge $T = \partial\Omega_k \cap \partial\Omega_l$ of the patches Ω_k and Ω_l, we assume that the basis functions of the two patches match, i.e., for all i with $B_k^{(i)}|_T \neq 0$, there is some j such that*

$$B_k^{(i)}|_T = B_l^{(j)}|_T. \tag{10.5}$$

If a basis function contributes to several interfaces, we have to find matching basis functions for all neighboring patches.

If this assumption is satisfied, a global basis is obtained as follows.

- Take the basis functions whose support is in the interior of one patch.
- Take a basis functions whose support touches the interfaces. Take the basis function on the neighboring patch(es), such that (10.5) is satisfied. Add up all basis functions obtained with this method and add this sum to the global basis.

Denote the resulting basis with $\Phi = (\phi_i)_{i=1}^N$.
Using this construction, we see that

$$(V_h)|_{\Omega_k} = V_k,$$

i.e., that the restriction of V_h to any of the patches yields the whole spline space that was defined on the patch.
Having the function space, we can set up the usual discrete variational formulation

$$\text{find} \quad u_h \in V_h \quad \text{such that} \quad (\nabla u_h, \nabla v_h)_{L_2(\Omega)} = (f, v_h)_{L_2(\Omega)} \quad \text{for all} \quad v_h \in V_h$$

and using the basis Φ also the matrix-vector formulation

$$A_h \, \underline{u}_h = \underline{f}_h.$$

Some comments:

- *Matrix assembling:* This is straight-forward: one just assembles the patch-local matrices. Then, the matrices are just added up.
- *Existence and uniqueness of a solution:* This follows from Lax Milgram lemma, which does not depend on the discretization.
- *Approximation error estimates:* Approximation error estimates can be constructed as it was done in [34]. The basic idea is as follows. First one, constructs univariate projection operators that are interpolatory on the boundary (i.e., the ends of the unit interval). Then, one constructs for each patch a patch-local projection operator Π_k which is "interpolatory on the boundary". This means:
  1. The projector is interpolatory on the vertices.
  2. On the edges, the projector coincides with the corresponding univariate projector.

If this construction is done properly, one obtains

1. on each patch the desired error estimates

$$\|u - \Pi_k u\|_{L_2(\Omega_k)} \lesssim h_k^r |u|_{H^r(\Omega_k)} \tag{10.6}$$

and

2. that the projected function agree on the interfaces, i.e.,

$$\Pi_k u = \Pi_l u \quad \text{on} \quad \partial\Omega_k \cap \partial\Omega_l. \tag{10.7}$$

Thus, we can define a global projection operator Π via

$$(\Pi u)|_{\Omega_k} = \Pi_k u.$$

We obtain from (10.7) that $\Pi u \in V_h$. And from (10.6), we obtain

$$\|u - \Pi u\|_{L_2(\Omega)} \lesssim (\max_k h_k)^r |u|_{H^r(\Omega)}.$$

- *Iterative solvers:* Certainly, the standard solvers (Cholesky factorization, MATLAB backslash, conjugate gradient, multigrid with Jacobi or Gauss-Seidel smoother, ...) can be directly extended to the multi-patch case. Also the convergence analysis can be extended to the multipatch case; some work is required for the extension of the approximation error estimates that are required for a multigrid analysis, cf. [34] for such an extension.

  The other solvers (fast diagonalization, multigrid with subspace corrected mass smoother) cannot be directly extended to the multi-patch case. Here, methods from domain decomposition are required. The extension of the subspace corrected mass smoother to the multipatch case has been worked out in [34].

10.3 Non-conforming discretizations

For simplicity, we assume $d = 2$. We assume that the open domain Ω consists of the patches $\Omega_1, \ldots, \Omega_k$:

$$\overline{\Omega} = \bigcup_{k=1}^{K} \overline{\Omega_k},$$

where each patch Ω_k is simply connected and open. Moreover, we assume that the patches are disjoint:

$$\Omega_k \cap \Omega_l = \emptyset \quad \text{for all} \quad k \neq l.$$

We assume (as for the single patch case) that each of the patches is represented by a bijective geometry function

$$G_k : \widehat{\Omega} := (0,1)^2 \to \Omega_k := G_k(\widehat{\Omega}) \subset \mathbb{R}^2,$$

which can be continuously extended to the closure of $\widehat{\Omega}$. Having a representation of the domain, we introduce the isogeometric function space. First, we define spline spaces $\widehat{V}_k$

(possibly different for each of the patches) for the parameter domain. On the physical patch Ω_k, we define the ansatz functions using the pull-back principle

$$V_k := \{u \in H^1(\Omega_k) \ : \ u \circ G_k \in \widehat{V}_k\}. \tag{10.8}$$

The *multi-patch function space* V_h is given by

$$V_h := \{u \in L_2(\Omega) \ : \ u|_{\Omega_k} \in V_k \text{ for } k = 1, \dots, K\}. \tag{10.9}$$

This definition states that the restriction of u to any patch is an isogeometric function. We do not impose any continuity on u across the patches (the space $L_2(\Omega)$ does not imply any continuity condition.) A basis for V_h is simply obtained by taking the individual basis functions of the patches (so, the support of each basis function lies within one patch).

We have

$$V_h \not\subset H^1(\Omega).$$

Thus, we speak of a *non-conforming* method. Since the basis functions in V_h, we also speak of *discontinuous Galerkin* (dG) methods.

The dG methods have originally been developed for the FEM world, see [35] for a survey paper. One of the simpler dG methods is the *symmetric interior penalty discontinuous Galerkin* (SIPG) method. In [36, 37], the SIPG method has been used to couple patches in a multi-patch IgA framework.

Consider the following model problem. Let $f \in L_2(\Omega)$ be given. Find $u \in H^1(\Omega)$ such that

$$-\Delta u = f \quad \text{in} \quad \Omega$$

and

$$\frac{\partial}{\partial n} u = 0 \quad \text{on} \quad \partial\Omega.$$

Assume that the problem is such that the solution satisfies the *regularity assumption*

$$u \in H^2(\Omega).$$

Now, we multiply the PDE with a test function in V_h. Then, we obtain

$$-(\Delta u, v_h)_{L_2(\Omega)} = (f, v_h)_{L_2(\Omega)}.$$

Since v_h is not continuous, we cannot apply integration by parts. However, we can represent the scalar product as sum

$$-\sum_{k=1}^{K} (\Delta u, v_h)_{L_2(\Omega_k)} = (f, v_h)_{L_2(\Omega)}.$$

and apply integration by parts to obtain

$$\sum_{k=1}^{K} \Big((\nabla u, \nabla v_h)_{L_2(\Omega_k)} + \big(\underbrace{\frac{\partial}{\partial n} u}_{= \nabla u \cdot n}, v_h \big)_{L_2(\partial\Omega_k)} \Big) = (f, v_h)_{L_2(\Omega)}. \tag{10.10}$$

Now, introduce for all interfaces $I_{k,l} := \partial\Omega_k \cap \partial\Omega_l$ with $k < l$ the following objects:

- $\mathbf{n}$ is the outer normal vector of Ω_k. Thus, $-\mathbf{n}$ is the outer normal vector of Ω_l.
- $[\![u]\!]$ is the jump:
$$[\![u]\!] := (u|_{\Omega_k}) - (u|_{\Omega_l}).$$
- $\{u\}$ is the average:
$$\{u\} := \frac{1}{2}\big((u|_{\Omega_k}) + (u|_{\Omega_l})\big).$$

Since $u \in H^2(\Omega)$, we have $[\![u]\!] = 0$ and $\{\nabla u\} = (\nabla u)|_{\Omega_k} = (\nabla u)|_{\Omega_l}$. Using this notation and using the homogenous Neumann boundary conditions, (10.10) can be equivalently rewritten as

$$\sum_{k=1}^{K}(\nabla u, \nabla v_h)_{L_2(\Omega_k)} + \sum_{(k,l)}(\{\nabla u\}\cdot\mathbf{n}, [\![v_h]\!])_{L_2(I_{k,l})} = (f, v_h)_{L_2(\Omega)}. \tag{10.11}$$

Since $[\![u]\!] = 0$, we can rewrite this equivalently as

$$a_h(u, v_h) = (f, v_h)_{L_2(\Omega)}, \tag{10.12}$$

where

$$\begin{aligned} a_h(u,v) = &\sum_{k=1}^{K}(\nabla u, \nabla v)_{L_2(\Omega_k)} \\ &+ \sum_{(k,l)}(\{\nabla u\}\cdot\mathbf{n}, [\![v]\!])_{L_2(I_{k,l})} + \sum_{(k,l)}(\{\nabla v\}\cdot\mathbf{n}, [\![u]\!])_{L_2(I_{k,l})} \\ &+ \frac{\sigma}{h}\sum_{(k,l)}([\![u]\!], [\![v]\!])_{L_2(I_{k,l})}. \end{aligned}$$

With this choice, we obtain that $a_h(u,v)$ is symmetric. If σ is large enough, we obtain that $a_h(u,v)$ is bounded and coercive, i.e.,

$$a_h(u_h, v_h) \lesssim \|u_h\|_{Q_h}\|v_h\|_{Q_h} \qquad \text{and} \qquad a_h(u_h, u_h) \gtrsim \|u_h\|^2_{Q_h} \tag{10.13}$$

for all $u_h, v_h \in V_h$ and with the choice

$$\|u\|^2_{Q_h} := \sum_{k=1}^{K}\|\nabla u\|^2_{L_2(\Omega_k)} + \frac{\sigma}{h}\sum_{(k,l)}\|[\![u]\!]\|^2_{L_2(I_{k,l})}.$$

Remark 10.3. *When using this method, one has really to choose σ large enough such that the bilinear form is coercive.*

While it is possible to just choose σ to be excessively large (over-penalization), this cannot be advised because all the estimates deteriorate if σ is chosen too large. From another viewpoint, over-penalization forces the solution to be smooth; this makes the method conforming. If the discretizations at the interfaces agree, this is fine. If they do not agree, we have the same locking phenomena which we would also have for conforming discretizations.

If one increases the spline degree p, the parameter σ has to be chosen to increase like p^2.

Moreover, we can show that

$$a_h(u, v_h) \lesssim \|u\|_{Q_h^+} \|v_h\|_{Q_h} \qquad \text{and} \tag{10.14}$$

for all $u \in H^2(\Omega)$ and $v_h \in V_h$ and with the choice

$$\|u\|_{Q_h^+}^2 := \|u\|_{Q_h}^2 + h^2 \sum_{k=1}^{K} |u|_{H^2(\Omega_k)}^2.$$

Using some standard inverse and trace estimates, we further obtain

$$\|u\|_{Q_h^+}^2 \leq \sigma \left(|u|_{H^1(\Omega)}^2 + h^2 |u|_{H^2(\Omega)}^2 \right) \tag{10.15}$$

for any $u \in H^2(\Omega)$.

Now, we can write down the discrete problem, which reads as follows. Find $u_h \in V_h$ such that

$$a_h(u_h, v_h) = (f, v_h)_{L_2(\Omega)} \qquad \text{for all} \qquad v_h \in V_h.$$

Using a basis, we can set up a linear system

$$A_h \, \underline{u}_h = \underline{f}_h.$$

Using (10.13) and Lax-Milgram, we obtain existence and uniqueness of a solution of this discrete problem. Using consistency (10.12), i.e., that the exact solution satisfies the variational problem, and using the estimates (10.13) and (10.14), we obtain using a result similar to Ceá's lemma

$$\|u - u_h\|_{Q_h} \lesssim \inf_{v_h \in V_h} \|u - v_h\|_{Q_h^+},$$

where u is the solution of the continuous problem and u_h is the solution of the discretized problem. Using (10.15) and a standard approximation error estimate, we further obtain

$$\|u - u_h\|_{Q_h}^2 \lesssim \sigma h^2 |u|_{H^2}^2,$$

i.e., error estimates of the desired kind.

10.4 Literature

The framework discussed in Section 10.2 is standard; the notation follows [34]. The Section 10.3 follows the ideas of [36, 37].

Bibliography

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

[2] L. B. da Veiga, A. Buffa, G. Sangalli, and R. Vázquez, *Mathematical analysis of variational isogeometric methods,* Acta Numerica, vol. 23, pp. 157 – 287, 2014.

[3] R. Hoppe, Finite element methods. `https://www.math.uh.edu/~rohop/spring_05/`, 2005.

[4] F. Auricchio, L. B. da Veiga, T. Hughes, A. Reali, and G. Sangalli, *Isogeometric collocation methods,* Mathematical Models and Methods in Applied Sciences, vol. 20, no. 11, pp. 2075 – 2107, 2010.

[5] L. L. Schumaker, Spline functions: basic theory. Cambridge University Press, 1981.

[6] Y. Bazilevs, L. B. da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli, *Isogeometric analysis: approximation, stability and error estimates for h-refined meshes,* Mathematical Models and Methods in Applied Sciences, vol. 16, no. 07, pp. 1031 – 1090, 2006.

[7] C. Schwab, *p*- and *hp*-Finite Element Methods: Theory and Applications in Solid and Fluid Mechanics. Numerical Mathematics and Scientific Computation, Oxford: Clarendon Press, 1998.

[8] L. B. da Veiga, A. Buffa, J. Rivas, and G. Sangalli, *Some estimates for h-p-k-refinement in isogeometric analysis,* Numerische Mathematik, vol. 118, no. 2, pp. 271 – 305, 2011.

[9] C. K. Chui, *An introduction to wavelets, wavelet analysis and its applications,* Academic Press, Boston, 1992.

[10] V. Pillwein, *Orthogonal polynomials and symbolic computation.* `https://www3.risc.jku.at/education/courses/ss2011/ops-sc/main.pdf`, 2012.

[11] S. Takacs and T. Takacs, *Approximation error estimates and inverse inequalities for B-splines of maximum smoothness,* Mathematical Models and Methods in Applied Sciences, vol. 26, no. 07, pp. 1411 – 1445, 2016.

[12] M. Floater and E. Sande, *Optimal spline spaces of higher degree for L_2 n-widths,* Journal of Approximation Theory, vol. 216, pp. 1 – 15, 2017.

[13] E. Sande, C. Manni, and H. Speleers, *Sharp error estimates for spline approximation: explicit constants, n-widths, and eigenfunction convergence,* 2018. [arXiv:1810.13418].

[14] C. Koutschan, M. Neumüler, and S. Radu, *Inverse inequality estimates with symbolic computation,* Advances in Applied Mathematics, vol. 80, pp. 1 – 23, 2016.

[15] P. Antolin, A. Buffa, F. Calabrò, M. Martinelli, and G. Sangalli, *Efficient matrix computation for tensor-product isogeometric analysis: The use of sum factorization,* Computer Methods in Applied Mechanics and Engineering, vol. 285, pp. 817 – 828, 2015.

[16] A. Bressan and S. Takacs, *Sum-factorization techniques in isogeometric analysis,* 2018. [arXiv:1809.05471].

[17] F. Calabrò, G. Sangalli, and M. Tani, *Fast formation of isogeometric Galerkin matrices by weighted quadrature,* Computer Methods in Applied Mechanics and Engineering, vol. 316, pp. 606 – 622, 2017.

[18] C. Hofreither, *A black-box low-rank approximation algorithm for fast matrix assembly in isogeometric analysis,* Computer Methods in Applied Mechanics and Engineering, vol. 333, pp. 311 – 330, 2018.

[19] J. M. Melenk, K. Gerdes, and C. Schwab, *Fully discret hp-finite elements: Fast quadrature,* Computer Methods in Applied Mechanics and Engineering, vol. 190, no. 32–33, pp. 4339 – 4364, 2001.

[20] A. Mantzaflaris, B. Jüttler, B. N. Khoromskij, and U. Langer, *Low rank tensor methods in Galerkin-based isogeometric analysis,* Computer Methods in Applied Mechanics and Engineering, vol. 316, pp. 1062 – 1085, 2017.

[21] H. Speleers, *Hierarchical spline spaces: quasi-interpolants and local approximation estimates,* Advances in Computational Mathematics, vol. 43, no. 2, pp. 235 – 255, 2017.

[22] H. Speleers and C. Manni, *Effortless quasi-interpolation in hierarchical spaces,* Numerische Mathematik, vol. 132, no. 1, pp. 155 – 184, 2016.

[23] C. Giannelli, B. Jüttler, and H. Speleers, *THB-splines: The truncated basis for hierarchical splines,* Computer Aided Geometric Design, vol. 29, no. 7, pp. 485 – 498, 2012.

[24] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri, *T-splines and T-NURCCs,* ACM transactions on graphics (TOG), vol. 22, no. 3, pp. 477 – 484, 2003.

[25] L. B. da Veiga, A. Buffa, D. Cho, and G. Sangalli, *Analysis-suitable T-splines are dual-compatible,* Computer Methods in Applied Mechanics and Engineering, vol. 249, pp. 42 – 51, 2012.

[26] L. B. da Veiga, A. Buffa, G. Sangalli, and R. Vázquez, *Analysis-suitable T-splines of arbitrary degree: definition, linear independence and approximation properties,* Mathematical Models and Methods in Applied Sciences, vol. 23, no. 11, pp. 1979 – 2003, 2013.

[27] G. H. Golub and C. F. V. Loan, Matrix Computations. Johns Hopkins University Press, fourth ed., 2012.

[28] I. Georgieva and C. Hofreither, *Greedy low-rank approximation in Tucker format of tensors and solutions of tensor linear systems,* NuMa Report 2018-02, Institute of Computational Mathematics, Johannes Kepler University Linz, Austria, 2018.

[29] G. Sangalli and M. Tani, *Isogeometric preconditioners based on fast solvers for the Sylvester equation,* SIAM Journal on Scientific Computing, vol. 38, no. 6, pp. A3644 – A3671, 2016.

[30] W. Hackbusch, Multi-Grid Methods and Applications. Springer, Berlin, 1985.

[31] K. Scherer and A. Y. Shadrin, *New Upper Bound for the B-Spline Basis Condition Number: II. A Proof of de Boor's* $2k$*-Conjecture,* Journal of Approximation Theory, vol. 99, no. 2, pp. 217 – 229, 1999.

[32] C. Hofreither and S. Takacs, *Robust multigrid for isogeometric analysis based on stable splittings of spline spaces,* SIAM Journal on Numerical Analysis, vol. 4, no. 55, pp. 2004 – 2024, 2017.

[33] C. Hofreither, S. Takacs, and W. Zulehner, *A robust multigrid method for isogeometric analysis in two dimensions using boundary correction,* Computer Methods in Applied Mechanics and Engineering, vol. 316, pp. 22 – 42, 2017.

[34] S. Takacs, *Robust approximation error estimates and multigrid solvers for isogeometric multi-patch discretizations,* Mathematical Models and Methods in Applied Sciences, vol. 28, no. 10, pp. 1899 – 1928, 2018.

[35] D. Arnold, F. Brezzi, B. Cockburn, and L. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems,* SIAM Journal on Numerical Analysis, vol. 39, no. 5, pp. 1749 – 1779, 2002.

[36] U. Langer, A. Mantzaflaris, S. Moore, and I. Toulopoulos, *Multipatch discontinuous Galerkin Isogeometric Analysis,* in Isogeometric Analysis and Applications 2014 (B. Jüttler and B. Simeon, eds.), pp. 1 – 32, Springer International Publishing, 2015.

[37] U. Langer and I. Toulopoulos, *Analysis of multipatch discontinuous galerkin iga approximations to elliptic boundary value problems,* Computing and Visualization in Science, vol. 17, no. 5, pp. 217 – 233, 2015.