Chapter 1
Isogeometric Analysis: mathematical and implementational aspects, with applications

DRAFT June 30, 2017

T. J. R. Hughes[1], G. Sangalli[2] and M. Tani[2]

[1]Institute for Computational Engineering and Sciences,
The University of Texas at Austin, Austin, TX 78712-0027

[2]Dipartimento di Matematica, Università di Pavia
and Istituto di Matematica Applicata e Tecnologie Informatiche "E. Magenes" (CNR),
via Ferrata 1, 27100, Pavia, Italy

Abstract Isogeometric analysis (IGA) is a recent and successful extension of the classical finite element analysis. IGA adopts smooth splines, NURBS and generalizations to approximate the problem unknowns, in order to simplifies the interaction with omputer aided geometric design (CAGD) where the same functions are used to parametrize the geometry of interest. Important features emerge from the use of smooth approximations of the unknown fields. IGA is a powerful high-order methodology for PDEs numerical solution, a careful implementation is adopted, which exploit its full potential. We present an overview of the mathematical properties of IGA, discuss computationally-efficient isogeometric algorithms and report some significant applications.

Keywords: Isogeometric analysis, splines, NURBS, multipatch, local refinement, spectrum and dispersion analysis, functional analysis, quadrature, preconditioners.

1.1 Introduction

Isogeometric Analysis (IGA) was proposed in the seminal paper [68], with a main motivation: to improve the interoperability between computer aided geometric design (CAGD) and the analysis, i.e., numerical simulation. In IGA, the functions that are used in CAGD geometry description (these are splines, NURBS, etcetera) are used also for the representation of the unknowns of Partial Differential Equations (PDEs) that model physical phenomena of interest.

In the last decade Isogeometric methods have been successfully used and tested on a variety of engineering problems. The use of splines and NURBS in the representation of unknown fields yields interesting features, when comparing to standard finite element methods. This is due to the spline smoothness which not only allows direct approximation of PDEs of order higher than two[1], but also boosts the accuracy per degree of freedom (comparing to standard C^0 finite elements) and the spectral accuracy[2], and moreover facilitates construction of spaces that can be used in schemes that

[1] IGA of Cahn-Hilliard 4^{th}-order model of phase separation is studied in [61], [62]; Kirchhoff-Love 4^{th}-order model of plates and shells in [76], [26], [77], [20]; IGA of crack propagation is studied in [114], with 4^{th}- and 6^{th}-order gradient-enhanced theories of damage [115]; 4^{th}-order phase-field fracture models are considered in [29] and [30], where higher-order convergence rates to sharp-interface limit solutions are numerically demonstrated.

[2] The effect of regularity on the spectral behavior of isogeometric discretizations has been studied in [48], [70], [69].

preserve specific fundamental properties of the PDE of interest (for example, smooth divergence-free isogeometric spaces, see [38], [36], [57] and [58]). Spline smoothness is the key ingredient of isogeometric collocation methods, see [7] and [93].

The mathematics of isogeometric methods is based on classical spline theory (see, e.g., [50], [96]), but also poses new challenges. The study of h-refinement of tensor-product isogeometric spaces addressed in [14] and [25]. The study of k-refinement, that is, the use of splines and NURBS of high order and smoothness (C^{p-1} continuity for p-degree splines) is partially developed in [21], [55], [106], [39]. Stability of mixed isogeometric methods with a saddle-point form is the aim of the works [32], [33], [56], [6], [36], [57], [58], [20], [31], [112].

Recent overview of IGA and its mathematical properties are [18] and ECM2 .

We present in the following sections an introduction of the construction of isogeometric scalar and vector spaces, their approximation and spectral properties, of the computationally-efficient algorithms that can be used to construct and solve isogeometric linear systems, and finally report (from the literature) some significant isogeometric analyses of benchmark applications,

1.2 Splines and NURBS: definition and properties

This section contains a quick introduction to B-splines and NURBS and their use in geometric modeling and CAGD. Reference books on this topic are [96] ,[50], [18], [43], [87], [88].

1.2.1 Univariate splines

Given two positive integers p and n, we say that $\Xi := \{\xi_1, \dots, \xi_{n+p+1}\}$ is a p-*open knot vector* if

$$\xi_1 = \ldots = \xi_{p+1} < \xi_{p+2} \le \ldots \le \xi_n < \xi_{n+1} = \ldots = \xi_{n+p+1},$$

where repeated knots are allowed, and $\xi_1 = 0$ and $\xi_{n+p+1} = 1$. The vector $Z = \{\zeta_1, \dots, \zeta_N\}$ contains the *breakpoints*, that is the knots without repetitions, where m_j is the multiplicity of the breakpoint ζ_j, such that

$$\Xi = \{\underbrace{\zeta_1, \dots, \zeta_1}_{m_1 \text{ times}}, \underbrace{\zeta_2, \dots, \zeta_2}_{m_2 \text{ times}}, \dots, \underbrace{\zeta_N, \dots, \zeta_N}_{m_N \text{ times}}\}, \tag{1.1}$$

with $\sum_{i=1}^{N} m_i = n+p+1$. We assume $m_j \le p+1$ for all internal knots. The points in Z form a mesh, and the local mesh size of the element $I_i = (\zeta_i, \zeta_{i+1})$ is denoted $h_i = \zeta_{i+1} - \zeta_i$, for $i = 1, \dots, N-1$.

B-spline functions of degree p are defined by the well-known Cox-DeBoor recursion:

$$\widehat{B}_{i,0}(\zeta) = \begin{cases} 1 \text{ if } \xi_i \le \zeta < \xi_{i+1}, \\ 0 \text{ otherwise,} \end{cases} \tag{1.2}$$

and

$$\widehat{B}_{i,p}(\zeta) = \frac{\zeta - \xi_i}{\xi_{i+p} - \xi_i} \widehat{B}_{i,p-1}(\zeta) + \frac{\xi_{i+p+1} - \zeta}{\xi_{i+p+1} - \xi_{i+1}} \widehat{B}_{i+1,p-1}(\zeta), \tag{1.3}$$

where $0/0 = 0$. This gives a set of n B-splines that are non-negative, form a partition of unity, have local support, are linear independent.

The $\{\widehat{B}_{i,p}\}$ form a basis for the space of univariate *splines*, that is, piecewise polynomials of degree p with $k_j := p - m_j$ continuous derivatives at the points ζ_j, for $j = 1, \dots, N$:

$$S^p(\Xi) = \operatorname{span}\{\widehat{B}_{i,p},\ i = 1, \dots, n\}. \tag{1.4}$$

Remark 1.2.1. The notation S^p_r will be adopted to refer to the space $S^p(\Xi)$ when the multiplicity m_j of all internal knots is $p-r$. Then, S^p_r is a spline space with continuity C^r.

The maximum multiplicity allowed, $m_j = p+1$, gives $k_j = -1$, which represents a discontinuity at ζ_j. The *regularity vector* $\mathbf{k} = \{k_1, \dots, k_N\}$ will collect the regularity of the basis functions at the internal knots, with $k_1 = k_N = -1$ for the boundary knots. An example of B-splines is given in Figure 1.1. B-splines are interpolatory at knots ζ_j if and only if the multiplicity $m_j \geq p$, that is where the B-spline is at most C^0.

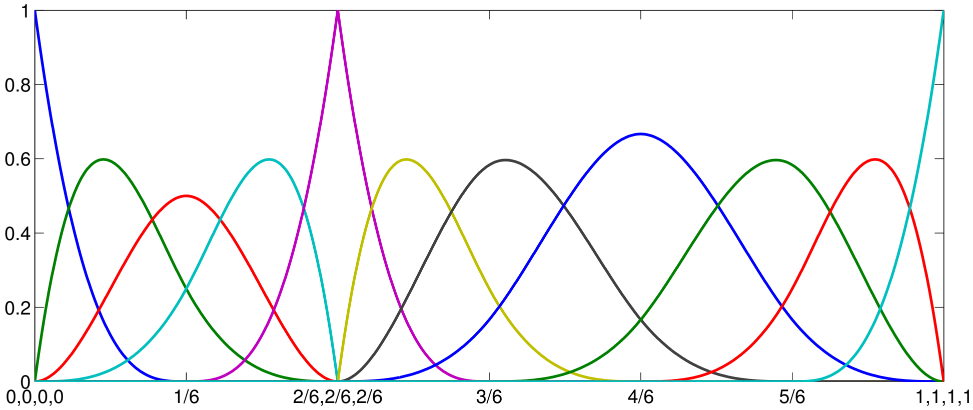


Fig. 1.1 Cubic B-splines and the corresponding knot vector with repetitions.

Each B-spline $\widehat{B}_{i,p}$ depends only on $p+2$ knots, which are collected in the *local knot vector*

$$\Xi_{i,p} := \{\xi_i, \dots, \xi_{i+p+1}\}.$$

When needed, we will stress this fact by adopting the notation

$$\widehat{B}_{i,p}(\zeta) = \widehat{B}[\Xi_{i,p}](\zeta). \tag{1.5}$$

The support of each basis function is exactly $\text{supp}(\widehat{B}_{i,p}) = [\xi_i, \xi_{i+p+1}]$.

A *spline curve* in $\mathbb{R}^d$, $d = 2, 3$ is a curve parametrized by a linear combination of B-splines and control points as follows:

$$\mathbf{C}(\zeta) = \sum_{i=1}^{n} \mathbf{c}_i \, \widehat{B}_{i,p}(\zeta) \qquad \mathbf{c}_i \in \mathbb{R}^d, \tag{1.6}$$

where $\{\mathbf{c}_i\}_{i=1}^n$ are called control points. Given a spline curve $\mathbf{C}(\zeta)$, its control polygon $\mathbf{C}_P(\zeta)$ is the piecewise linear interpolant of the control points $\{\mathbf{c}_i\}_{i=1}^n$ (see Figure 1.2).

In general, conic sections cannot be parametrized by polynomials but can be parametrized with rational polynomials, see [87, Sect. 1.4]. This has motivated the introduction of *Non-Uniform Rational B-Splines* (NURBS). In order to define NURBS, we set the *weight function* $W(\zeta) = \sum_{\ell=1}^n w_\ell \widehat{B}_{\ell,p}(\zeta)$ where the positive coefficients $w_\ell > 0$ for $\ell = 1, \dots, n$ are called *weights*. We define the NURBS basis functions as

$$\widehat{N}_{i,p}(\zeta) = \frac{w_i \widehat{B}_{i,p}(\zeta)}{\sum_{\ell=1}^n w_\ell \widehat{B}_{\ell,p}(\zeta)} = \frac{w_i \widehat{B}_{i,p}(\zeta)}{W(\zeta)}, \qquad i = 1, \dots, n, \tag{1.7}$$

which are rational B-splines. NURBS (1.7) inherit the main properties of B-splines mentioned above, that is they are are non-negative, form a partition of unity, and have local support. We denote the NURBS space they span by

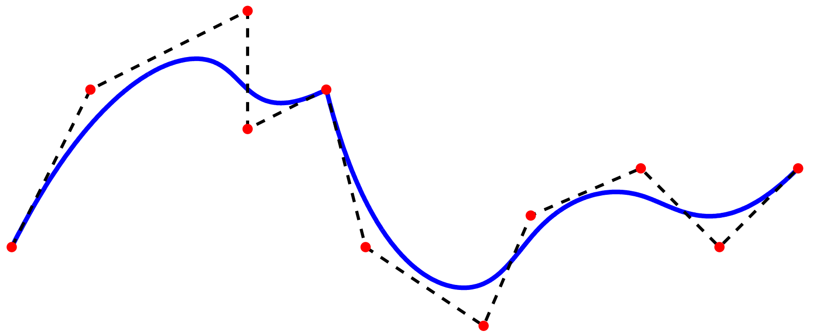


Fig. 1.2 Spline curve (solid line), control polygon (dashed line) and control points (red dots).

$$N^p(\Xi, W) = \mathrm{span}\{\widehat{N}_{i,p},\, i = 1, \ldots, n\}. \tag{1.8}$$

Similarly to splines, a NURBS curve is defined by associating one control point to each basis function, in the form:

$$\mathbf{C}(\zeta) = \sum_{i=1}^{n} \mathbf{c}_i\, \widehat{N}_{i,p}(\zeta) \qquad \mathbf{c}_i \in \mathbb{R}^d. \tag{1.9}$$

Actually, the NURBS curve is a projection into $\mathbb{R}^d$ of a non-rational B-spline curve in the space $\mathbb{R}^{d+1}$, which is defined by

$$\mathbf{C}^w(\zeta) = \sum_{i=1}^{n} \mathbf{c}_i^w\, \widehat{B}_{i,p}(\zeta),$$

where $\mathbf{c}_i^w = [\mathbf{c}_i, w_i] \in \mathbb{R}^{d+1}$ (see Figure 1.3).

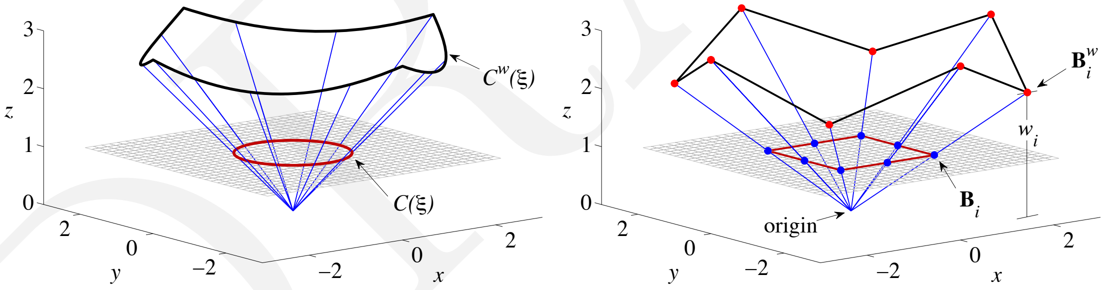


Fig. 1.3 On the left, the NURBS function $\xi \mapsto C(\xi)$ parametrizes the red circumference of a circle, given as the projection of the non-rational black spline curve, parametrized by the spline $\xi \mapsto C^w(\xi)$. The NURBS and spline control points are denoted $\mathbf{B}_i$ and $\mathbf{B}_i^w$, respectively, in the right plot.

For splines and NURBS curves, refinement is performed by *knot insertion* and *degree elevation*. In IGA, these two algorithms generate two kinds of refinement (see [68]): *h-refinement* which corresponds to mesh refinement and is obtained by insertion of new knots, and *p-refinement* which corresponds to degree elevation while maintaining interelement regularity, that is, by increasing the multiplicity of all knots. Furthermore, in IGA literature *k-refinement* denotes degree elevation, with increasing interelement regularity. This is not refinement in the sense of nested spaces, since the sequence of spaces generated by k-refinement has increasing global smoothness.

Having defined the spline space $S^p(\Xi)$, the next step is to introduce suitable projectors onto it. We focus on so called *quasi-interpolants* A common way to define them is by giving a dual basis, i.e,

$$\mathit{\Pi}_{p,\Xi} : C^\infty([0,1]) \to S^p(\Xi), \qquad \mathit{\Pi}_{p,\Xi}(f) = \sum_{j=1}^{n} \lambda_{j,p}(f)\widehat{B}_{j,p}, \tag{1.10}$$

where $\lambda_{j,p}$ are a set of dual functionals satisfying

$$\lambda_{j,p}(\widehat{B}_{k,p}) = \delta_{jk}, \tag{1.11}$$

δ_{jk} being the Kronecker symbol. The quasi-interpolant $\mathit{\Pi}_{p,\Xi}$ preserves splines, that is

$$\mathit{\Pi}_{p,\Xi}(f) = f, \quad \forall f \in S^p(\Xi). \tag{1.12}$$

From now on we assume local quasi-uniformity of the knot vector $\zeta_1, \zeta_2, \ldots, \zeta_N$, that is, there exists a constant $\theta \geq 1$ such that the mesh sizes $h_i = \zeta_{i+1} - \zeta_i$ satisfy the relation $\theta^{-1} \leq h_i/h_{i+1} \leq \theta$, for $i = 1, \ldots, N-2$. Among possible choices for the dual basis $\{\lambda_{j,p}\}$, $j = 1, \ldots, n$, a classical one is given in [96, Sect. 4.6], yielding to the following stability property (see [96], [14], [18]).

Proposition 1.2.2. *For any non-empty knot span $I_i = (\zeta_i, \zeta_{i+1})$,*

$$\|\mathit{\Pi}_{p,\Xi}(f)\|_{L^2(I_i)} \leq C\|f\|_{L^2(\widetilde{I}_i)}, \tag{1.13}$$

where the constant C depends only upon the degree p, and $\widetilde{I}_i$ is the support extension, i.e., the interior of the union of the supports of basis functions whose support intersects I_i

1.2.2 Multivariate splines and NURBS

Multivariate B-splines are defined from univariate B-splines by tensorization. Let d be the space dimensions (in practical cases, $d = 2, 3$). Assume $n_\ell \in \mathbb{N}$, the degree $p_\ell \in \mathbb{N}$ and the p_ℓ-open knot vector $\Xi_\ell = \{\xi_{\ell,1}, \ldots, \xi_{\ell,n_\ell+p_\ell+1}\}$ are given, for $\ell = 1, \ldots, d$. We define a polynomial degree vector $\mathbf{p} = (p_1, \ldots, p_d)$ and $\boldsymbol{\Xi} = \Xi_1 \times \ldots \times \Xi_d$. The corresponding knot values without repetitions are given for each direction ℓ by $Z_\ell = \{\zeta_{\ell,1}, \ldots, \zeta_{\ell,N_\ell}\}$. The knots Z_ℓ form a Cartesian grid in the *parametric domain* $\widehat{\Omega} = (0,1)^d$, giving the *Bézier mesh*, which is denoted by $\widehat{\mathcal{M}}$:

$$\widehat{\mathcal{M}} = \{Q_{\mathbf{j}} = I_{1,j_1} \times \ldots \times I_{d,j_d} \text{ such that } I_{\ell,j_\ell} = (\zeta_{\ell,j_\ell}, \zeta_{\ell,j_\ell+1}) \text{ for } 1 \leq j_\ell \leq N_\ell - 1\}. \tag{1.14}$$

For a generic Bézier element $Q_{\mathbf{j}} \in \widehat{\mathcal{M}}$, we also define its *support extension* $\widetilde{Q}_{\mathbf{j}} = \widetilde{I}_{1,j_1} \times \ldots \times \widetilde{I}_{d,j_d}$, with $\widetilde{I}_{\ell,j_\ell}$ the univariate support extension as defined in Proposition 1.2.2. We make use of the the set of multi-indices $\mathbf{I} = \{\mathbf{i} = (i_1, \ldots, i_d) : 1 \leq i_\ell \leq n_\ell\}$, and for each multi-index $\mathbf{i} = (i_1, \ldots, i_d)$, we define the local knot vector $\boldsymbol{\Xi}_{\mathbf{i},\mathbf{p}} = \Xi_{i_1,p_1} \times \ldots \times \Xi_{i_d,p_d}$. Then we introduce the set of multivariate B-splines

$$\left\{\widehat{B}_{\mathbf{i},\mathbf{p}}(\boldsymbol{\zeta}) = \widehat{B}[\Xi_{i_1,p_1}](\zeta_1) \ldots \widehat{B}[\Xi_{i_d,p_d}](\zeta_d), \, \forall \mathbf{i} \in \mathbf{I}\right\}. \tag{1.15}$$

The spline space in the parametric domain $\widehat{\Omega}$ is then

$$S^{\mathbf{p}}(\boldsymbol{\Xi}) = \operatorname{span}\{\widehat{B}_{\mathbf{i},\mathbf{p}}(\boldsymbol{\zeta}), \ \mathbf{i} \in \mathbf{I}\}, \tag{1.16}$$

which is the space of piecewise polynomials of degree $\mathbf{p}$ with the regularity across Bézier elements given by the knot multiplicities.

Multivariate NURBS are defined as rational tensor product B-splines. Given a set of *weights* $\{w_{\mathbf{i}}, \, \mathbf{i} \in \mathbf{I}\}$, and the weight function $W(\boldsymbol{\zeta}) = \sum_{\mathbf{j} \in \mathbf{I}} w_{\mathbf{j}} \widehat{B}_{\mathbf{j},\mathbf{p}}(\boldsymbol{\zeta})$, we define the NURBS basis functions

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

The NURBS space in the parametric domain $\widehat{\Omega}$ is then

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

As in the case of NURBS curves, the choice of the weights depends on the geometry to parametrize, and in IGA applications it is preserved by refinement.

Tensor-product B-splines and NURBS (1.7) are non-negative, form a partition of unity and have local support. As for curves, we define spline (NURBS, respectively) parametrizations of multivariate geometries in $\mathbb{R}^m$, $m = 2, 3$. A spline (NURBS, respectively) parametrization is then any linear combination of B-spline (NURBS, respectively) basis functions via control points $\mathbf{c}_{\mathbf{i}} \in \mathbb{R}^m$

$$\mathbf{F}(\boldsymbol{\zeta}) = \sum_{\mathbf{i}\in\mathbf{I}} \mathbf{c}_{\mathbf{i}}\widehat{B}_{\mathbf{i},\mathbf{p}}(\boldsymbol{\zeta}), \qquad \text{with } \boldsymbol{\zeta}\in\widehat{\Omega}. \tag{1.17}$$

Depending on the values of d and m, the map (1.17) can define a planar surface in $\mathbb{R}^2$ $(d = 2, m = 2)$, a manifold in $\mathbb{R}^3$ $(d = 2, m = 3)$, or a volume in $\mathbb{R}^3$ $(d = 3, m = 3)$.

The definition of the control polygon is generalized for multivariate splines and NURBS to a control mesh, the mesh connecting the control points $\mathbf{c}_{\mathbf{i}}$. Since B-splines and NURBS are not interpolatory, the control mesh is not a mesh on the domain Ω. Instead, the image of the Bézier mesh in the parametric domain through $\mathbf{F}$ gives the physical Bézier mesh in Ω, simply denoted the Bézier mesh if there is no risk of confusion (see Figure 1.4).

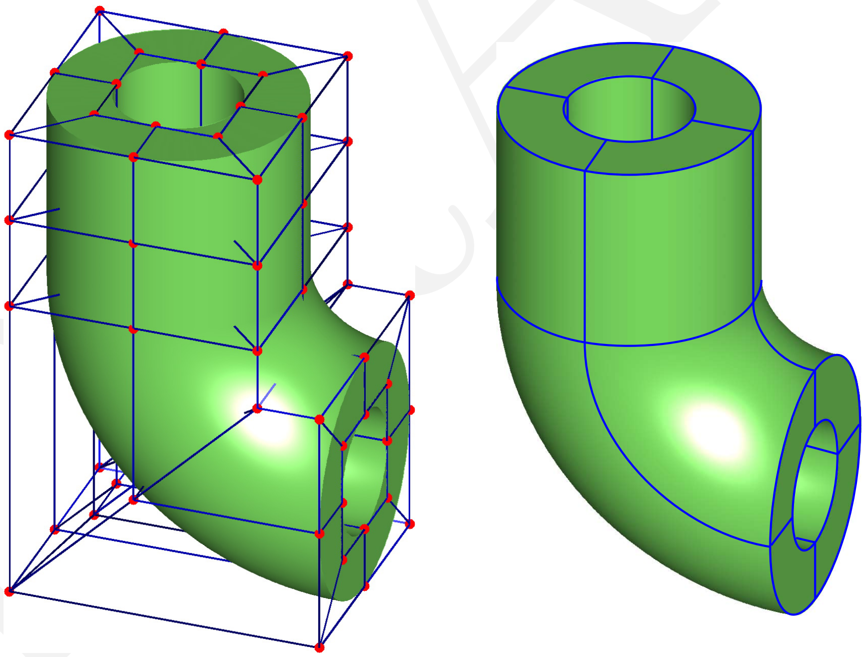


Fig. 1.4 The control mesh (left) and the physical Bézier mesh (right) for a pipe elbow is represented.

The interpolation and quasi-interpolation projectors can be also extended to the multi-dimensional case by a tensor product construction. Let, for $i = 1, \dots, d$, the notation $\mathit{\Pi}^i_{p_i}$ denote the univariate projector $\mathit{\Pi}_{p,\Xi}$ onto the space $S^{p_i}(\Xi_i)$, then define

$$\Pi_{\mathbf{p}}(f) = (\Pi^1_{p_1} \otimes \ldots \otimes \Pi^d_{p_d})(f). \tag{1.18}$$

Analogously, the multivariate quasi-interpolant is also defined from a dual basis (see [50, Chapter XVII]). Indeed, we have

$$\Pi_{\mathbf{p},\boldsymbol{\Xi}}(f) = \sum_{\mathbf{i}\in\mathbf{I}} \lambda_{\mathbf{i},\mathbf{p}}(f)\widehat{B}_{\mathbf{i},\mathbf{p}},$$

where each dual functional is defined from the univariate dual bases as $\lambda_{\mathbf{i},\mathbf{p}} = \lambda_{i_1,p_1} \otimes \ldots \otimes \lambda_{i_d,p_d}$.

1.2.3 Splines spaces with local tensor-product structure

A well developed research area concerns extensions of splines spaces beyond the tensor product structure, and allow local mesh refinement: for example T-splines, Locally-refinable (LR) splines, and hierarchical splines. T-splines have been proposed in [103] and have been adopted for isogeometric methods since [15]. They have been applied to shell problems [67], fluid-structure interaction problems [16] and contact mechanics simulation [51]. The algorithm for local refinement has evolved since its introduction in [102] (see, e.g., [100]), in order to overcome some initial limitations (see, e.g.,[54]). Other possibilities are LR-splines [52] and hierarchical splines [116].

We summarize here the definition of a T-spline and its main properties, following [18]. A T-mesh is a mesh that allows T-junctions. See Figure 1.5 (left) for an example. A T-spline set

$$\left\{\widehat{B}_{\mathbf{A},\mathbf{p}},\ \mathbf{A} \in \mathcal{A}\right\}, \tag{1.19}$$

is a generalization of the tensor-product set of multivariate splines (1.15). Indeed the functions in (1.19) have the structure

$$\widehat{B}_{\mathbf{A},\mathbf{p}}(\boldsymbol{\zeta}) = \widehat{B}[\Xi_{\mathbf{A},1,p_1}](\zeta_1)\ldots\widehat{B}[\Xi_{\mathbf{A},d,p_d}](\zeta_d) \tag{1.20}$$

where the set of indices, usually referred to as *anchors*, $\mathcal{A}$ and the associated local knot vectors $\Xi_{\mathbf{A},\ell,p_\ell}$, for all $\mathbf{A} \in \mathcal{A}$ are obtained from the T-mesh. If the polynomial degree is odd (in all directions) the anchors are associated with the vertices of the T-mesh, if the the polynomial degree is even (in all directions) the anchors are associated with the elements. Different polynomial degrees in different directions are possible. The local knot vectors are obtained from the anchors by moving along one direction and recording the knots corresponding to the intersections with the mesh. See the example in Figure 1.6.

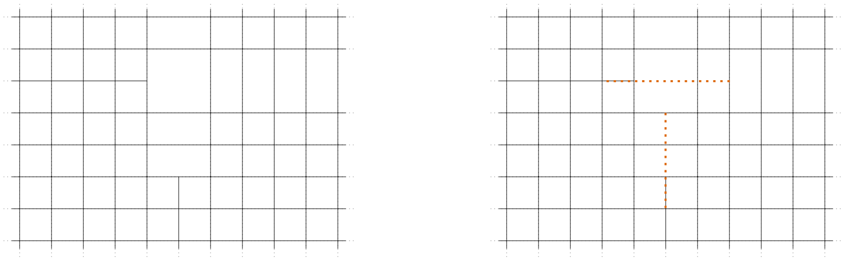


Fig. 1.5 A T-mesh with two T-junctions (on the left) and the same T-mesh with the T-junction extensions (on the right). The degree for this example is cubic and the T-mesh is analysis suitable since the extensions, one horizontal and the other vertical, do not intersect.

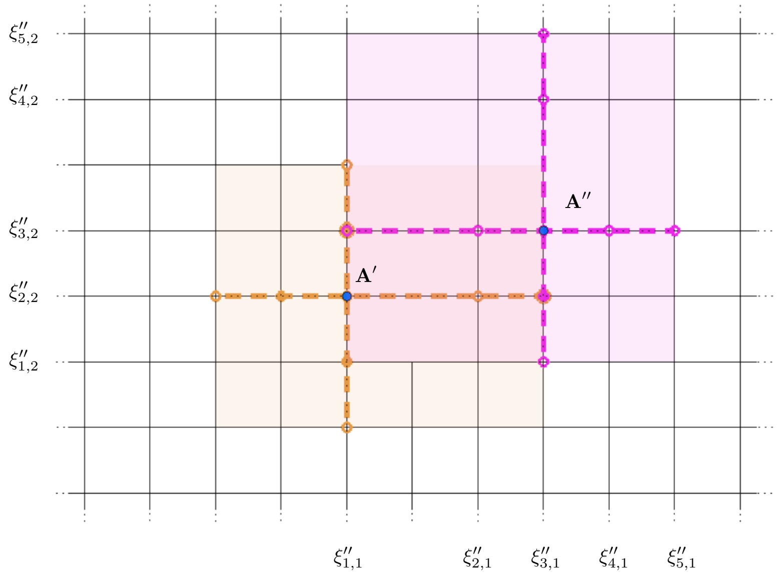


Fig. 1.6 Two bi-cubic T-spline anchors $\mathbf{A}'$ and $\mathbf{A}''$ and related local knot vectors. In particular, the local knot vectors for $\mathbf{A}''$ are $\Xi_{\mathbf{A}'',d,3} = \{\xi''_{1,d}, \xi''_{2,d}, \xi''_{3,d}, \xi''_{4,d}, \xi''_{5,d}, \}$, $d = 1, 2$. In this example, the two T-splines $\widehat{B}_{\mathbf{A}',\mathbf{p}}$ and $\widehat{B}_{\mathbf{A}'',\mathbf{p}}$ partially overlap (the overlapping holds in the horizontal direction).

On the parametric domain $\widehat{\Omega}$ we can define a Bézier mesh $\widehat{\mathcal{M}}$ as the collection of the the maximal open sets $Q \subset \widehat{\Omega}$ where the T-splines of (1.19) are polynomials in Q. We remark that the Bézier mesh and the T-mesh are different meshes.

The theory of T-splines focuses on the notion of Analysis-Suitable (AS) T-splines or, equivalently, Dual-Compatible (DC) T-splines: these are a subset of T-splines for which fundamental mathematical properties hold, of crucial importance for IGA.

We say that the two p-degree local knot vectors $\Xi' = \{\xi'_1, \dots \xi'_{p+2}\}$ and $\Xi'' = \{\xi''_1, \dots \xi''_{p+2}\}$ *overlap* if they are sub-vectors of consecutive knots taken from the same knot vector. For example $\{\xi_1, \xi_2, \xi_3, \xi_5, \xi_7\}$ and $\{\xi_3, \xi_5, \xi_7, \xi_8, \xi_9\}$ overlap, while $\{\xi_1, \xi_2, \xi_3, \xi_5, \xi_7\}$ and $\{\xi_3, \xi_4, \xi_5, \xi_6, \xi_8\}$ do not overlap. Then we say that two T-splines $\widehat{B}_{\mathbf{A}',\mathbf{p}}$ and $\widehat{B}_{\mathbf{A}'',\mathbf{p}}$ in (1.19) partially overlap if, when $\mathbf{A}' \neq \mathbf{A}''$, there exists a direction ℓ such that the local knot vectors $\Xi_{\mathbf{A}',\ell,p_\ell}$ and $\Xi_{\mathbf{A}'',\ell,p_\ell}$ are different and overlap. This is the case of Figure 1.6. Finally, the set (1.19) is a Dual-Compatible (DC) set of T-splines if each pair of T-splines in it partially overlaps. Its span

$$S^{\mathbf{p}}(\mathcal{A}) = \operatorname{span}\left\{\widehat{B}_{\mathbf{A},\mathbf{p}},\, \mathbf{A} \in \mathcal{A}\right\}, \qquad (1.21)$$

is denoted a Dual-Compatible (DC) T-spline space. The definition of a DC set of T-splines simplifies in two dimension ([22]): when $d = 2$, a T-spline space is a DC set of splines if and only if each pair of T-splines in it have overlapping local knot vector in at least one direction.

A full tensor-product space (see Section 1.2.2) is a particular case of a DC spline space. In general, partial overlap is sufficient for the construction of a dual basis, as in the full tensor-product case. We only need, indeed, a univariate dual basis (e.g., the one in [96]), and denote by $\lambda[\Xi_{\mathbf{A},\ell,p_\ell}]$ the univariate functional as in (1.11), depending on the local knot vector $\Xi_{\mathbf{A},\ell,p_\ell}$ and dual to each univariate B-spline with overlapping knot vector.

Proposition 1.2.3. *Assume that* (1.19) *is a DC set, and consider an associated set of functionals*

$$\{\lambda_{\mathbf{A},\mathbf{p}},\, \mathbf{A} \in \mathcal{A}\}, \qquad (1.22)$$

$$\lambda_{\mathbf{A},\mathbf{p}} = \lambda[\Xi_{\mathbf{A},1,p_1}] \otimes \ldots \otimes \lambda[\Xi_{\mathbf{A},d,p_d}]. \qquad (1.23)$$

Then (1.22) *is a dual basis for* (1.19).

Above, we assume that the local knot vectors in (1.23) are the same as in (1.19), (1.20). The proof of Proposition 1.2.3 can be found in [18]. The existence of dual functionals implies important properties for a DC set (1.19) and the related space $S^{\mathbf{p}}(\mathcal{A})$ in (1.21), as stated in the following theorem.

Theorem 1.2.4. *The T-splines in a DC set* (1.19) *are linearly independent. If the constant function belongs to $S^{\mathbf{p}}(\mathcal{A})$, they form a partition of unity. If the space of global polynomials of multi-degree* $\mathbf{p}$ *is contained in $S^{\mathbf{p}}(\mathcal{A})$, then the DC T-splines are locally linearly independent, that is, given $Q \in \widehat{\mathcal{M}}$, then the non-null T-splines restricted to the element Q are linearly independent.*

An important consequence of Proposition 1.2.3 is that we can build a projection operator $\Pi_{\mathbf{p}} : L^2(\widehat{\Omega}) \to S^{\mathbf{p}}(\mathcal{A})$ by

$$\Pi_{\mathbf{p}}(f)(\boldsymbol{\zeta}) = \sum_{\mathbf{A}\in\mathcal{A}} \lambda_{\mathbf{A},\mathbf{p}}(f)\widehat{B}_{\mathbf{A},\mathbf{p}}(\boldsymbol{\zeta}) \qquad \forall f \in L^2(\widehat{\Omega}),\ \forall \boldsymbol{\zeta} \in \widehat{\Omega}. \qquad (1.24)$$

This is the main tool (as mentioned in Section 1.4.1) to prove optimal approximation properties of T-splines.

In general, for DC T-splines, and in particular for tensor-product B-splines, we can define so called Greville abscissae. Each Greville abscissa

$$\boldsymbol{\gamma}_{\mathbf{A}} = (\gamma[\Xi_{\mathbf{A},1,p_1}], \ldots, \gamma[\Xi_{\mathbf{A},d,p_d}])$$

is a point in the parametric domain $\widehat{\Omega}$ and its d-component $\gamma[\Xi_{\mathbf{A},\ell,p_\ell}]$is the average of the p_ℓ internal knots of $\Xi_{\mathbf{A},\ell,p_\ell}$. They are the coefficients of the identity function in the T-spline expansion. Indeed, assuming that linear polynomials belong to the space $S^{\mathbf{p}}(\mathcal{A})$, we have that

$$\zeta_\ell = \sum_{\mathbf{A}\in\mathcal{A}} \gamma[\Xi_{\mathbf{A},\ell,p_\ell}]\widehat{B}_{\mathbf{A},\mathbf{p}}(\boldsymbol{\zeta}), \quad \forall\boldsymbol{\zeta}\in\widehat{\Omega}, \quad 1\le\ell\le d. \tag{1.25}$$

Greville abscissae are used as interpolation points (see [50]) and therefore for collocation based IGA ([7], [5], [93]).

A useful result, proved in [19, 22], is that a T-spline set is DC if and only if (under minor technical assumptions) it comes from a T-mesh that is Analysis-Suitable. The latter is a topological condition for the T-mesh [15] and it refers to dimension $d = 2$. A horizontal T-junction *extension* is a horizontal line that extend the T-mesh from a T-junctions of kind $\vdash$ and $\dashv$ in the direction of the missing edge for a length of $\lceil p_1/2\rceil$ elements, and in the opposite direction for $\lfloor p_1/2\rfloor$ elements; analogously a vertical T-junction *extension* is a vertical line that extend the T-mesh from a T-junctions of kind $\perp$ or $\top$ in the direction of the missing edge for a length of $\lceil p_2/2\rceil$ elements, and in the opposite direction for $\lfloor p_2/2\rfloor$ elements, see Figure 1.6 (right). Then, a T-mesh is *Analysis-Suitable* (AS) if horizontal T-junction extensions do not intersect vertical T-junction extensions.

1.2.4 Beyond tensor-product structure

Multivariate unstructured spline spaces are spanned by basis functions that are not, in general, tensor products. Non-tensor-product basis functions appear around so-called *extraordinary points*. Subdivision schemes, but also multipatch or T-splines spaces in the most general setting, are unstructured spaces. The construction and mathematical study of these spaces is important especially for IGA and is one of the most important recent research activities, see [92], [35]. We will further address this topic in Sections 1.3.3–1.4.3.

1.3 Isogeometric spaces

In this section, following [18], we give the definition of isogeometric spaces. We consider a single patch domain, i.e., the physical domain Ω is the image of the unit square, or the unit cube (the parametric domain $\widehat{\Omega}$) by a single NURBS parametrization. Then, for a given degree vector $\mathbf{p}^0$, knot vectors $\boldsymbol{\Xi}^0$ and a weight function $W \in S_{\mathbf{p}^0}(\boldsymbol{\Xi}^0)$, a map $\mathbf{F} \in (N^{\mathbf{p}^0}(\boldsymbol{\Xi}^0, W))^d$ is given such that $\Omega = \mathbf{F}(\widehat{\Omega})$, as in Figure 1.7.

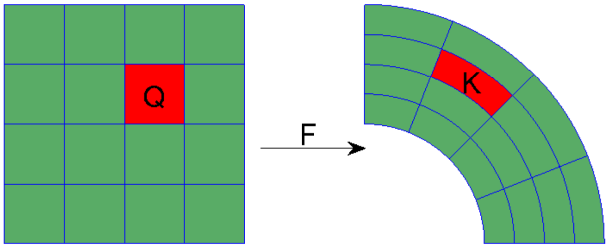


Fig. 1.7 Mesh $\widehat{\mathcal{M}}$ in the parametric domain, and its image $\mathcal{M}$ in the physical domain.

After having introduced the parametric Bézier mesh $\widehat{\mathcal{M}}$ in (1.14), as the mesh associated to the knot vectors $\boldsymbol{\Xi}$, we now define the physical Bézier mesh (or simply Bézier mesh) as the image of the elements in $\widehat{\mathcal{M}}$ through $\mathbf{F}$:

$$\mathcal{M} := \{K \subset \Omega : K = \mathbf{F}(Q), Q \in \widehat{\mathcal{M}}\}, \tag{1.26}$$

see Figure 1.7. The meshes for the coarsest knot vector $\boldsymbol{\Xi}^0$ will be denoted by $\widehat{\mathcal{M}}_0$ and $\mathcal{M}_0$. For any element $K = \mathbf{F}(Q) \in \mathcal{M}$, we define its support extension as $\widetilde{K} = \mathbf{F}(\widetilde{Q})$, with $\widetilde{Q}$ the support extension of Q. We denote the element size of any element $Q \in \widehat{\mathcal{M}}$ by $h_Q = \mathrm{diam}(Q)$, and the global mesh size by $h = \max\{h_Q : Q \in \widehat{\mathcal{M}}\}$. Analogously, we define the element sizes $h_K = \mathrm{diam}(K)$ and $h_{\widetilde{K}} = \mathrm{diam}(\widetilde{K})$.

For the sake of simplicity, we assume that the parametrization $\mathbf{F}$ is regular, that is, the inverse parametrization $\mathbf{F}^{-1}$ is well defined, and piecewise differentiable of any order with bounded derivatives. Assuming $\mathbf{F}$ is regular ensures that $h_Q \simeq h_K$. The case a of singular parametrization, that is, non-regular parametrization, will be discussed in Section 1.4.4.

1.3.1 Isoparametric spaces

Isogeometric spaces are constructed as push-forward through $\mathbf{F}$ of (refined) splines or NURBS spaces. In detail, let $\widehat{V}_h = N^{\mathbf{p}}(\boldsymbol{\Xi}, W)$ be a refinement of $N^{\mathbf{p}^0}(\boldsymbol{\Xi}^0, W)$, we define the scalar isogeometric space as:

$$V_h = \{f \circ \mathbf{F}^{-1} : f \in \widehat{V}_h\}. \tag{1.27}$$

Analogously,

$$V_h = \mathrm{span}\{N_{\mathbf{i},\mathbf{p}}(\mathbf{x}) := \widehat{N}_{\mathbf{i},\mathbf{p}} \circ \mathbf{F}^{-1}(\mathbf{x}),\ \mathbf{i} \in \mathbf{I}\}, \tag{1.28}$$

that is, the functions $N_{\mathbf{i},\mathbf{p}}$ form a basis of the space V_h. Isogeometric spaces with boundary conditions are defined straightforwardly.

Following [14], the construction of a projector on the NURBS isogeometric space V_h (defined in (1.27)) is based on a pull-back on the parametric domain, on a decomposition of the function into a numerator and weight denominator, and finally a spline projection of the numerator. We have $\Pi_{V_h} : \mathcal{V}(\Omega) \to V_h$defined as

$$\Pi_{V_h} f := \frac{\Pi_{\mathbf{p}}(W(f \circ \mathbf{F}))}{W} \circ \mathbf{F}^{-1}, \tag{1.29}$$

where $\Pi_{\mathbf{p}}$ is the spline projector (1.18) and $\mathcal{V}(\Omega)$ is a suitable function space. The approximation properties of Π_{V_h} will be discussed in Section 1.4.

The isogeometric vector space, as introduced in [68], is just $(V_h)^d$, that is a space of vector-valued functions whose components are in V_h. In parametric coordinates a spline isogeometric vector field of this kind reads

$$\mathbf{u}(\boldsymbol{\zeta}) = \sum_{\mathbf{i} \in \mathbf{I}} \mathbf{u}_{\mathbf{i}} \widehat{B}_{\mathbf{i},\mathbf{p}}(\boldsymbol{\zeta}), \qquad \text{with } \boldsymbol{\zeta} \in \widehat{\Omega}, \tag{1.30}$$

where $\mathbf{u}_{\mathbf{i}}$ are the degrees-of-freedom, also referred as *control variables* since they play the role of the control points of the geometry parametrization (1.17). This is an isoparametric construction.

1.3.2 De Rham compatible spaces

The following diagram

$$\begin{tikzcd} \mathbb{R} \arrow[r] & H^1(\Omega) \arrow[r, "\mathbf{grad}"] & \mathbf{H}(\mathbf{curl};\Omega) \arrow[r, "\mathbf{curl}"] & \mathbf{H}(\mathrm{div};\Omega) \arrow[r, "\mathrm{div}"] & L^2(\Omega) \arrow[r] & 0 \end{tikzcd} \tag{1.31}$$

is the De Rham cochain complex. The Sobolev spaces involved are the two standard scalar-valued, $H^1(\Omega)$ and $L^2(\Omega)$, and the two vector valued

$$\begin{aligned} \mathbf{H}(\mathbf{curl};\Omega) = &\quad \{\mathbf{u} \in L^2(\Omega)^3 \ : \ \mathbf{curl}\,\mathbf{u} \in L^2(\Omega)^3\} \\ \mathbf{H}(\mathrm{div};\Omega) = &\quad \{\mathbf{u} \in L^2(\Omega)^3 \ : \ \mathrm{div}\ \mathbf{u} \in L^2(\Omega)\}. \end{aligned}$$

Furthermore, as in general for complexes, the image of a differential operator in (1.31) is subset of the kernel of the next: for example, constants have null **grad**, gradients are **curl**-free fields, and so on. De Rham cochain complexes are related to the well-posedness of PDEs of key importance, for example in electromagnetic or fluid applications. This is why it is important to discretize (1.31) while preserving its structure. This is a well developed area of research for classical finite elements, called Finite Element Exterior Calculus (see the reviews [3, 4]) and likewise a successful development of IGA.

For the sake of simplicity, again, we restrict to a single patch domain and we do not include boundary conditions in the spaces. The dimension here is $d = 3$. The construction of isogeometric De Rham compatible spaces involves two stages.

The fist stage is the definition of spaces on the parametric domain $\widehat{\Omega}$. These are tensor-product spline spaces, as (1.16), with a specific choice for the degree and regularity in each direction. For that, we use the expanded notation $S^{p_1,p_2,p_3}(\Xi_1,\Xi_2,\Xi_3)$ for $S^{\mathbf{p}}(\boldsymbol{\Xi})$. Given degrees p_1, p_2, p_3 and knot vectors Ξ_1, Ξ_2, Ξ_3 we then define on $\widehat{\Omega}$ the spaces:

$$\begin{aligned} \widehat{X}_h^0 &= S^{p_1,p_2,p_3}(\Xi_1,\Xi_2,\Xi_3), \\ \widehat{X}_h^1 &= S^{p_1-1,p_2,p_3}(\Xi_1',\Xi_2,\Xi_3) \times S^{p_1,p_2-1,p_3}(\Xi_1,\Xi_2',\Xi_3) \times S^{p_1,p_2,p_3-1}(\Xi_1,\Xi_2,\Xi_3'), \\ \widehat{X}_h^2 &= S^{p_1,p_2-1,p_3-1}(\Xi_1,\Xi_2',\Xi_3') \times S^{p_1-1,p_2,p_3-1}(\Xi_1',\Xi_2,\Xi_3') \times S^{p_1-1,p_2-1,p_3}(\Xi_1',\Xi_2',\Xi_3), \\ \widehat{X}_h^3 &= S^{p_1-1,p_2-1,p_3-1}(\Xi_1',\Xi_2',\Xi_3'), \end{aligned} \tag{1.32}$$

where, given $\Xi_\ell = \{\xi_{\ell,1},\ldots,\xi_{\ell,n_\ell+p_\ell+1}\}$, Ξ_ℓ' is defined as the knot vector $\{\xi_{\ell,2},\ldots,\xi_{\ell,n_\ell+p_\ell}\}$, and we assume the knot multiplicities $1 \le m_{\ell,i} \le p_\ell$, for $i = 2,\ldots,N_\ell - 1$ and $\ell = 1,2,3$. With this choice, the functions in $\widehat{X}_h^0$ are at least continuous. Then, $\widehat{\mathbf{grad}}\,(\widehat{X}_h^0) \subset \widehat{X}_h^1$, and analogously, from the definition of the curl and the divergence operators we get $\widehat{\mathbf{curl}}(\widehat{X}_h^1) \subset \widehat{X}_h^2$, and $\widehat{\mathrm{div}}\,(\widehat{X}_h^2) \subset \widehat{X}_h^3$. This follows easily from the action of the derivative operator on tensor-product splines, for example:

$$\frac{\partial}{\partial \zeta_1} : S^{p_1,p_2,p_3}(\Xi_1,\Xi_2,\Xi_3) \to S^{p_1-1,p_2,p_3}(\Xi_1',\Xi_2,\Xi_3)$$

It is also proved in [38] that the kernel of each operator is exactly the image of the preceding one. In other words, these spaces form an exact sequence:

$$\begin{tikzcd} \mathbb{R} \arrow[r] & \widehat{X}_h^0 \arrow[r, "\widehat{\mathbf{grad}}"] & \widehat{X}_h^1 \arrow[r, "\widehat{\mathbf{curl}}"] & \widehat{X}_h^2 \arrow[r, "\widehat{\mathrm{div}}"] & \widehat{X}_h^3 \arrow[r] & 0. \end{tikzcd} \tag{1.33}$$

This is consistent with (1.31).

The second stage is the push forward of the isogeometric De Rham compatible spaces from the parametric domain $\widehat{\Omega}$ onto Ω. The classical isoparametric transformation on all spaces does not preserve the structure of the De Rham cochain complex. We need to use the transformations:

$$\begin{aligned} &\iota^0(f) := f \circ \mathbf{F}, && f \in H^1(\Omega)\,, \\ &\iota^1(\mathbf{f}) := (D\mathbf{F})^T(\mathbf{f} \circ \mathbf{F}), && \mathbf{f} \in \mathbf{H}(\mathbf{curl};\Omega)\,, \\ &\iota^2(\mathbf{f}) := \det(D\mathbf{F})(D\mathbf{F})^{-1}(\mathbf{f} \circ \mathbf{F}), && \mathbf{f} \in \mathbf{H}(\mathrm{div};\Omega)\,, \\ &\iota^3(f) := \det(D\mathbf{F})(f \circ \mathbf{F}), && f \in L^2(\Omega)\,, \end{aligned} \tag{1.34}$$

where $D\mathbf{F}$ is the Jacobian matrix of the mapping $\mathbf{F} : \widehat{\Omega} \to \Omega$. The transformation above preserve the structure of the De Rham cochain complex, in the sense of the following commuting diagram (see [65, Sect. 2.2] and [81, Sect. 3.9]):

$$\begin{tikzcd}
\mathbb{R} \arrow[r] & \widehat{X}_h^0 \arrow[r, "\widehat{\mathbf{grad}}"] & \widehat{X}_h^1 \arrow[r, "\widehat{\mathbf{curl}}"] & \widehat{X}_h^2 \arrow[r, "\widehat{\mathrm{div}}"] & \widehat{X}_h^3 \arrow[r] & 0 \\
\mathbb{R} \arrow[r] & X_h^0 \arrow[r, "\mathbf{grad}"] \arrow[u, "\iota^0"] & X_h^1 \arrow[r, "\mathbf{curl}"] \arrow[u, "\iota^1"] & X_h^2 \arrow[r, "\mathrm{div}"] \arrow[u, "\iota^2"] & X_h^3 \arrow[r] \arrow[u, "\iota^3"] & 0.
\end{tikzcd} \tag{1.35}$$

Note that the diagram above implicitly defines the isogeometric De Rham compatible spaces on Ω, that is X_h^0, X_h^1, X_h^2 and X_h^3; for example:

$$X_h^2 = \left\{ \mathbf{f} : \Omega \to \mathbb{R}^3 \text{ such that } \det(D\mathbf{F})(D\mathbf{F})^{-1}(\mathbf{f} \circ \mathbf{F}) \in \widehat{X}_h^2 \right\}. \tag{1.36}$$

In this setting, the geometry parametrization $\mathbf{F}$ can be either a spline in $(\widehat{X}_h^0)^3$ or a NURBS.

In fact, thanks to the smoothness of splines, isogeometric De Rham compatible spaces enjoy a wider applicability than their finite element counterpart. For example, assuming $m_{\ell,i} \leq p_\ell - 1$, for $i = 2, \ldots, N_\ell - 1$ and $\ell = 1, 2, 3$, then the space X_h^2 is subset of $(\,H^1(\Omega)\,)^3$. Furthermore there exists a subset $K_h \subset X_h^2$ of divergence-free isogeometric vector fields, i.e.,

$$K_h = \left\{ \mathbf{f} \in X_h^2 \text{ such that } \operatorname{div} \mathbf{f} = 0 \right\}, \tag{1.37}$$

that can be characterized as

$$\mathbf{f} \in K_h \quad \Leftrightarrow \quad \int_\Omega (\operatorname{div} \mathbf{f}) v = 0, \, \forall v \in X_h^3, \tag{1.38}$$

as well as

$$\mathbf{f} \in K_h \quad \Leftrightarrow \quad \exists \mathbf{v} \in X_h^1 \text{ such that } \mathbf{curl}\, \mathbf{v} = \mathbf{f}. \tag{1.39}$$

Both K_h and X_h^2 play an important role in the IGA of incompressible fluids, allowing exact point-wise divergence-free solutions that are difficult to achieve by finite element methods, or in linear small-deformation elasticity for incompressible materials, allowing point-wise preservation of the linearized volume under deformation. We refer to [36], [56], [57], [58] and the numerical tests of Section 1.8. We should also mention that for large deformation elasticity the volume preservation constraint becomes $\det \mathbf{f} = 1$, $\mathbf{f}$ denoting the deformation gradient, and the construction of isogeometric spaces that allow its exact preservation is an open and very challenging problem.

1.3.3 Extensions

Isogeometric spaces can be constructed from non-tensor-product or unstructured spline spaces, as the ones listed in Section 1.2.3.

Unstructured multipatch isogeometric spaces may have C^0 continuity at patch interfaces, of higher continuity. The implementation of C^0-continuity over multipatch domains is well understood (see e.g. [78], [23], [99] for strong and [34] for weak imposition of C^0 conditions). Some papers have tackled the problem of constructing isogeometric spaces of higher order smoothness, such as [35], [73], [92], [83], [46]. The difficulty is how to construct analysis-suitable unstructured isogeometric spaces with global C^1 or higher continuity. The main question concerns the approximation properties of these spaces, see Section 1.4.3.

An important operation, derived from CAGD, and applied to isogeometric spaces is trimming. Indeed trimming is very common in geometry representation, since it is the natural outcome of Boolean operations (union, intersection, subtraction of domains). One possibility is to approximate (up to some prescribed tolerance) the trimmed domain by an untrimmed multipatch or T-spline parametrized domain, see [101]. Another possibility is to use directly the trimmed geometry and deal with the two major difficulties that arise: efficient quadrature and imposition of boundary conditions, see [90], [95], [89].

1.4 Isogeometric spaces: approximation

1.4.1 h-refinement

The purpose of this section is to summarize the approximation properties of the isogeometric space V_h defined in (1.27). We focus on the convergence analysis under h-refinement, presenting results first obtained in [14] and [25]. To express the error bounds, we will make use of Sobolev spaces on a domain D, that can be either Ω or $\widehat{\Omega}$ or subsets such as as Q, $\widetilde{Q}$, K or $\widetilde{K}$. For example, $H^s(D)$, $s \in \mathbb{N}$ is the space of square integrable functions $f \in L^2(\Omega)$ such that its derivatives up to order s are square integrable. However, conventional Sobolev spaces are not enough. Indeed, since the mapping $\mathbf{F}$ is not arbitrarily regular across mesh lines, even if a scalar function f in physical space satisfies $f \in H^s(\Omega)$, its pull-back $\widehat{f} = f \circ \mathbf{F}$ is not in general in $H^s(\widehat{\Omega})$. As a consequence, the natural function space in parametric space, in order to study the approximation properties of mapped NURBS, is not the standard Sobolev space H^s but rather a "bent" version that allows for less regularity across mesh lines. In the following, as usual, C will denote a constant, possibly different at each occurrence, but independent of the mesh-size h. Note that, unless noted otherwise, C depends on the polynomial degree p and regularity.

Let $d = 1$ first. We recall that $I_i = (\zeta_i, \zeta_{i+1})$ are the intervals of the partition of $I = (0, 1)$ given by the knot vector. We define for any $q \in \mathbb{N}$ the piecewise polynomial space

$$\mathcal{P}_q(\Xi) = \{v \in L^2(I) \text{ such that } v|_{I_i} \text{ is a } q\text{-degree polynomial, } \forall i = 1, \ldots, N-1\}.$$

Given $s \in \mathbb{N}$ and any sub-interval $E \subset I$, we indicate by $H^s(E)$ the usual Sobolev space endowed with norm $\|\cdot\|_{H^s(E)}$ and semi-norm $|\cdot|_{H^s(E)}$. We define the *bent Sobolev space* (see [14]) on I as

$$\mathcal{H}^s(I) = \left\{ \begin{array}{l} f \in L^2(I) \;\text{ such that } f|_{I_i} \in H^s(I_i) \;\forall\, i = 1, \ldots, N-1, \text{ and} \\ D^k_- f(\zeta_i) = D^k_+ f(\zeta_i), \,\forall k = 0, \ldots, \min\{s-1, k_i\}, \forall i = 2, \ldots, N-1, \end{array} \right\} \tag{1.40}$$

where $D^k_\pm$ denote the k^{th}-order left and right derivative (or left and right limit for $k = 0$), and k_i is the number of continuous derivatives at the break point ζ_i. We endow the above space with the broken norm and semi-norms

$$\|f\|^2_{\mathcal{H}^s(I)} = \sum_{j=0}^{s} |f|^2_{\mathcal{H}^j(I)} \;, \quad |f|^2_{\mathcal{H}^j(I)} = \sum_{i=1}^{N-1} |f|^2_{H^j(I_i)} \quad \forall j = 0, 1, \ldots, s,$$

where $|\cdot|_{H^0(I_i)} = \|\cdot\|_{L^2(I_i)}$.

In higher dimensions, the tensor product bent Sobolev spaces are defined as follows. Let $\mathbf{s} = (s_1, s_2, \ldots, s_d)$ in $\mathbb{N}^d$. By a tensor product construction starting from (1.40), we define the tensor product bent Sobolev spaces in the parametric domain $\widehat{\Omega} := (0, 1)^d$

$$\mathcal{H}^{\mathbf{s}}(\widehat{\Omega}) := \mathcal{H}^{s_1}(0, 1) \otimes \mathcal{H}^{s_2}(0, 1) \otimes \ldots \otimes \mathcal{H}^{s_d}(0, 1),$$

endowed with the tensor-product norm and seminorms. The above definition clearly extends immediately to the case of any hyper-rectangle $E \subset \widehat{\Omega}$ that is a union of elements in $\widehat{\mathcal{M}}$.

We restrict, for simplicity of exposition, to the two-dimensional case. As in the one-dimensional case, we assume local quasi-uniformity of the mesh in each direction. Let $\Pi_{p_i, \Xi_i} : L^2(I) \to S^{p_i}(\Xi_i)$, for $i = 1, 2$, indicate the univariate quasi-interpolant associated to the knot vector Ξ_i and polynomial degree p_i. Let moreover $\Pi_{\mathbf{p},\boldsymbol{\Xi}} = \Pi_{p_1,\Xi_1} \otimes \Pi_{p_2,\Xi_2}$ from $L^2(\Omega)$ to $S^{\mathbf{p}}(\boldsymbol{\Xi})$ denote the tensor product quasi-interpolant built using the Π_{p_i,Ξ_i} defined in (1.18) for $d=2$. In what follows, given any sufficiently regular function $f : \widehat{\Omega} \to \mathbb{R}$, we will indicate the partial derivative operators with the symbol

$$\widehat{D}^{\mathbf{r}} f = \frac{\partial^{r_1} \partial^{r_2} f}{\partial \zeta_1^{r_1} \partial \zeta_2^{r_2}} \qquad \mathbf{r} = (r_1, r_2) \in \mathbb{N}^2. \tag{1.41}$$

Let $E \subset \widehat{\Omega}$ be any union of elements $Q \in \widehat{\mathcal{M}}$ of the spline mesh. We will adopt the notation

$$\|f\|^2_{L^2_h(E)} := \sum_{Q \in \widehat{\mathcal{M}} \text{ s.t. } Q \subset E} \|f\|^2_{L^2(Q)}.$$

The element size of a generic element $Q_{\mathbf{i}} = I_{1,i_1} \times \ldots \times I_{d,i_d} \in \widehat{\mathcal{M}}$ will be denoted by $h_{Q_{\mathbf{i}}} = \operatorname{diam}(Q_{\mathbf{i}})$. We will indicate the length of the edges of $Q_{\mathbf{i}}$ by $h_{1,i_1}, h_{2,i_2},$. Because of the local quasi-uniformity of the mesh in each direction, the length of the two edges of the extended patch $\widetilde{Q}_{\mathbf{i}}$ are bounded from above by h_{1,i_1} and h_{2,i_2}, up to a multiplicative factor. The quasi-uniformity constant is denoted θ. We have the following result (see [25], [18] for its proof), that can be established for spaces with boundary conditions as well.

Proposition 1.4.1. *Given integers $0 \le r_1 \le s_1 \le p_1 + 1$ and $0 \le r_2 \le s_2 \le p_2 + 1$, there exists a constant C depending only on $\mathbf{p}, \theta$ such that for all elements $Q_{\mathbf{i}} \in \widehat{\mathcal{M}}$,*

$$\|\widehat{D}^{(r_1,r_2)}(f - \Pi_{\mathbf{p},\boldsymbol{\Xi}} f)\|_{L^2(Q_{\mathbf{i}})} \le C\Big((h_{1,i_1})^{s_1-r_1} \|\widehat{D}^{(s_1,r_2)} f\|_{L^2_h(\widetilde{Q}_{\mathbf{i}})} + (h_{2,i_2})^{s_2-r_2} \|\widehat{D}^{(r_1,s_2)} f\|_{L^2_h(\widetilde{Q}_{\mathbf{i}})}\Big)$$

for all f in $\mathcal{H}^{(s_1,r_2)}(\widehat{\Omega}) \cap \mathcal{H}^{(r_1,s_2)}(\widehat{\Omega})$.

We can state the approximation estimate for the projection operator on the isogeometric space V_h, that is $\Pi_{V_h} : L^2(\Omega) \to V_h$, defined in (1.29). In the physical domain $\Omega = \mathbf{F}(\widehat{\Omega})$, we introduce the coordinate system naturally induced by the geometrical map $\mathbf{F}$, referred to as the $\mathbf{F}$-coordinate system, that associates to a point $\mathbf{x} \in \Omega$ the Cartesian coordinates in $\widehat{\Omega}$ of its counter-image $\mathbf{F}^{-1}(\mathbf{x})$. At each $\mathbf{x} \in K \in \mathcal{M}_0$ (more generally, at each $\mathbf{x}$ where $\mathbf{F}$ is differentiable) the tangent *base vectors* $\mathbf{g}_1$ and $\mathbf{g}_2$ of the $\mathbf{F}$-coordinate system can be defined as

$$\mathbf{g}_i = \mathbf{g}_i(\mathbf{x}) = \frac{\partial \mathbf{F}}{\partial \zeta_i}(\mathbf{F}^{-1}(\mathbf{x})), \quad i = 1, 2; \tag{1.42}$$

these are the images of the canonical base vectors $\widehat{\mathbf{e}}_i$ in $\widehat{\Omega}$, and represent the axis directions of the $\mathbf{F}$-coordinate system (see Figure 1.8).

Analogously to the derivatives in the parametric domain (1.41), the derivatives of $f : \Omega \to \mathbb{R}$ in Cartesian coordinates are denoted by

$$D^{\mathbf{r}} f = \frac{\partial^{r_1} \partial^{r_2} f}{\partial x_1^{r_1} \partial x_2^{r_2}} \qquad \mathbf{r} = (r_1, r_2) \in \mathbb{N}^2.$$

We also consider the derivatives of $f : \Omega \to \mathbb{R}$ with respect to the $\mathbf{F}$-coordinates. These are just the directional derivatives: for the first order we have

$$\frac{\partial f}{\partial \mathbf{g}_i}(\mathbf{x}) = \nabla f(\mathbf{x}) \cdot \mathbf{g}_i(\mathbf{x}) = \lim_{t \to 0} \frac{f(\mathbf{x} + t\mathbf{g}_i(\mathbf{x})) - f(\mathbf{x})}{t}, \tag{1.43}$$

which is well defined for any $\mathbf{x}$ in the (open) elements of the coarse triangulation $\mathcal{M}_0$, as already noted. Higher order derivatives are defined by recursion

$$\frac{\partial^{r_i} f}{\partial \mathbf{g}_i^{r_i}} = \frac{\partial}{\partial \mathbf{g}_i}\left(\frac{\partial^{r_i - 1} f}{\partial \mathbf{g}_i^{r_i - 1}}\right) = \left(\frac{\partial}{\partial \mathbf{g}_i}\left(\ldots\left(\frac{\partial}{\partial \mathbf{g}_i}\left(\frac{\partial f}{\partial \mathbf{g}_i}\right)\right)\right)\right);$$

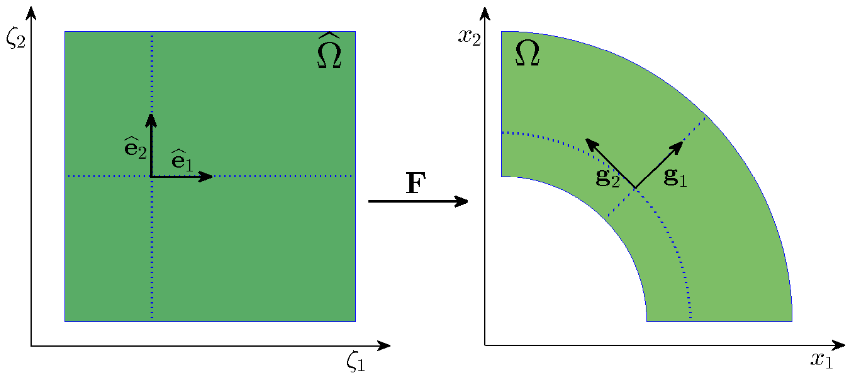


Fig. 1.8 Illustration of the **F**-coordinate system in the physical domain.

more generally, we adopt the notation

$$D^{\mathbf{r}}_{\mathbf{F}} f = \frac{\partial^{r_1}}{\partial \mathbf{g}_1^{r_1}} \frac{\partial^{r_2} f}{\partial \mathbf{g}_2^{r_2}} \qquad \mathbf{r} = (r_1, r_2) \in \mathbb{N}^2. \tag{1.44}$$

Derivatives with respect to the **F**-coordinates are directly related to derivatives in the parametric domain, by

$$D^{\mathbf{r}}_{\mathbf{F}} f = \left(\widehat{D}^{\mathbf{r}} \left(f \circ \mathbf{F} \right) \right) \circ \mathbf{F}^{-1}. \tag{1.45}$$

Let E be a union of elements $K \in \mathcal{M}$. We introduce the broken norms and seminorms

$$\|f\|^2_{\mathcal{H}^{(s_1,s_2)}_{\mathbf{F}}(E)} = \sum_{r_1=0}^{s_1} \sum_{r_2=0}^{s_2} |f|^2_{\mathcal{H}^{(r_1,r_2)}_{\mathbf{F}}(E)}, \qquad |f|^2_{\mathcal{H}^{(s_1,s_2)}_{\mathbf{F}}(E)} = \sum_{K \in \mathcal{M} \text{ s.t. } K \subset E} |f|^2_{H^{(s_1,s_2)}_{\mathbf{F}}(K)}, \tag{1.46}$$

where

$$|f|_{H^{(s_1,s_2)}_{\mathbf{F}}(K)} = \left\| D^{(s_1,s_2)}_{\mathbf{F}} f \right\|_{L^2(K)}.$$

We also introduce the following space

$$H^{(s_1,s_2)}_{\mathbf{F}}(\Omega) = \text{closure of } C^\infty(\Omega) \text{ with respect to the norm } \|\cdot\|_{\mathcal{H}^{(s_1,s_2)}_{\mathbf{F}}(\Omega)}.$$

The following theorem from [25] states the main estimate for the approximation error of $\Pi_{V_h} f$ and, making use of derivatives in the **F**-coordinate system, it is suitable for anisotropic meshes. For a generic element $K_{\mathbf{i}} = \mathbf{F}(Q_{\mathbf{i}}) \in \mathcal{M}$, the notation $\widetilde{K}_{\mathbf{i}} = \mathbf{F}(\widetilde{Q}_{\mathbf{i}})$ indicates its support extension.

Theorem 1.4.2. *Given integers r_i, s_i, such that $0 \leq r_i \leq s_i \leq p_i + 1$, $i = 1, 2$, there exists a constant C depending only on $\mathbf{p}, \theta, \mathbf{F}, W$ such that for all elements $K_{\mathbf{i}} = \mathbf{F}(Q_{\mathbf{i}}) \in \mathcal{M}$,*

$$|f - \Pi_{V_h} f|_{\mathcal{H}^{(r_1,r_2)}_{\mathbf{F}}(K_{\mathbf{i}})} \leq C \Big((h_{1,i_1})^{s_1 - r_1} \|f\|_{\mathcal{H}^{(s_1,r_2)}_{\mathbf{F}}(\widetilde{K}_{\mathbf{i}})} + (h_{2,i_2})^{s_2 - r_2} \|f\|_{\mathcal{H}^{(r_1,s_2)}_{\mathbf{F}}(\widetilde{K}_{\mathbf{i}})} \Big) \tag{1.47}$$

for all f in $H^{(s_1,r_2)}_{\mathbf{F}}(\Omega) \cap H^{(r_1,s_2)}_{\mathbf{F}}(\Omega)$.

We have the following corollary of Theorem 1.4.2, similar to [14, Theorem 3.1], or [18, Theorem 4.24] (the case with boundary conditions is handled similarly).

Corollary 1.4.3. *Given integers r, s, such that $0 \leq r \leq s \leq \min(p_1, \ldots, p_d) + 1$, there exists a constant C depending only on $\mathbf{p}, \theta, \mathbf{F}, W$ such that*

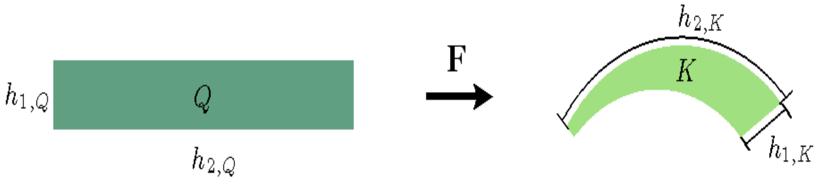


Fig. 1.9 Q is mapped by the geometrical map $\mathbf{F}$ to K.

$$\begin{aligned} &\|f - \Pi_{V_h} f\|_{H^r(K_\mathbf{i})} \leq C(h_{K_\mathbf{i}})^{s-r}\|f\|_{H^s(\widetilde{K}_\mathbf{i})} \quad \forall K_\mathbf{i} \in \mathcal{M}, \\ &\|f - \Pi_{V_h} f\|_{H^r(\Omega)} \leq C h^{s-r}\|f\|_{H^s(\Omega)}, \end{aligned} \tag{1.48}$$

for all f in $H^s(\Omega)$.

The error bound above straightforwardly covers isogeometric/isoparametric vector fields. The error theory is possible also for isogeometric De Rham compatible vector fields. In this framework there exists commuting projectors, i.e., projectors that make the diagram

$$\begin{tikzcd} \mathbb{R} \arrow[r] & H^1(\Omega) \arrow[r, "\mathbf{grad}"] \arrow[d, "\Pi^0"'] & \mathbf{H}(\mathbf{curl};\Omega) \arrow[r, "\mathbf{curl}"] \arrow[d, "\Pi^1"'] & \mathbf{H}(\mathrm{div};\Omega) \arrow[r, "\mathrm{div}"] \arrow[d, "\Pi^2"'] & L^2(\Omega) \arrow[r] \arrow[d, "\Pi^3"'] & 0 \\ \mathbb{R} \arrow[r] & X_h^0 \arrow[r, "\mathbf{grad}"] & X_h^1 \arrow[r, "\mathbf{curl}"] & X_h^2 \arrow[r, "\mathrm{div}"] & X_h^3 \arrow[r] & 0. \end{tikzcd} \tag{1.49}$$

commutative. These projectors not only are important for stating approximation estimates, but also play a fundamental role in the stability of isogeometric schemes; see [38], [3].

1.4.2 p-refinement and k-refinement

Approximation estimates in Sobolev norms have the general form

$$\inf_{f_h \in V_h} \|f - f_h\|_{H^r(\Omega)} \leq C(h, p, k; r, s)\|f\|_{H^s(\Omega)} \tag{1.50}$$

where the optimal constant is therefore

$$C(h, p, k; r, s) = \sup_{f \in B^s(\Omega)} \inf_{f_h \in V_h} \|f - f_h\|_{H^r(\Omega)} \tag{1.51}$$

where $B^s(\Omega) = \{f \in H^s(\Omega) \text{ such that } \|f\|_{H^s(\Omega)} \leq 1\}$ is the unit ball in $H^s(\Omega)$. The study in Section 1.4.1 covers the approximation under h-refinement, giving an asymptotic bound to (1.51) with respect to h which is sharp, for $s \leq p + 1$,

$$C(h, p, k; r, s) \approx C(p, k; r, s)h^{s-r}, \qquad \text{for } h \to 0. \tag{1.52}$$

This is the fundamental and most standard analysis, but it does not explain the benefits of k-refinement, a unique feature of IGA. High-degree, high-continuity splines and NURBS are superior to standard high-order finite elements when considering accuracy per degree-of-freedom. The study of k-refinement is still incomplete even though some important results are available in the literature. In particular, [21] contains h, p, k-explicit approximation bounds for spline spaces of degree $2q + 1$ and up to C^q global continuity, while the recent report [106] contains the error estimate

$$\inf_{f_h \in V_h} |f - f_h|_{H^r(\Omega)} \leq (\sqrt{2}h)^{q-r}|f|_{H^q(\Omega)}$$

for univariate C^{p-1}, p-degree splines, with $0 \leq r \leq q \leq p+1$ on uniform knot vectors.

An innovative approach, and alternative to standard error analysis, is developed in [55]. There a theoretical/numerical investigation provides clear evidence of the importance of k-refinement. The space of smooth splines is shown to be very close to a best approximation space in the Sobolev metric. The approach is as follows: given the isogeometric space V_h, with $N = \mathrm{dim} V_h$ together with (1.51), we consider the *Kolmogorov N-width*:

$$d_N(B^s(\Omega), H^r(\Omega)) = \inf_{\substack{W_h \subset H^s(\Omega) \\ \mathrm{dim} W_h = N}} \sup_{f \in B^s(\Omega)} \inf_{f_h \in W_h} \|f - f_h\|_{H^r(\Omega)}. \tag{1.53}$$

Then the optimality ratio is defined as

$$\Lambda(B^s(\Omega), V_h, H^r(\Omega)) = \frac{C(h, p, k; r, s)}{d_N(B^s(\Omega), H^r(\Omega))}. \tag{1.54}$$

In general, the quantity $\Lambda(B^s(\Omega), V_h, H^r(\Omega))$ is hard to compute analytically but can be accurately approximated numerically, by solving suitable generalized eigenvalue problems (see [55]). In Figure 1.10 we compare smooth C^3 quartic splines and standard quartic finite elements (that is, C^0 splines) under h-refinement. An interesting result is that smooth splines asymptotically achieve optimal approximation in the context considered, that is, they tend to be an optimal approximation space given the number of degrees-of-freedom, since $\Lambda(B^5(0,1), S^4_3, L^2(0,1)) \to 1$. This is not surprising as it is known that uniform periodic spline spaces are optimal in the periodic setting. On the contrary, C^0 finite elements are far from optimal. In Figure 1.11 we plot the optimality ratios for the L^2 error for different Sobolev regularity s and for smooth splines with different degrees p. There is numerical evidence that $\Lambda(B^s(0,1), S^p_{p-1}, L^2(0,1))$ is bounded and close to 1 for all $p \geq s-1$. It is a surprising result, but in fact confirms that high-degree smooth splines are accurate even when the solution to be approximated has low Sobolev regularity (see [55] for further considerations).

This issue has been further studied in [39], for the special case of solutions that are piecewise analytic with a localized singularity, which is typical of elliptic PDEs on domains with corners or sharp edges. See Section **??** for an example in two-dimensions. The work [39] focus instead on the simplified one-dimensional problem, and consider a model singular solution $f(\zeta) = \zeta^\alpha - \zeta$ on the interval $[0,1]$, with $0 < \alpha < 1$. From the theory of hp-FEMs (i.e., hp finite elements; see [97]) it is known that exponential convergence is achieved, precisely

$$|f - f_h|_{H^1(0,1)} \leq C e^{-b\sqrt{N}} \tag{1.55}$$

where C and b are positive constants, N is the total number of degrees-of-freedom, and f_h is a suitable finite element approximation of f. The bound (1.55) holds if the mesh is geometrically refined towards the singularity point $\zeta = 0$ and with a suitable selection of the polynomial degree, growing from left (the singularity) to the right of the interval $[0,1]$. The seminal paper [9] gives the reference hp-FEM convergence rate which is reported in Figure 1.12. Likewise, exponential convergence occurs with C^{p-1}, p-degree spline approximation on a geometrically graded knot span, as reported in the same figure. Remarkably, convergence is faster (with the constant b in (1.55) that appears to be higher) for smooth splines, even though for splines the degree p is the same for all mesh elements, and grows proportionally with the total number of elements, whereas for hp-FEM a locally varying polynomial degree is utilized on an element-by-element basis.

Exponential convergence for splines is proved in the the main theorem of [39], reported below.

Theorem 1.4.4. *Assume that $f \in H^1_0(0,1)$ and*

$$\left\| \zeta^{\beta+k-2} \frac{\partial^k f}{\partial^k \zeta} \right\|_{L^2(0,1)} \leq C_u d_u^{k-2} (k-2)!, \qquad k = 2, 3, \ldots \tag{1.56}$$

for some $0 < \beta \leq 1$ *and* $C_u, d_u > 0$. *Then there exist* $b > 0$ *and* $C > 0$ *such that for any* $q > 1$, *for any* σ *with* $0 < \sigma < 1$ *and* $1 > \sigma > (1 + 2/d_u)^{-1}$,

$$\inf_{f_h \in S^p(\Xi)} \|f - f_h\|_{H^1(\Omega)} \leq C e^{-b(\sigma,\beta)\sqrt{N}}, \tag{1.57}$$

where $p = 2q + 1$,

$$\Xi = \{\underbrace{0, \ldots, 0}_{p\ times}, \underbrace{\sigma^{p-1}, \ldots, \sigma^{p-1}}_{q\ times}, \underbrace{\sigma^{p-2}, \ldots, \sigma^{p-2}}_{q\ times}, \ldots, \underbrace{\sigma, \ldots, \sigma}_{q\ times}, \underbrace{1, \ldots, 1}_{p\ times}\}, \tag{1.58}$$

and N *is the dimension of* $S^p(\Xi)$.

Condition (1.56) expresses the piecewise analytic regularity of f. Theorem 1.4.4 is based on [21], and as such it covers approximation by $2q + 1$ degree splines having C^q global continuity. However, as is apparent from Figure 1.12, exponential convergence is also observed for maximally smooth splines.

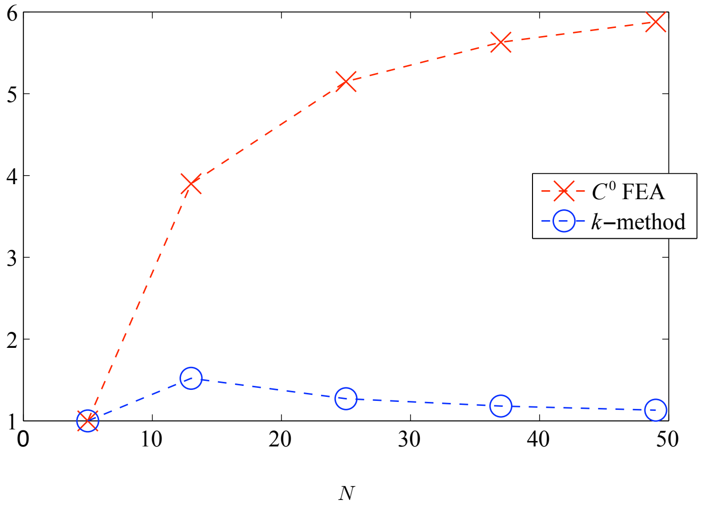


Fig. 1.10 Optimality ratios: comparison between quartic C^3 splines (i.e., $\Lambda(B^5(0,1), S^4_3, L^2(0,1))$, blue line with circles) and C^0 finite elements (i.e., $\Lambda(B^5(0,1), S^4_0, L^2(0,1))$, red line with crosses) on the unit interval for different mesh-sizes h (the total number of degrees-of-freedom N is the abscissa).

1.4.3 Multipatch

While C^0 isogeometric spaces with optimal approximation properties are easy to construct, when the mesh is conforming at the interfaces (see, e.g., [18]), the construction of smooth isogeometric spaces with optimal approximation properties on unstructured geometries is a challenging problem and still open in its full generality. The problem is related to one of accurate representation (fitting) of smooth surfaces having complex topology, which is a fundamental area of research in the community of CAGD.

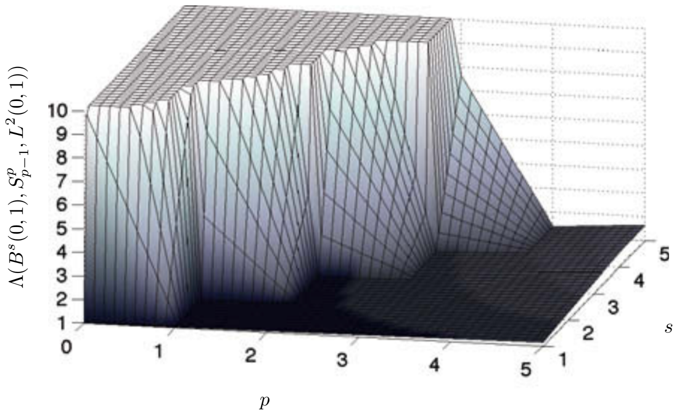


Fig. 1.11 Optimality ratios: for different Sobolev regularity s and for different spline degree p with maximal smoothness. The number of degrees-of-freedom is $N = 30$. The surface plot is capped at 10 for purposes of visualization. Note that if $p \geq s - 1$ the optimality ratio is near 1. Even for low regularity (i.e., low s), smooth splines (i.e., high p) produce optimality ratios near 1. This supports the claim that "smooth splines are always good".

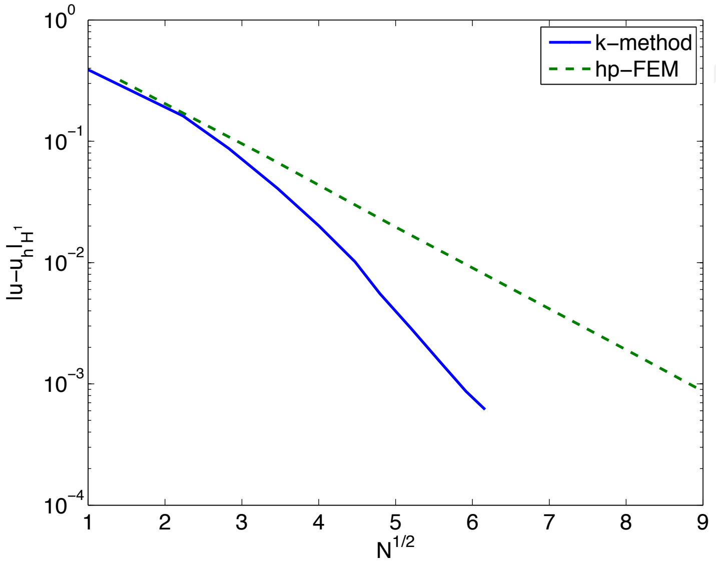


Fig. 1.12 Energy norm error versus the (square root of) number of degrees-of-freedom N for the approximation of the solution $u(x) = x^{0.7} - x$ of the problem $-u'' = f$ with homogeneous Dirichlet boundary conditions. The mesh is geometrically graded (with ratio $q = 0.35$ for IGA) and the spline degree is proportional to the number of elements for IGA, and the smoothness is maximal, that is the spline space is C^{p-1} globally continuous. Mesh-size and degrees are optimally selected for hp-FEM, according to the criteria of [9]. Exponential convergence $|u - u_h|_{H^1} \leq C \exp(-b\sqrt{N})$ is evidenced in both cases, with larger b for IGA.

There are mainly two strategies for constructing smooth multipatch geometries and corresponding isogeometric spaces. One strategy is to adopt a geometry parametrization which is globally smooth almost everywhere, with the exception of a neighborhood of the extraordinary points (or edges in 3D), see Figure 1.13 (left). The other strategy is to use geometry parametrizations that are only C^0 at patch interfaces; see Figure 1.13 (right). The first option includes subdivision surfaces [41] and the T-spline construction in [98] and, while possessing attractive features, it seems to lack optimal approximation properties in general [83], [73]. On the other hand, some optimal constructions have

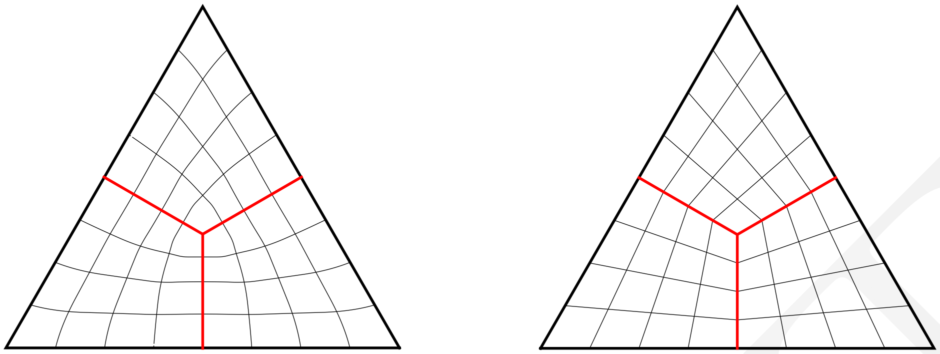


Fig. 1.13 Two possible parametrization schemes: C^1 away from the extraordinary point (left) and C^0 at patch interfaces (right).

been recently obtained following the second strategy, pictured in Figure 1.13 (right) (see [27], [74], [82], [46]). We summarize here the main concepts and results from [46], referring to the paper itself for a complete presentation.

Consider a planar ($d = 2$) spline multipatch domain of interest

$$\Omega = \Omega^{(1)} \cup \ldots \cup \Omega^{(N)} \subset \mathbb{R}^2, \tag{1.59}$$

where the closed sets $\Omega^{(i)}$ form a regular partition without hanging nodes. Assume each $\Omega^{(i)}$ is a non-singular spline patch, with at least C^1 continuity within each patch, and that there exist parametrizations

$$\mathbf{F}^{(i)} : [0,1] \times [0,1] = \widehat{\Omega} \to \Omega^{(i)}, \tag{1.60}$$

where

$$\mathbf{F}^{(i)} \in \mathcal{S}^p(\boldsymbol{\Xi}) \times \mathcal{S}^p(\boldsymbol{\Xi}) \subset C^1(\widehat{\Omega}); \tag{1.61}$$

Furthermore, assume global continuity of the patch parametrizations. This means the following. Let us fix $\Gamma = \Gamma^{(i,j)} = \Omega^{(i)} \cap \Omega^{(j)}$. Let $\mathbf{F}^{(L)}$, $\mathbf{F}^{(R)}$ be given such that

$$\begin{aligned} \mathbf{F}^{(L)} &: [-1,0] \times [0,1] = \widehat{\Omega}^{(L)} \to \Omega^{(L)} = \Omega^{(i)}, \\ \mathbf{F}^{(R)} &: [0,1] \times [0,1] = \widehat{\Omega}^{(R)} \to \Omega^{(R)} = \Omega^{(j)}, \end{aligned} \tag{1.62}$$

where $(\mathbf{F}^{(L)})^{-1} \circ \mathbf{F}^{(i)}$ and $(\mathbf{F}^{(R)})^{-1} \circ \mathbf{F}^{(j)}$ are linear transformations. The set $[-1,1] \times [0,1]$ plays the role of a combined parametric domain. The coordinates in $[-1,1] \times [0,1]$ are denoted u and v. The global continuity condition states that the parametrizations agree at $u = 0$, i.e., there is an $\mathbf{F}_0 : [0,1] \to \mathbb{R}^2$ with

$$\Gamma = \{\mathbf{F}_0(v) = \mathbf{F}^{(L)}(0,v) = \mathbf{F}^{(R)}(0,v), v \in [0,1]\}. \tag{1.63}$$

For the sake of simplicity we assume that the knot vectors of all patches and in each direction coincide, are open and uniform. An example is depicted in Figure 1.14.

The multipatch isogeometric space is given as

$$\mathcal{V} = \left\{ \phi : \Omega \to \mathbb{R} \text{ such that } \phi \circ \mathbf{F}^{(i)} \in \mathcal{S}^p_r(\widehat{\Omega}), i = 1, \ldots, N \right\}; \tag{1.64}$$

the space of continuous isogeometric functions is

$$\mathcal{V}^0 = \mathcal{V} \cap C^0(\Omega), \tag{1.65}$$

and the space of C^1 isogeometric functions is

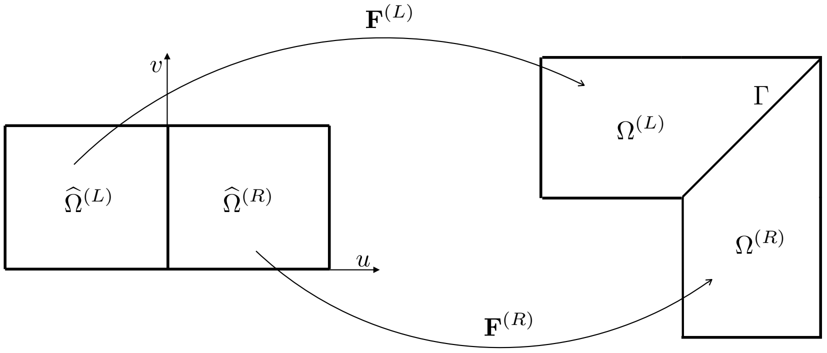


Fig. 1.14 Example of the setting of (1.62)–(1.63).

$$\mathcal{V}^1 = \mathcal{V} \cap C^1(\Omega). \tag{1.66}$$

The graph $\Sigma \subset \Omega \times \mathbb{R}$ of an isogeometric function $\phi : \Omega \to \mathbb{R}$ splits into patches Σ^i having the parametrization

$$\begin{bmatrix} \mathbf{F}^{(i)} \\ g^{(i)} \end{bmatrix} : [0,1] \times [0,1] = \widehat{\Omega} \to \Sigma^{(i)} \tag{1.67}$$

where $g^{(i)} = \phi \circ \mathbf{F}^{(i)}$. As in (1.62), we can select a patch interface $\Gamma = \Gamma^{(i,j)} = \Omega^{(i)} \cap \Omega^{(j)}$, define $g^{(L)}$, $g^{(R)}$ such that

$$\begin{aligned} \begin{bmatrix} \mathbf{F}^{(L)} \\ g^{(L)} \end{bmatrix} &: [-1,0] \times [0,1] = \widehat{\Omega}^{(L)} \to \Sigma^{(i)} = \Sigma^{(L)}, \\ \begin{bmatrix} \mathbf{F}^{(R)} \\ g^{(R)} \end{bmatrix} &: [0,1] \times [0,1] = \widehat{\Omega}^{(R)} \to \Sigma^{(j)} = \Sigma^{(R)}, \end{aligned} \tag{1.68}$$

see Figure 1.15. Continuity of ϕ is implied by the continuity of the graph parametrization, then we set

$$g_0(v) = g^{(L)}(0,v) = g^{(R)}(0,v), \tag{1.69}$$

for all $v \in [0,1]$, analogous to (1.63).

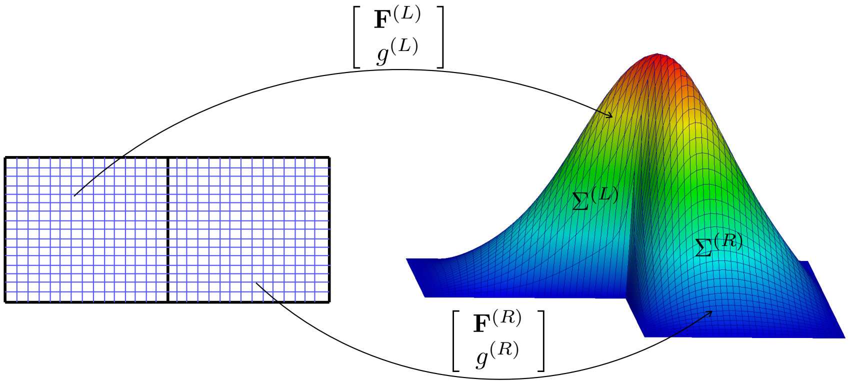

$\begin{bmatrix} \mathbf{F}^{(L)} \\ g^{(L)} \end{bmatrix}$
$\Sigma^{(L)}$
$\Sigma^{(R)}$
$\begin{bmatrix} \mathbf{F}^{(R)} \\ g^{(R)} \end{bmatrix}$

Fig. 1.15 Example of the general setting of (1.68).

Under suitable conditions, smoothness of a function is equivalent to the smoothness of the graph, considered as a geometric entity. In particular, for an isogeometric function that is C^1 within each patch and globally continuous, the global C^1 continuity is then equivalent to the *geometric continuity of order* 1 (in short G^1) of its graph parametrization. Geometric continuity of the graph parametrization means that, on each patch interface, with notation (1.68), the tangent vectors

$$\begin{bmatrix} D_u\mathbf{F}^{(L)}(0,v) \\ D_u g^{(L)}(0,v) \end{bmatrix}, \begin{bmatrix} D_v\mathbf{F}_0(v) \\ D_v g_0(v) \end{bmatrix} \text{ and } \begin{bmatrix} D_u\mathbf{F}^{(R)}(0,v) \\ D_u g^{(R)}(0,v) \end{bmatrix},$$

are co-planar, i.e., linearly dependent. In the CAGD literature, G^1 continuity is commonly stated as below (see, e.g., [17, 79, 86]).

Definition 1.4.5 (G^1-continuity at $\Sigma^{(i)} \cap \Sigma^{(j)}$). Given the parametrizations $\mathbf{F}^{(L)}$, $\mathbf{F}^{(R)}$, $g^{(L)}$, $g^{(R)}$ as in (1.62), (1.68), fulfilling (1.61) and (1.69), we say that the graph parametrization is G^1 at the interface $\Sigma^{(i)} \cap \Sigma^{(j)}$ if there exist $\alpha^{(L)} : [0,1] \to \mathbb{R}$, $\alpha^{(R)} : [0,1] \to \mathbb{R}$ and $\beta : [0,1] \to \mathbb{R}$ such that for all $v \in [0,1]$,

$$\alpha^{(L)}(v)\alpha^{(R)}(v) > 0 \tag{1.70}$$

and

$$\alpha^{(R)}(v) \begin{bmatrix} D_u\mathbf{F}^{(L)}(0,v) \\ D_u g^{(L)}(0,v) \end{bmatrix} - \alpha^{(L)}(v) \begin{bmatrix} D_u\mathbf{F}^{(R)}(0,v) \\ D_u g^{(R)}(0,v) \end{bmatrix} + \beta(v) \begin{bmatrix} D_v\mathbf{F}_0(v) \\ D_v g_0(v) \end{bmatrix} = \mathbf{0}. \tag{1.71}$$

Since the first two equations of (1.71) are linearly independent, $\alpha^{(L)}$, $\alpha^{(R)}$ and β are uniquely determined, up to a common multiplicative factor, by $\mathbf{F}^{(L)}$ and $\mathbf{F}^{(R)}$, i.e. from the equation

$$\alpha^{(R)}(v)D_u\mathbf{F}^{(L)}(0,v) - \alpha^{(L)}(v)D_u\mathbf{F}^{(R)}(0,v) + \beta(v)D_v\mathbf{F}_0(v) = \mathbf{0}. \tag{1.72}$$

We have indeed the following proposition (see [46] and [86]).

Proposition 1.4.6. *Given any $\mathbf{F}^{(L)}$, $\mathbf{F}^{(R)}$ then* (1.72) *holds if and only if $\alpha^{(S)}(v) = \gamma(v)\bar{\alpha}^{(S)}(v)$, for $S \in \{L, R\}$, and $\beta(v) = \gamma(v)\bar{\beta}(v)$, where*

$$\bar{\alpha}^{(S)}(v) = \det\left[\, D_u\mathbf{F}^{(S)}(0,v)\ D_v\mathbf{F}_0(v) \,\right], \tag{1.73}$$

$$\bar{\beta}(v) = \det\left[\, D_u\mathbf{F}^{(L)}(0,v)\ D_u\mathbf{F}^{(R)}(0,v) \,\right], \tag{1.74}$$

and $\gamma : [0,1] \to \mathbb{R}$ is any scalar function. In addition, $\gamma(v) \neq 0$ if and only if (1.70) *holds. Moreover, there exist functions $\beta^{(S)}(v)$, for $S \in \{L, R\}$, such that*

$$\beta(v) = \alpha^{(L)}(v)\beta^{(R)}(v) - \alpha^{(R)}(v)\beta^{(L)}(v). \tag{1.75}$$

In the context of isogeometric methods we consider Ω and its parametrization given. Then for each interface $\alpha^{(L)}$, $\alpha^{(R)}$ and β are determined from (1.72) as stated in Proposition 1.4.6. It should be observed that for planar domains, there always exist $\alpha^{(L)}$, $\alpha^{(R)}$ and β fulfilling (1.72) (this is not the case for surfaces, see [46]). Then, the C^1 continuity of isogeometric functions is equivalent to the last equation in (1.71), that is

$$\alpha^{(R)}(v)D_u g^{(L)}(0,v) - \alpha^{(L)}(v)D_u g^{(R)}(0,v) + \beta(v)D_v g_0(v) = 0 \tag{1.76}$$

for all $v \in [0,1]$. Optimal approximation properties of the isogeometric space on Ω holds under restrictions on $\alpha^{(L)}$, $\alpha^{(R)}$ and β, i.e. on the geometry parametrization. This leads to the definition below ([46]).

Definition 1.4.7 (Analysis-suitable G^1-continuity). $\mathbf{F}^{(L)}$ and $\mathbf{F}^{(R)}$ are *analysis-suitable G^1-continuous* at the interface Γ (in short, AS G^1) if there exist $\alpha^{(L)}, \alpha^{(R)}, \beta^{(L)}, \beta^{(R)} \in \mathcal{P}^1([0,1])$ such that (1.72) and (1.75) hold.

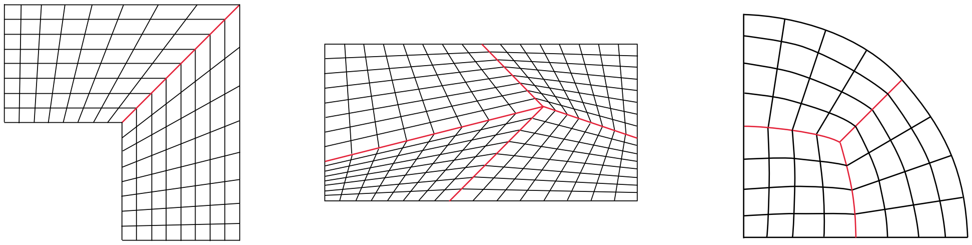


Fig. 1.16 Examples of planar domain having an AS G^1 parametrization.

The class of planar *AS* G^1 parametrizations contains all the bilinear ones and more, see Figure 1.16

In [46], the structure of C^1 isogeometric spaces over AS G^1 geometries is studied, providing an explanation of the optimal convergence of the space of p-degree isogeometric functions, having up to C^{p-2} continuity within the patches (and global C^1 continuity). On the other hand, no convergence under h-refinement occurs for C^{p-1} continuity within the patches. This phenomenon is referred to as C^1 *locking*. Moreover, it is shown that AS G^1 geometries are needed to guarantee optimal convergence, in general.

1.4.4 Singular parametrizations

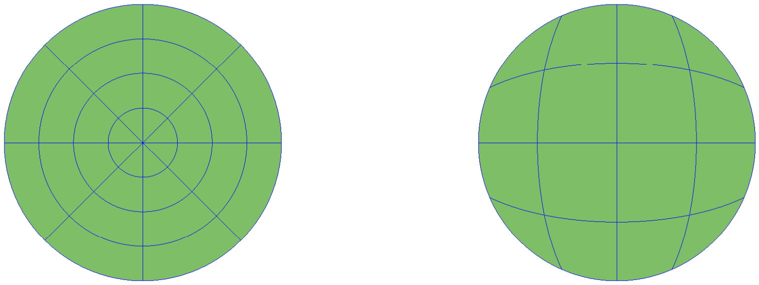
The theory of isogeometric spaces we have reviewed in previous sections assumes that the geometry parametrization is regular. However, singular parametrizations are used in IGA, as they allow more flexibility in the geometry representation. Figure 1.17 shows two examples of this kind, for a a single-patch parametrization of the circle. Typically, a singularity appears when some of the control points near the boundary coincide or are collinear.

Isogeometric spaces with singular mapping have been studied in the papers [106], [107], [108], [110] and [111]. The paper [107] addresses a class of singular geometries that includes the two circles of Figure 1.17. It is shown that in these cases the standard isogeometric spaces, as they are constructed in the non-singular case, are not in $H^1(\Omega)$. However, [107] identifies the subspace of H^1 isogeometric functions, and constructs a basis. The study is generalized to H^2 smoothness in [108]. In [110], function spaces of higher-order smoothness C^k are explicitly constructed on polar parametrizations that are obtained by linear transformation and degree elevation from a triangular Bézier patch. For general parametrizations, [109] gives a representation of the derivatives of isogeometric functions. The theory of approximation properties of isogeometric (sub)spaces for a polar parametrization (as the one in Figure 1.17 (a)) is carried on in [111].

Singular parametrizations can be used to design smooth isogeometric spaces on unstructured multipatch domains. Two different C^1 constructions are proposed in [84] and [118]. In both cases, the singular mapping is employed at the extraordinary vertices.

From the practical point of view, isogeometric methods are surprisingly robust with respect to singular parametrizations. Even if some of the integrals appearing in the linear system matrix are divergent, the use of Gaussian quadrature hides the trouble and the Galerkin variational formulation returns the correct approximation. However, it is advisable to use the correct subspace basis, given in [107] and [108], to avoid ill-conditioning of the isogeometric formulation.

In [11], the authors use isogeometric analysis on the sphere with a polar parametrization (the extension of Figure 1.17 (a)), and benchmark the h-convergence in H^2 and H^3 norms, for solution of 4^{th} and 6^{th} order differential equations, respectively. It is shown that enforcing C^0 continuity at the poles yields optimal convergence, that is, the higher-order smoothness of the isogeometric solution at the poles is naturally enforced by the variational formulation.



(a) One singularity at the origin

(b) Four singularities on the boundary

Fig. 1.17 Two possible singular parametrizations of the circle.

1.5 Isogeometric spaces: spectral properties

We are interested in the Galerkin approximation of the eigenvalues and eigenfunction of the Laplacian differential operator, as a model problem. We will consider mainly the univariate case. As we will see in this section, the use of C^{p-1}-continuous splines yields advantages when compared to standard C^0 FEM. The results shown here are taken from [69, 70]; we refer to that works for more details. Contrary to the previous Section 1.4, the error analysis considered here is *not* asymptotic, rather it may be characterized as a *global analysis* approach.

The asymptotic approach is more commonly found in the literature. Classical functional analysis results state that, given an eigenvalue of the differential operator, for a small enough mesh size this eigenvalue is well approximated in the discrete problem. However, for a given mesh size, this kind of analysis offers no information about which discrete modes are a good approximation of the exact modes, and which ones are not.

What happens in practice is that only the lowest discrete modes are accurate. In general, a large portion of the eigenvalue/eigenfunction spectrum, the so-called "higher modes," are not approximations of their exact counterparts in any meaningful sense. It is well-known in the structural engineering discipline that the higher modes are grossly inaccurate, but the precise point in the spectrum where the eigenvalues and eigenfunctions cease to approximate their corresponding exact counterparts is never known in realistic engineering situations.

First, we focus on the approximations of eigenvalues from a global perspective, that is, we study the approximation errors of the full spectrum. This is done for the simplest possible case, that is the second derivative operator. Based on Fourier/von Neumann analysis, we show that, per degree-of-freedom and for the same polynomial degree p, C^{p-1} splines (i.e., k-method) are more accurate than C^0 splines (p-method), i.e., finite elements.

Then, we study the accuracy of k-method and p-method approximations to the eigenfunctions of the elliptic eigenvalue problem. The inaccuracy of p-method higher modal eigenvalues has been known for quite some time. We show that there are large error spikes in the L^2-norms of the eigenfunction errors centered about the transitions between branches of the p-method eigenvalue spectrum. The k-method errors are better behaved in every respect. The L^2-norms of the eigenfunction errors are indistinguishable from the L^2 best approximation errors of the eigenfunctions. As shown in [69], when solving an elliptic boundary-value problem, or a parabolic or an hyperbolic problem, the error can be expressed entirely in terms of the eigenfunction and eigenvalue errors. This is an important result but the situation is potentially very different for elliptic boundary-value problems and for parabolic and hyperbolic problems. In these cases, all modes may participate in the solution to some extent and inaccurate higher modes may not always be simply ignored. The different mathematical structures of these cases lead to different conclusions. The inaccuracy of the higher p-method modes becomes a significant concern primarily for the hyperbolic initial-value problem, while the k-method produces accurate results in the same circumstances.

1.5.1 Spectrum and dispersion analysis

We consider as a model problem for the eigenvalue study the one of free vibrations of a linear (∞-dimensional) structural system, without damping and force terms:

$$\mathcal{M}\frac{\mathrm{d}^2\mathbf{u}}{\mathrm{d}t^2} + \mathcal{K}\mathbf{u} = \mathbf{0}, \tag{1.77}$$

where $\mathcal{M}$ and $\mathcal{K}$ are, respectively, the mass and stiffness operators, and $\mathbf{u} = \mathbf{u}(t, \mathbf{x})$ is the displacement. The n^{th} normal mode $\boldsymbol{\phi}_n$ and its frequency ω_n are obtained from the eigenvalue problem $\mathcal{K}\boldsymbol{\phi}_n = \omega_n^2\mathcal{M}\boldsymbol{\phi}_n$. Separating the variables as $\mathbf{u}(t, \mathbf{x}) = \sum_n \widehat{u}_n(t)\boldsymbol{\phi}_n(\mathbf{x})$, and, using equation (1.77), we obtain

$$\frac{\mathrm{d}^2\widehat{u}_n(t)}{\mathrm{d}t^2} + \omega_n^2\widehat{u}_n(t) = 0;$$

Then $\widehat{u}_n(t) = C_- e^{-\mathrm{i}\omega_n t} + C_+ e^{\mathrm{i}\omega_n t}$, that is each modal coefficient $\widehat{u}_n$ oscillates at a frequency ω_n. After discretization, the following discrete equations of motion are obtained

$$\mathbf{M}\frac{\mathrm{d}^2\mathbf{u}^h}{\mathrm{d}t^2} + \mathbf{K}\mathbf{u}^h = \mathbf{0}, \tag{1.78}$$

where $\mathbf{M}$ and $\mathbf{K}$ are, respectively, the finite-dimensional consistent mass and stiffness matrices, and $\mathbf{u}^h = \mathbf{u}^h(t, \mathbf{x})$ is the discrete displacement vector. Analogously to the continuum case, the discrete normal modes $\boldsymbol{\phi}_n^h$ and the frequencies ω_n^h are obtained from the eigenproblem

$$\mathbf{K}\boldsymbol{\phi}_n^h = (\omega_n^h)^2\mathbf{M}\boldsymbol{\phi}_n^h, \tag{1.79}$$

and separating the variables as $\mathbf{u}^h(t, \mathbf{x}) = \sum_n \widehat{u}_n^h(t)\boldsymbol{\phi}_n^h(\mathbf{x})$, we end up with $\widehat{u}_n^h$ oscillating at a frequency ω_n^h, that is: $\widehat{u}_n^h = C_- e^{-\mathrm{i}\omega_n^h t} + C_+ e^{\mathrm{i}\omega_n^h t}$. The n^{th} discrete normal mode $\boldsymbol{\phi}_n^h$ is in general different from the n^{th} exact normal mode $\boldsymbol{\phi}_n$ (Figure 1.18), for $n = 1, \ldots, N$, N being the total number of degrees-of-freedom. The corresponding discrete and exact frequencies will be different The target of the frequency analysis is to evaluate how well the discrete spectrum approximates the exact spectrum.

We begin dealing with the eigenproblem (1.79) associated to a linear ($p = 1$) approximation on the one-dimensional domain $(0, L)$. We employ a uniform mesh $0 = \zeta_0 < \zeta_1 < \ldots < \zeta_A < \ldots < \zeta_{N+1} = L$, where the number of elements is $n_{el} = N+1$ and the mesh-size is $h = L/n_{el}$. Considering homogeneous Dirichlet (fixed-fixed) boundary conditions, the eigenproblem (1.79) can be written as

$$\frac{1}{h}(\phi_{A-1} - 2\phi_A + \phi_{A+1}) + \frac{h(\omega^h)^2}{6}(\phi_{A-1} + 4\phi_A + \phi_{A+1}) = 0, \quad A = 1, \ldots, N, \tag{1.80}$$

$$\phi_0 = \phi_{N+1} = 0, \tag{1.81}$$

where N is the total number of degrees-of-freedom, and $\phi_A = \phi^h(\zeta_A)$ is the nodal value of the discrete normal mode at node ζ_A. Equation (1.80) solutions are linear combinations of exponential functions $\phi_A = (\rho_1)^A$ and $\phi_A = (\rho_2)^A$, where ρ_1 and ρ_2 are the distinct roots of the characteristic polynomial

$$(1 - 2\rho + \rho^2) + \frac{(\omega^h h)^2}{6}(1 + 4\rho + \rho^2) = 0. \tag{1.82}$$

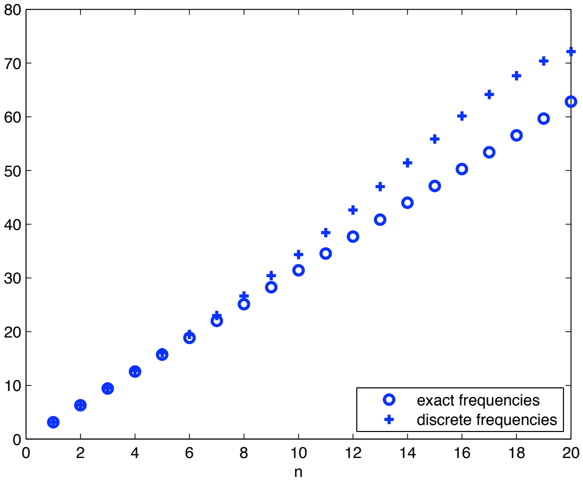


Fig. 1.18 Exact and discrete natural frequencies for the one-dimensional model problem of free vibration of an elastic rod with homogeneous Dirichlet boundary conditions. The discrete method is based on linear finite elements.

Actually, (1.82) admits distinct roots when $\omega^h h \neq 0, \sqrt{12}$; for $\omega^h h = 0$, (1.82) admits the double root $\rho = 1$ (in this case, solutions of (1.80) are combinations of $\phi_A \equiv 1$ and $\phi_A = A$, that is, the affine functions), while for $\omega^h h = \sqrt{12}$ there is a double root $\rho = -1$ (and solutions of (1.80) are combinations of $\phi_A = (-1)^A$ and $\phi_A = A(-1)^A$). Observe that, in general, $\rho_2 = \rho_1^{-1}$. For the purpose of spectrum analysis, we are interested in $0 < \omega^h h < \sqrt{12}$, which we assume for the remainder of this section. In this case, $\rho_{1,2}$ are complex conjugate (we assume $Im(\rho_1) \geq 0$) and of unit modulus. Moreover, in order to compare the discrete spectrum to the exact spectrum, it is useful to represent the solutions of (1.80) as linear combinations of $e^{\pm \mathrm{i} A\omega h}$ (that is, $\phi_A = C_- e^{-\mathrm{i}A\omega h} + C_+ e^{\mathrm{i}A\omega h}$), by introducing ω such that $e^{\mathrm{i}\omega h} = \rho_1$. With this hypothesis, ω is real and, because of periodicity, we restrict to $0 \leq \omega h \leq \pi$. Using this representation in (1.82) and using the identity $2\cos(\alpha) = e^{\mathrm{i}\alpha} + e^{-\mathrm{i}\alpha}$, after simple computations the relation between ωh and $\omega^h h$ is obtained:

$$\frac{(\omega^h h)^2}{6}(2 + \cos(\omega h)) - (1 - \cos(\omega h)) = 0. \tag{1.83}$$

Solving for $\omega^h h \geq 0$, we get

$$\omega^h h = \sqrt{6\,\frac{1 - \cos(\omega h)}{2 + \cos(\omega h)}}. \tag{1.84}$$

Furthermore, taking into account the boundary conditions, (1.80)–(1.81) admit the non-null solution

$$\phi_A = C\frac{e^{+\mathrm{i}An\pi/(N+1)} - e^{-\mathrm{i}An\pi/(N+1)}}{2\mathrm{i}} \equiv C\sin\left(\frac{An\pi}{N+1}\right) \tag{1.85}$$

for all $\omega = \pi/L, 2\pi/L, \ldots, N\pi/L$. Precisely, (1.85) is the n^{th} discrete normal mode, associated to the corresponding n^{th} discrete natural frequency ω^h, given by (1.84):

$$\omega^h = \frac{N+1}{L}\sqrt{6\,\frac{1 - \cos(n\pi/(N+1))}{2 + \cos(n\pi/(N+1))}}. \tag{1.86}$$

The n^{th} discrete mode $\phi_A = C\sin(An\pi/(N+1))$ is the nodal interpolant of the n^{th} exact mode $\phi(x) = C\sin(n\pi x/L)$, whose natural frequency is $\omega = n\pi/L$. The quantity $\dfrac{\omega^h}{\omega} - 1 = \dfrac{\omega^h - \omega}{\omega}$ represents the relative error for the natural frequency. The plot of

$$\frac{\omega^h}{\omega} = \frac{1}{\omega h}\sqrt{6\frac{1-\cos(\omega h)}{2+\cos(\omega h)}} \tag{1.87}$$

is shown in Figure 1.19.

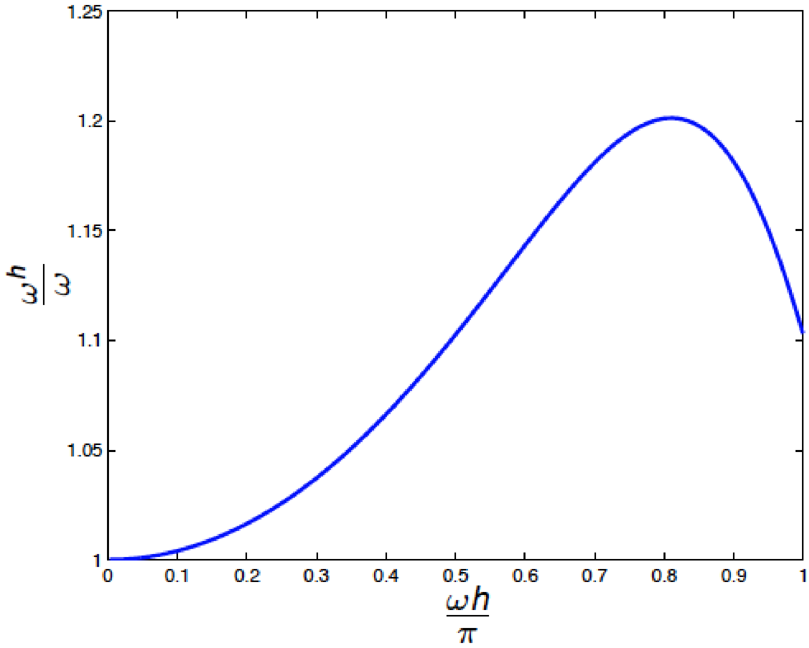


Fig. 1.19 Discrete-to-exact frequencies ratio for linear approximation.

We now consider the quadratic p-method for the eigenproblem (1.79). Assuming to have the same mesh as in the linear case, there are $N = 2n_{el} - 1$ degrees-of-freedom. If we consider the usual Lagrange nodal basis, the corresponding stencil equation is different for element-endpoint degrees-of-freedom and bubble (internal to element) degrees-of-freedom: one has

$$\begin{aligned} &\frac{1}{3h}(-\phi_{A-1} + 8\phi_{A-1/2} - 14\phi_A + 8\phi_{A+1/2} - \phi_{A+1}) \\ &+ (\omega^h)^2\frac{h}{30}(-\phi_{A-1} + 2\phi_{A-1/2} + 8\phi_A + 2\phi_{A+1/2} - \phi_{A+1}) = 0, \quad A = 1,\dots,N. \end{aligned} \tag{1.88}$$

and

$$\frac{1}{3h}(8\phi_A - 16\phi_{A+1/2} + 8\phi_{A+1}) + (\omega^h)^2\frac{h}{30}(2\phi_A + 16\phi_{A+1/2} + 2\phi_{A+1}) = 0, \quad A = 1,\dots,N, \tag{1.89}$$

respectively. We also have the boundary conditions $\phi_0 = \phi_{N+1} = 0$. The bubble degrees-of-freedom can be calculated as

$$\phi_{A+1/2} = \frac{40 + (\omega^h h)^2}{8(10 - (\omega^h h)^2)}(\phi_A + \phi_{A+1}). \tag{1.90}$$

Eliminating them, we obtain a system of equations for the element-endpoints degrees of freedom:

$$\begin{aligned}
&\frac{1}{3h}\left[\left(\frac{30 + 2(\omega^h h)^2}{10 - (\omega^h h)^2}\right)\phi_{A-1} + \left(\frac{-60 + 16(\omega^h h)^2}{10 - (\omega^h h)^2}\right)\phi_A \right.\\
&\qquad\left. + \left(\frac{30 + 2(\omega^h h)^2}{10 - (\omega^h h)^2}\right)\phi_{A+1}\right]\\
&+ k^2\frac{h}{30}\left[\left(\frac{5(\omega^h h)^2}{40 - 4(\omega^h h)^2}\right)\phi_{A-1} + \left(\frac{200 - 15(\omega^h h)^2}{20 - 2(\omega^h h)^2}\right)\phi_A \right.\\
&\qquad\quad\left. + \left(\frac{5(\omega^h h)^2}{40 - 4(\omega^h h)^2}\right)\phi_{A+1}\right] = 0.
\end{aligned} \tag{1.91}$$

for $A = 1, \ldots, N$. The bubble elimination is not possible when the bubble equation (1.89) is singular for $u_{A+1/2}$, that happens for $\omega^h h = \sqrt{10}$.

Normal modes at the element-endpoints nodes can be written as

$$\phi_A = C_- e^{-\mathrm{i}\omega h A} + C_+ e^{\mathrm{i}\omega h A}, \qquad A = 1, \ldots, N. \tag{1.92}$$

The boundary condition $\phi_0 = 0$ determines $C_- = -C_+$, while $\phi_{n_{el}} = 0$ determines $\frac{\omega L}{\pi} \in \mathbb{Z}$. Substituting (1.92) into (1.91), we obtain the relation between $\omega^h h$ and ωh:

$$\cos(\omega h) = \frac{3\,(\omega^h h)^4 - 104\,(\omega^h h)^2 + 240}{(\omega^h h)^4 + 16\,(\omega^h h)^2 + 240}. \tag{1.93}$$

The natural frequencies are obtained solving (1.93) with respect to $\omega^h h$. Unlike the linear case, each real value of ωh is associated with two values of $\omega^h h$, on two different branches, termed *acoustical* and *optical*. It can be shown that a monotone $\omega^h h$ versus ωh relation is obtained representing the two branches in the range $\omega h \in [0, \pi]$ and $\omega h \in [\pi, 2\pi]$ respectively. Therefore, we associate to

$$\omega h = \frac{n\pi}{n_{el}}, \quad n = 1, \ldots n_{el} - 1, \tag{1.94}$$

the smallest positive root of (1.93), obtaining the acoustical branch, and we associate to

$$\omega h = \frac{n\pi}{n_{el}}, \quad n = n_{el} + 1, \ldots 2n_{el} - 1 \equiv N; \tag{1.95}$$

the highest root of (1.93), obtaining the optical branch. These roots are the natural frequencies that can be obtained by bubble elimination. The frequency $\omega^h h = \sqrt{10}$, which gives bubble resonance is associated with the normal mode

$$\begin{aligned}
\phi_A &= 0, & \forall A &= 0, \ldots, n_{el},\\
\phi_{A+1/2} &= C(-1)^A & \forall A &= 0, \ldots, n_{el} - 1.
\end{aligned} \tag{1.96}$$

Since $\omega^h h = \sqrt{10}$ is located between the two branches, this frequency is associated with mode number $n = n_{el}$. Then, all normal modes at element endpoints are given by

$$\phi_A = C \sin\left(\frac{An\pi}{N+1}\right), \qquad A = 0, 1, \ldots n_{el}, \tag{1.97}$$

n being the mode number. Therefore, (1.97) is an interpolate of the exact modes (at element endpoint nodes).

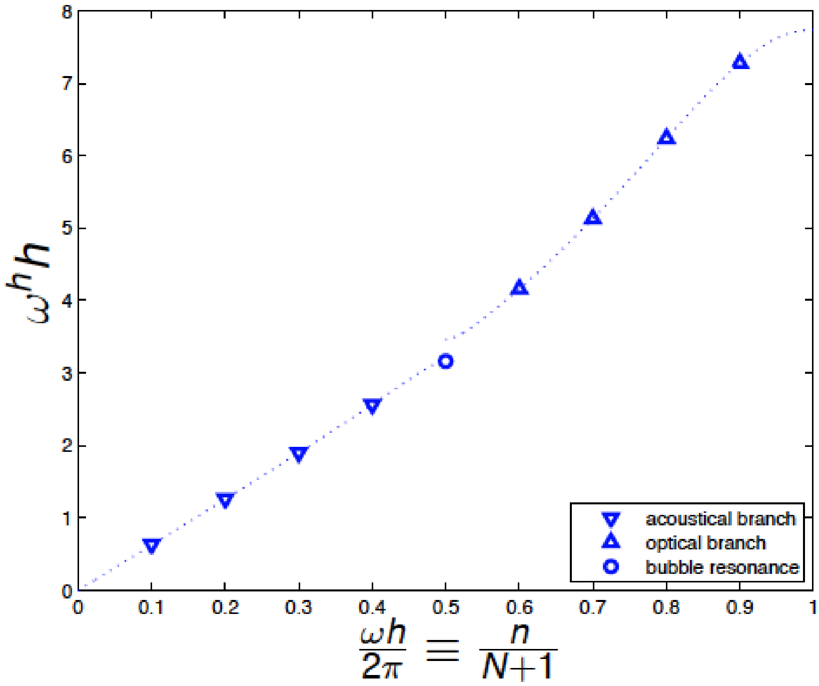


Fig. 1.20 Analytically computed (discrete) natural frequencies for the quadratic p-method ($N = 9$).

The numerical error in the calculation of natural frequencies is visualized by the graph of ω^h/ω versus ωh, shown in Figure 1.21.

Finally, we discuss the quadratic k-method. A rigorous analysis of this case would be too technical; here we prefer to maintain the discussion informal and refer the reader to [70] for the technical details. The equations of (1.79) have different expression for the interior stencil points and for the stencil points close to the boundary (the first and last two equations). We also have for the boundary conditions $\phi_0 = \phi_{N+1} = 0$. In the interior stencil points, the equations read

$$\begin{aligned} &\frac{1}{6h}(\phi_{A-2} + 2\phi_{A-1} - 6\phi_A + 2\phi_{A+1} + \phi_{A+2}) \\ &\quad + (\omega^h)^2 \frac{h}{120}(\phi_{A-2} + 26\phi_{A-1} + 66\phi_A + 26\phi_{A+1} + \phi_{A+2}) = 0, \quad \forall A = 3, \dots, N-2. \end{aligned} \tag{1.98}$$

A major difference from the cases considered previously is that (1.98) is a homogeneous recurrence relation of order 4. Because of its structure, its solutions can be written as linear combinations of the four solutions $e^{\pm \mathrm{i}\omega h A}$ and $e^{\pm \mathrm{i}\widetilde{\omega} h A}$. Here ω^h is real and positive while $\widetilde{\omega}^h$ has a nonzero imaginary part. More precisely, the general solution of (1.98) has the form

$$\phi_A = C_+ e^{\mathrm{i}\omega h A} + C_- e^{-\mathrm{i}\omega h A} + \widetilde{C}_+ e^{\mathrm{i}\widetilde{\omega} h A} + \widetilde{C}_- e^{-\mathrm{i}\widetilde{\omega} h A}, \tag{1.99}$$

for any constants C_+, C_-, $\widetilde{C}_+$, $\widetilde{C}_-$. Plugging this expression of ϕ_A into the boundary equations and imposing the boundary conditions, one finds that $\widetilde{C}_+ = \widetilde{C}_- = 0$ and that $C_+ = -C_-$. Similarly as before, substituting 1.99 into (1.98), we obtain the relation between $\omega^h h$ and ωh:

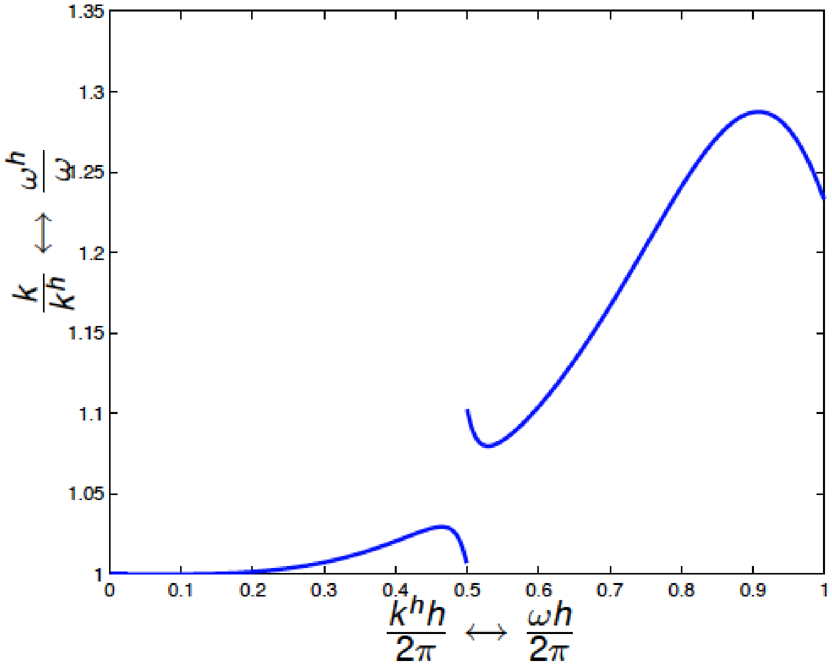


Fig. 1.21 Analytically computed (discrete) natural frequencies for the quadratic p-method.

$$\omega^h h = \sqrt{\frac{20(2-\cos(\omega h)-\cos(\omega h)^2)}{16+13\cos(\omega h)+\cos(\omega h)^2}}. \tag{1.100}$$

The plot of $\omega^h h$ vs. ωh is shown in Figure 1.23

The study above addresses a very simple case but can be generalized. The most interesting direction is to consider arbitrary degree. For degree higher than 2 “outlier frequencies” appear in the k-method: these are $O(p)$ highest frequencies that are numerically spurious and, though they can be filtered out by a suitable geometric parametrization [70] or mesh refinement [42], their full understanding is an open problem. Most importantly, the higher-order p-elements give rise to so-called “optical branches” to spectra, which have no approximation properties, having relative errors that *diverge* with p; on the other hand there are no optical modes with the k-method and, excluding the possible outlier frequencies, the spectral errors *converge* with p. Based on the previous observations, we are able to confidently use numerics to calculate invariant analytical spectra for both p-method and k-method. This comparison is reported in Figure 1.24 and registers a significant advantage for the latter. These results may at least partially explain why classical higher-order finite elements have not been widely adopted in problems for which the upper part of the discrete spectrum participates in a significant way, such as, for example, impact problems and turbulence.

The study can be extended to multidimensional problems as well, mainly confirming the previous findings. We refer again to [70] for the details.

Finally, we present a simple problem that shows how the spectrum properties presented above may affect a numerical solution. Consider the model equation

$$\phi'' + k\phi = 0, \tag{1.101}$$

with boundary conditions

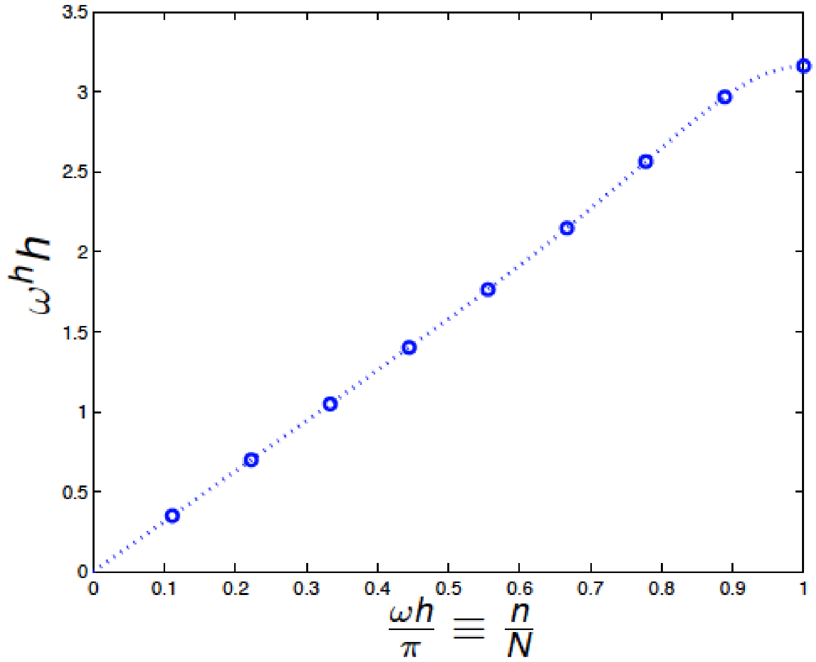


Fig. 1.22 Analytically computed (discrete) natural frequencies for the quadratic k-method ($N = 9$).

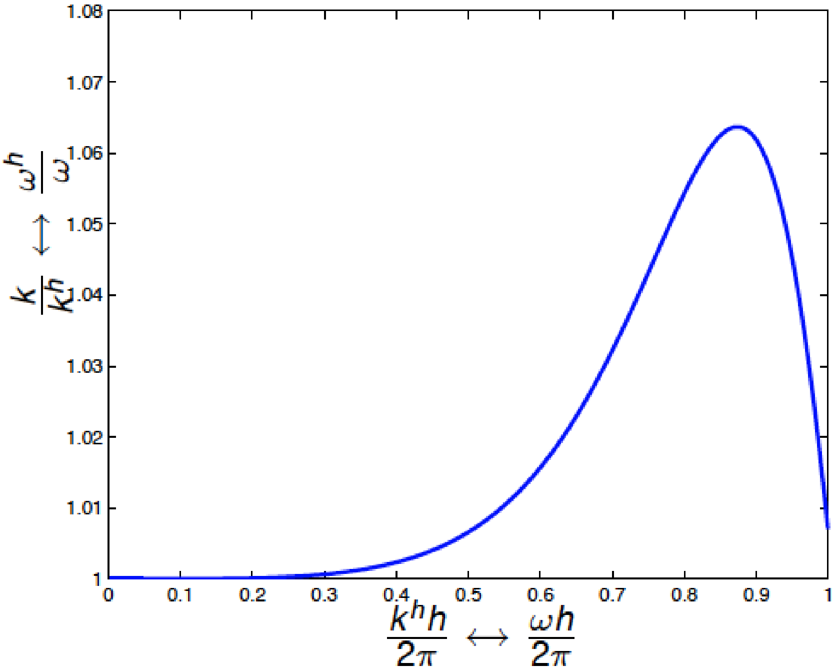


Fig. 1.23 Analytically computed (discrete) natural frequencies for the quadratic k-method.

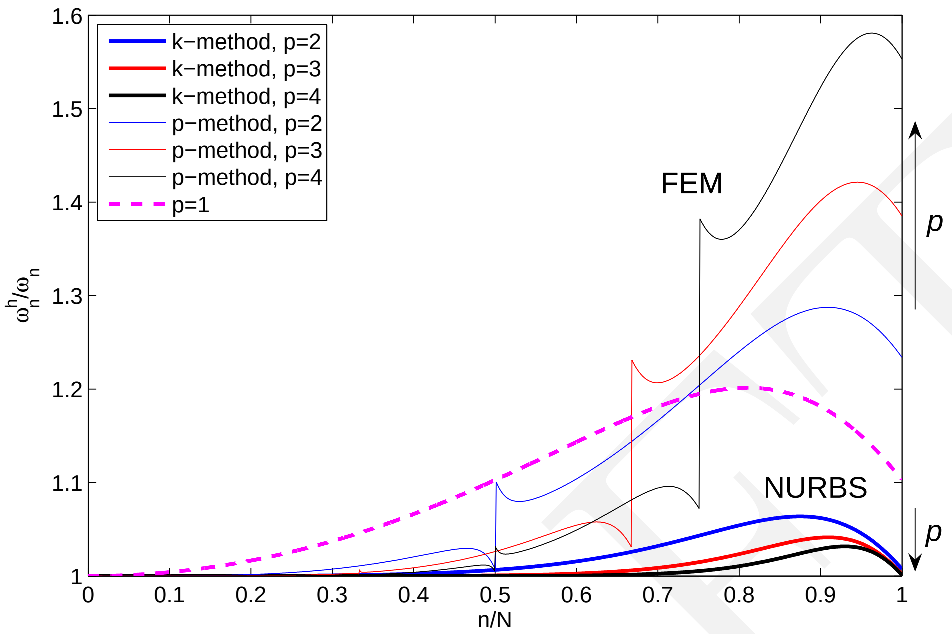


Fig. 1.24 Comparison of k-method and p-method numerical spectra.

$$\phi(0) = 1, \quad \phi(1) = 0. \tag{1.102}$$

The solution to problem (1.101)-(1.102) can be written as

$$\phi(x, k) = \frac{\sin(k(1 - x))}{\sin(k)}. \tag{1.103}$$

We numerically solve (1.101)–(1.102) for $k = 71$, selecting $p = 3$ and 31 degrees-of-freedom for the k- and p-method. The results are reported in Figure 1.25. The k-method is able to reproduce correctly the oscillations of the exact solutions (phase and amplitude are approximately correct). There are no stopping bands for the k-method. On the contrary, since $k = 71$ is within the 2^{nd} stopping band of the p-method, a spurious attenuation is observed. We refer to [70] for the complete study.

1.5.2 Eigenfunction approximation

Let Ω be a bounded and connected domain in $\mathbb{R}^d$, where $d \in \mathbb{Z}^+$ is the number of space dimensions. We assume Ω has a Lipschitz boundary $\partial\Omega$. We assume both are continuous and coercive in the following sense: For all $v, w \in \mathcal{V}$,

$$a(v, w) \leq \|v\|_E \|w\|_E \tag{1.104}$$

$$\|w\|_E^2 = a(w, w) \tag{1.105}$$

$$(v, w) \leq \|v\| \|w\| \tag{1.106}$$

$$\|w\|^2 = (v, w) \tag{1.107}$$

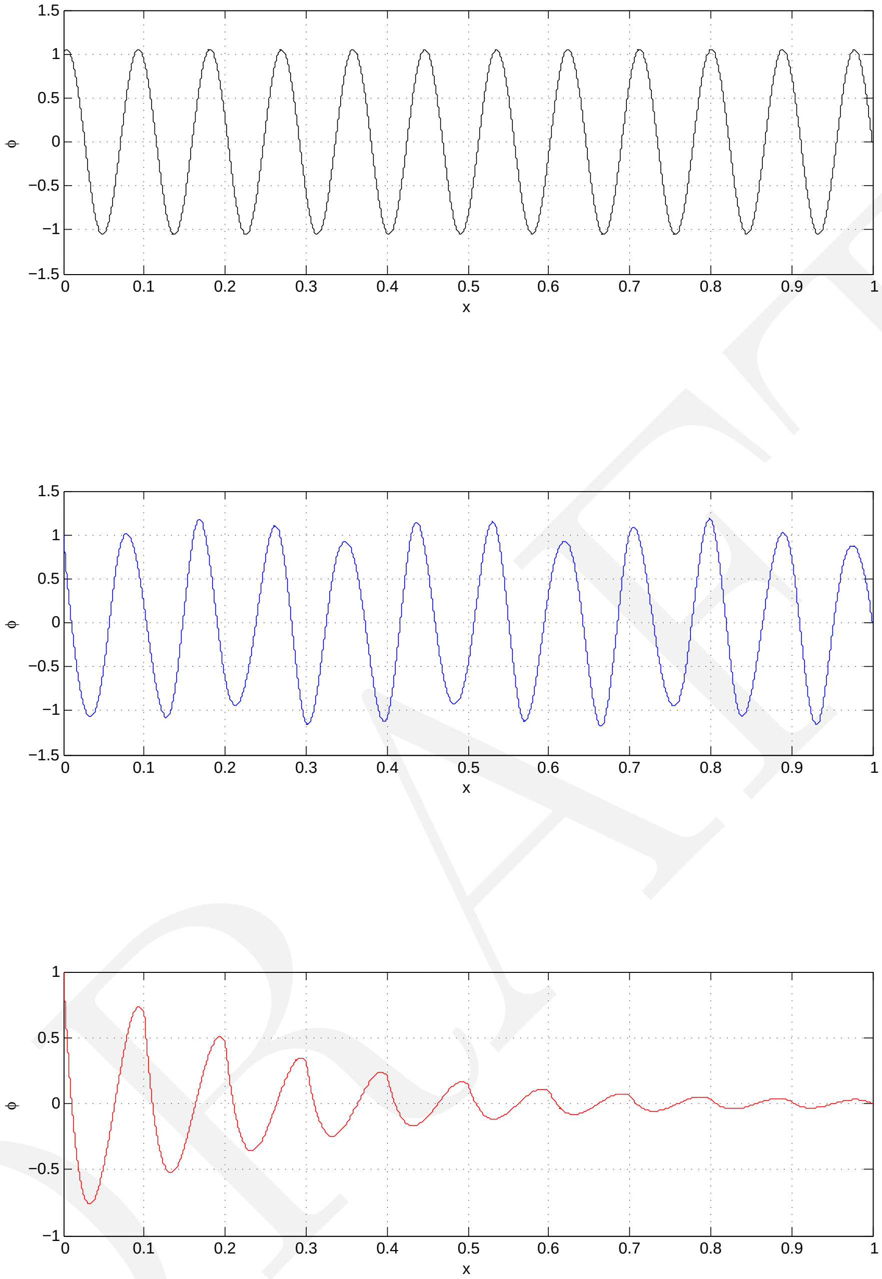


Fig. 1.25 Solutions of the boundary value problem (1.101)–(1.102) for $p = 3$ computed with $k = 71$: exact solution (top), k-method (31 degrees-of-freedom, center) and p-method (31 degrees-of-freedom, bottom).

where $\|\cdot\|_E$ is the energy-norm which is assumed equivalent to the $(H^m(\Omega))^n$-norm on $\mathcal{V}$ and $\|\cdot\|$ is the $(L^2(\Omega))^n = (H^0(\Omega))^n$ norm. The elliptic eigenvalue problem is stated as follows: Find eigenvalues $\lambda_l \in \mathbb{R}^+$ and eigenfunctions $u_l \in \mathcal{V}$, for $l = 1, 2, \ldots, \infty$, such that, for all $w \in \mathcal{V}$,

$$\lambda_l(w, u_l) = a(w, u_l) \tag{1.108}$$

It is well-known that $0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \ldots$, and that the eigenfunctions are $(L^2(\Omega))^n$-orthonormal, that is, $(u_k, u_l) = \delta_{kl}$ where δ_{kl} is the Kronecker delta, for which $\delta_{kl} = 1$ if $k = l$ and $\delta_{kl} = 0$ otherwise. The normalization of the eigenfunctions is actually arbitrary. We have assumed without loss of generality that $\|u_l\| = 1$, for all $l = 1, 2, \ldots, \infty$. It follows from (1.108) that

$$\|u_l\|_E^2 = a(u_l, u_l) = \lambda_l \tag{1.109}$$

and $a(u_k, u_l) = 0$ for $k \neq l$. Let $\mathcal{V}^h$ be either a standard finite element space (p-method) or a space of maximally smooth B-splines (k-method). The discrete counterpart of (1.108) is: Find $\lambda_l^h \in \mathbb{R}^+$ and $u_l^h \in \mathcal{V}^h$ such that for all $w^h \in \mathcal{V}^h$,

$$\lambda_l^h(w^h, u_l^h) = a(w^h, u_l^h) \tag{1.110}$$

The solution of (1.110) has similar properties to the solution of (1.108). Specifically, $0 < \lambda_1^h \leq \lambda_2^h \leq \ldots \leq \lambda_N^h$, where N is the dimension of $\mathcal{V}^h$, $(u_k^h, u_l^h) = \delta_{kl}$, $\|u_l^h\|_E^2 = a(u_l^h, u_l^h) = \lambda_l^h$, and $a(u_k^h, u_l^h) = 0$ if $k \neq l$. The comparison of $\left\{\lambda_l^h, u_l^h\right\}$ to $\{\lambda_l, u_l\}$ for all $l = 1, 2, \ldots, N$ is the key to gaining insight into the errors of the discrete approximations to the elliptic boundary-value problem and the parabolic and hyperbolic initial-value problems.

The fundamental global error analysis result for elliptic eigenvalue problems is the *Pythagorean eigenvalue error theorem*. It is simply derived and is done so on page 233 of Strang and Fix [105] The theorem is global in that it is applicable to each and every mode in the discrete approximation. Provided that $\|u_l^h\| = \|u_l\|$,

$$\frac{\lambda_l^h - \lambda_l}{\lambda_l} + \frac{\|u_l^h - u_l\|^2}{\|u_l\|^2} = \frac{\|u_l^h - u_l\|_E^2}{\|u_l\|_E^2}, \qquad \forall l = 1, 2, \ldots, N \tag{1.111}$$

Note that the relative error in the l^{th} eigenvalue and the square of the relative $(L^2(\Omega))^n$-norm error in the l^{th} eigenfunction sum to equal the square of the relative energy-norm error in the l^{th} eigenfunction. Due to the normalization introduced earlier, (1.111) can also be written as

$$\frac{\lambda_l^h - \lambda_l}{\lambda_l} + \|u_l^h - u_l\|^2 = \frac{\|u_l^h - u_l\|_E^2}{\lambda_l}, \qquad \forall l = 1, 2, \ldots, N \tag{1.112}$$

See Figure 1.26. We note that the first term in (1.112) is always non-negative as $\lambda_l^h \geq \lambda_l$, a consequence of the "minimax" characterization of eigenvalues (see [105], page 223). It also immediately follows from (1.112) that

$$\lambda_l^h - \lambda_l \leq \|u_l^h - u_l\|_E^2 \tag{1.113}$$

$$\|u_l^h - u_l\|^2 \leq \frac{\|u_l^h - u_l\|_E^2}{\lambda_l} \tag{1.114}$$

We consider the elliptic eigenvalue problem for the second-order differential operator in one-dimension with homogeneous Dirichlet boundary conditions. The variational form of the problem is given by (1.108), in which

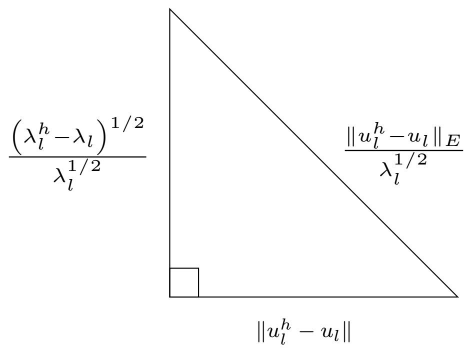

$\frac{\left(\lambda_l^h - \lambda_l\right)^{1/2}}{\lambda_l^{1/2}}$
$\frac{\|u_l^h - u_l\|_E}{\lambda_l^{1/2}}$
$\|u_l^h - u_l\|$

Fig. 1.26 Graphical representation of the Pythagorean eigenvalue error theorem.

$$a(w, u_l) = \int_0^1 \frac{dw}{dx}\frac{du_l}{dx} dx \tag{1.115}$$

$$(w, u_l) = \int_0^1 w u_l dx \tag{1.116}$$

The eigenvalues are $\lambda_l = \pi^2 l^2$ and the eigenfunctions are $u_l = \sqrt{2}\sin(l\pi x)$, $l = 1, 2, \ldots, \infty$. Now, we will present the eigenvalue errors, rather than the eigenfrequency errors, and, in addition, $L^2(0,1)$- and energy-norm eigenfunction errors. We will plot the various errors in a format that represents the Pythagorean eigenvalue error theorem budget. We will restrict our study to quadratic, cubic, and quartic finite elements and B-splines. In all cases, we assume linear geometric parametrizations and uniform meshes. Strictly speaking, for the k-method the results are only true for sufficiently large N, due to the use of open knot vectors, but in this case "sufficiently large" is not very large at all, say $N > 30$. For smaller spaces, the results change slightly. The results that we present here were computed using $N \approx 1,000$ and, in [69], have been validated using a mesh convergence study and by comparing to analytical computations.

Let us begin with results for the quadratic k-method, i.e. C^1-continuous quadratic B-splines, presented in Figure 1.27(a). The results for the relative eigenvalue errors (red curve) follow the usual pattern that has been seen before. The squares of the eigenfunction errors in $L^2(0,1)$ are also well-behaved (blue curve) with virtually no discernible error until about $l/N = 0.6$, and then monotonically increasing errors in the highest modes. The sums of the errors produce the squares of the relative energy-norm errors (black curve), as per the Pythagorean eigenvalue error theorem budget. There are no surprises here.

Next we compare with quadratic p-method, i.e., C^0-continuous quadratic finite elements in Figure 1.27(b). The pattern of eigenvalue errors (red curve), consisting of two branches, the acoustic branch for $l/N < 1/2$, and the optical branch for $l/N \geq 1/2$, is the one known from Section 1.5.1. However, the eigenfunction error in $L^2(0,1)$ (blue curve) represents a surprise in that there is a large spike about $l/N = 1/2$, the transition point between the acoustic and optical branches. Again, the square of the energy-norm eigenfunction error term (black curve) is the sum, as per the budget. This is obviously not a happy result. It suggests that if modes in the neighborhood of $l/N = 1/2$ are participating in the solution of a boundary-value or initial-value problem, the results will be in significant error. The two unpleasant features of this result are (i) the large magnitude of the eigenfunction errors about $l/N = 1/2$ and (ii) the fact that they occur at a relatively low mode number. That the highest modes are significantly in error is well-established for C^0-continuous finite elements, but that there are potential danger zones much earlier in the spectrum had not been recognized previously. The midpoint of the spectrum in one-dimension corresponds to the quarter point in two dimensions and the eighth point in three dimensions, and so one must be aware of the fact that the onset of inaccurate modes occurs much earlier in higher dimensions.

The spikes in the eigenfunction error spectrum for C^0-finite elements raise the question as to whether or not the eigenfunctions are representative of the best approximation to eigenfunctions in the vicinity of $l/N = 1/2$. To answer this question, we computed the $L^2(0, 1)$ best approximations of some of the exact eigenfunctions and plotted them in Figure 1.28(b). (They are indicated by ×.) The case for C^1-continuous quadratic B-splines is presented in Figure 1.28(a) for comparison. For this case there are almost no differences between the best approximation of the exact eigenfunctions and the computed eigenfunctions. However, for the C^0-continuous quadratic finite elements, the differences between the computed eigenfunctions and the $L^2(0, 1)$ best approximations of the exact eigenfunctions are significant, as can be seen in Figure 1.28(b). The spike is nowhere to be seen in the best approximation results. We conclude that the Galerkin formulation of the eigenvalue problem is simply not producing good approximations to the exact eigenfunctions about $l/N = 1/2$ in the finite element case.

For higher-order cases, in particular cubic and quartic, see [69] where it is shown that the essential observations made for the quadratic case persist. An investigation of the behavior of outlier frequencies and eigenfunctions is also presented in [69], along with discussion of the significance of eigenvalue and eigenfunction errors in the context of elliptic, parabolic and hyperbolic partial differential equations.

1.6 Computational efficiency

High-degree high-regularity splines and extensions deliver higher accuracy per degree-of-freedom in comparison to C^0 finite elements but at a higher computational when standard finite element implementation is adopted. In this section we present recent advances on both the formation of the system matri and solution of linear systems of the spline based isogeometric method.

An algorithm, either for the formation of the matrix or for the solution of the linear system, is said to be *efficient* if the computational cost is proportional to the number of non-zero entries of the matrix that have to be calculated (storage cost).

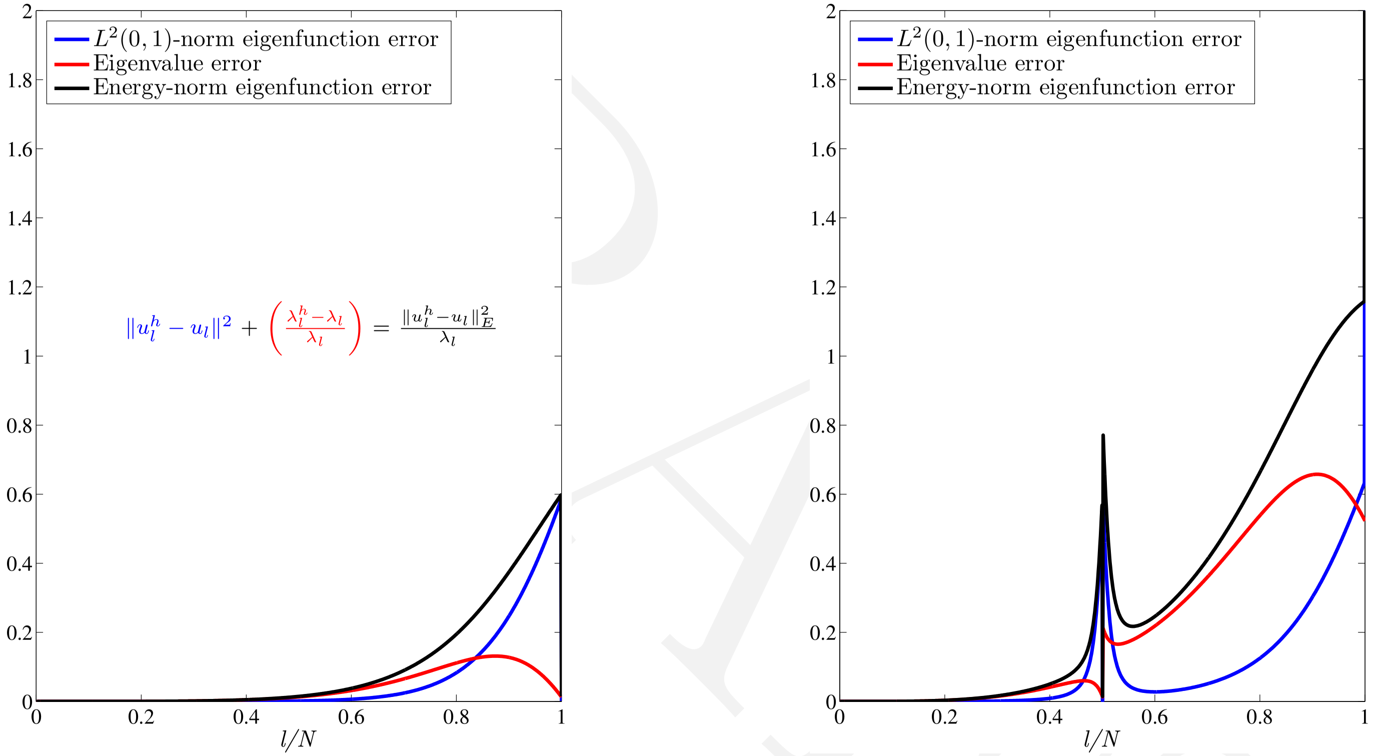
We consider, as a model case, the d-dimensional Poisson problem on a single-patch domain, and an isogeometric tensor-product space of degree p, continuity C^{p-1} and total dimension N, with $N \gg p$. This is the typical setting for the k-method. The resulting *stiffness matrix* has about $N(2p + 1)^d \approx CNp^d$ non-zero entries. Therefore, CNp^d floating point operations FLOPs is the reference computational cost of an optimal isogeometric code implementation.

1.6.1 Formation of isogeometric matrices

When a finite element code architecture is adopted, the simplest approach is to use element-wise Gaussian quadrature and element-by-element assembling. Each elemental stiffness matrix has dimension $(p + 1)^{2d}$ and each entry is calculated by quadrature on $(p + 1)^d$ Gauss points. The total cost is $CN_{\mathsf{EL}}p^{3d} \approx CNp^{3d}$, where N_{EL} is the number of elements and, for the k-method, $N_{\mathsf{EL}} \approx N$.

A strategy to reduce the cost is to reduce the number of quadrature points. The paper [71] proposed to use *generalized* Gaussian rules for smooth spline integrands. These rules are not known analytically and need to be computed numerically (see also [8], [12], [13] and the recent paper [72] where the problem is effectively solved by a Newton method with continuation). Furthermore, *reduced* quadrature rules have been considered in [1], [94] and [64]. Another important step is to reduce the number of operations by arranging the computations in a way that exploits the tensor-product structure of multivariate splines: this is done by so-called sum factorization achieving a computational cost of CNp^{2d+1}, see [2].

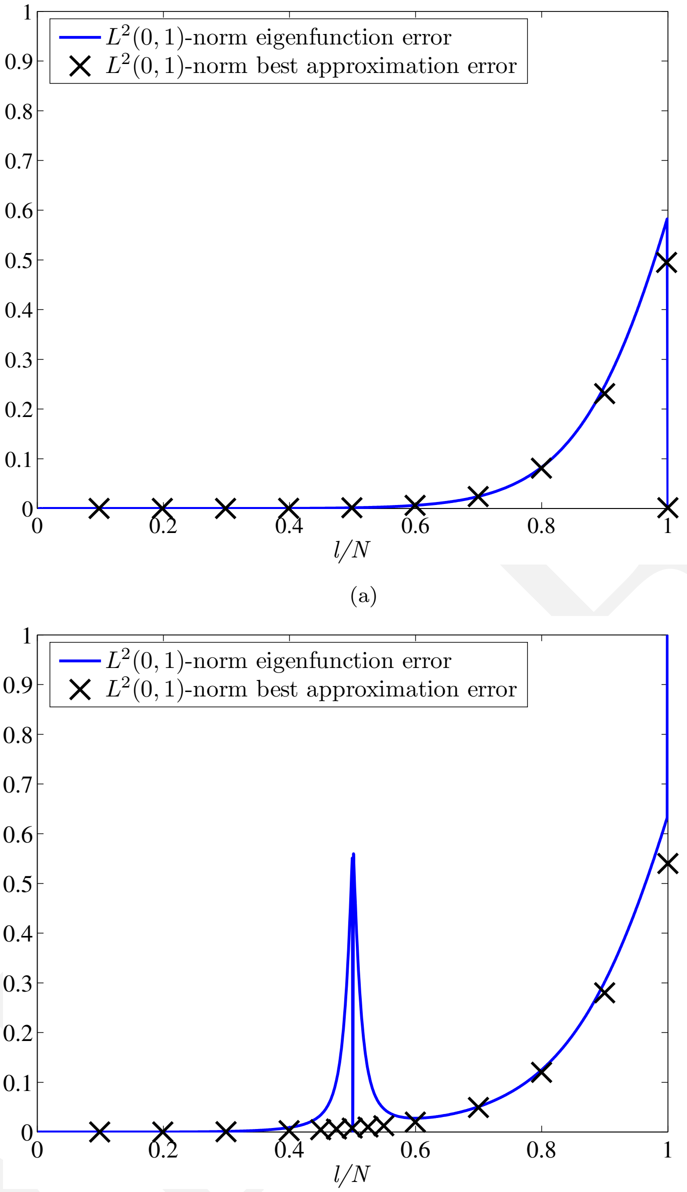
Keeping the element-wise assembling loop is convenient, as it allows reusing available finite element routines. On the other hand, as the computation of each elemental stiffness matrix needs at least Cp^{2d} FLOPs (proportional to the elemental matrix size and assuming integration cost does not depend on p) the total cost is at least $CN_{\mathsf{EL}}p^{2d} \approx CNp^{2d}$.



(a)

(b)

Fig. 1.27 Pythagorean eigenvalue error theorem budget for quadratic elements. (a) C^1-continuous B-splines; (b) C^0-continuous finite elements. The blue curves are $\|u_l^h - u_l\|^2$, the red curves are $(\lambda_l^h - \lambda_l)/\lambda_l$, and the black curves are $\|u_l^h - u_l\|_E^2/\lambda_l$. Note that $\|u_l\| = \|u_l^h\| = 1$, $\|u_l\|_E^2 = \lambda_l$, and $\|u_l^h\|_E^2 = \lambda_l^h$.



(a)

(b)

Fig. 1.28 Comparisons of eigenfunctions computed by the Galerkin method with $L^2(0,1)$ best approximations of the exact eigenfunctions. (a) C^1-continuous quadratic B-splines; (b) C^0-continuous quadratic finite elements. The blue curves are $\|u_l^h - u_l\|^2$, where u_l^h is the Galerkin approximation of u_l, and the ×'s are $\|\widetilde{u}_l^h - u_l\|^2$, where $\widetilde{u}_l^h$ is the $L^2(0,1)$ best approximation of u_l.

Further cost reduction is possible but only with a change of paradigm from element-wise assembling. This study has been recently initiated and two promising strategies have emerged.

The first idea, in [80], is to use a low-rank expansion in order to approximate the stiffness matrix by a sum of R Kronecker type matrices that can be easily formed, thanks to their tensor-product structure. This approach has a computational cost of $CNRp^d$ FLOPs.

The second approach, from [40], is based on two new concepts. The first is the use of a row loop instead of an element loop, and the second is the use of *weighted* quadrature. Consider, for the sake of simplicity, the case of a *mass matrix* $\mathbb{M} = \{m_{\boldsymbol{i},\boldsymbol{j}}\}$ whose entries are

$$m_{\boldsymbol{i},\boldsymbol{j}} = \int_{\widehat{\Omega}} c(\boldsymbol{\zeta})\, B_{\boldsymbol{i}}(\boldsymbol{\zeta})\, B_{\boldsymbol{j}}(\boldsymbol{\zeta})\, d\boldsymbol{\zeta} \ ,$$

where $B_{\boldsymbol{i}}$ and $B_{\boldsymbol{j}}$ are the tensor-product B-splines, c is a non-tensor product coefficient (incorporating the effect of the geometry parametrization) and $\widehat{\Omega} = [0,1]^d$ is the parametric domain. Consider a weighted quadrature rule

$$WQ_{\boldsymbol{i}}(\phi(\cdot)) = \sum_{\boldsymbol{r}} \omega_{\boldsymbol{r},\boldsymbol{i}}\, \phi(\boldsymbol{\eta}_{\boldsymbol{r}}) \approx \int_{\widehat{\Omega}} \phi(\boldsymbol{\zeta})\, B_{\boldsymbol{i}}(\boldsymbol{\zeta}) d\boldsymbol{\zeta},$$

associated with the integration weight $B_{\boldsymbol{i}}(\boldsymbol{\zeta})d\boldsymbol{\zeta}$. This can be used to integrate the $\boldsymbol{i}$-th row of $\mathbb{M}$ as follows

$$WQ_{\boldsymbol{i}}(c(\cdot)B_{\boldsymbol{j}}(\cdot)) \approx \int_{\widehat{\Omega}} c(\boldsymbol{\zeta})\, B_{\boldsymbol{j}}(\boldsymbol{\zeta})\, (B_{\boldsymbol{i}}(\boldsymbol{\zeta}) d\boldsymbol{\zeta}).$$

Observe that $WQ_{\boldsymbol{i}}$ depends on the row index $\boldsymbol{i}$: precisely, in [40] quadrature points $\boldsymbol{\eta}_{\boldsymbol{r}}$ are selected a-priori as suitable interpolation points and without dependence on $\boldsymbol{i}$ while the quadrature weights $\omega_{\boldsymbol{r},\boldsymbol{i}}$ depend on $\boldsymbol{i}$ and are selected from the linear exactness condition

$$\int_{\widehat{\Omega}} B_{\boldsymbol{j}}(\boldsymbol{\zeta})\, (B_{\boldsymbol{i}}(\boldsymbol{\zeta}) d\boldsymbol{\zeta}) = WQ_{\boldsymbol{i}}(B_{\boldsymbol{j}}). \tag{1.117}$$

The quadrature weights outside supp$(B_{\boldsymbol{i}})$ are set to zero by construction. In (1.117), exactness can be required for the B-spline functions $B_{\boldsymbol{j}}$, while for the generalized Gaussian rules of [70] exactness is imposed on the larger space of splines of degree $2p$ and regularity C^{p-1}, in order to exactly integrate the product of B-splines $B_{\boldsymbol{i}}B_{\boldsymbol{j}}$. As a result, for weighted quadrature $\approx 2^d$ quadrature points per element suffice, independent of the polynomial degree, leading to a significant gain in performance. This will be explained in more details in Section 1.6.1.1.

1.6.1.1 Weighted quadrature

Assume we want to compute integrals of the kind:

$$\int_0^1 \widehat{B}_i(\zeta)\, \widehat{B}_j(\zeta)\, d\zeta, \tag{1.118}$$

where $\{\widehat{B}_i\}_{i=1,\ldots,n_{\mathsf{DOF}}}$ are p-degree univariate B-spline basis functions defined on the parametric patch $[0,1]$. We denote by $\boldsymbol{\chi}$ the knot vector of distinct knots that define the univariate B-splines $\widehat{B}_i(\zeta)$. Moreover we define *knot-spans* as the intervals $[\chi_k, \chi_{k+1}], k = 1, \ldots, n_{\mathsf{EL}}$, where $n_{\mathsf{EL}} := (\#\boldsymbol{\chi}) - 1$. The knot vector

$$\Xi := \left\{\xi_1, \xi_2, \ldots, \xi_{n_{\mathsf{KNT}}}\right\} \ . \tag{1.119}$$

that defines the univariate B-splines contains knots with repetitions depending on the regularity: if a knot χ_k has multiplicity $p - r$ then the univariate spline is C^r continuous at χ_k. For simplicity, we consider $r = p - 1$ throughout this paper. Though it is not difficult to consider arbitrary r, the proposed strategy takes advantage of high regularity. In order to focus on the relevant properties, we restrict the attention in this section to the uniform knot-spans, i.e. $\chi_{k+1} - \chi_k = h \ \forall k = 1, \ldots, n_{\mathsf{EL}} - 1$. Moreover, we do not consider boundary functions, so we assume that the knot vector is periodic. Being in the context of Galerkin method, $\widehat{B}_i(\zeta)$ is denoted as a test function and $\widehat{B}_j(\zeta)$ as a trial function.

We are interested in a fixed point quadrature rule. In the lowest degree case, $p = 1$, exact integration is performed by a composite Cavalieri-Simpson rule (note that in this case this quadrature is also the Gauss-Lobatto 3 points rule):

$$\int_0^1 \widehat{B}_i(\zeta)\,\widehat{B}_j(\zeta)\,d\zeta = \mathbb{Q}^{CS}(\widehat{B}_i\,\widehat{B}_j) = \sum_q w_q^{CS}\,\widehat{B}_i(x_q^{CS})\,\widehat{B}_j(x_q^{CS}), \tag{1.120}$$

where x_q^{CS} are the quadrature points and w_q^{CS} the relative weights, see Figure **??**. In the above hypotheses the points x_q^{CS} are the knots and the midpoints of the knot-spans and $w_q^{CS} = \frac{h}{3}$ on knots and $w_q^{CS} = \frac{2h}{3}$ on midpoints.

Unbalancing the role of the test and the trial factors in (1.120), we can see it as a weighted quadrature:

$$\int_0^1 \widehat{B}_i(\zeta)\,\widehat{B}_j(\zeta)\,d\zeta = \mathbb{Q}_i^{WQ}(\widehat{B}_j) = \sum_q w_{q,i}^{WQ}\,\widehat{B}_j(x_{q,i}^{WQ}), \tag{1.121}$$

where $x_q^{CS} = x_{q,i}^{WQ}$ and $w_{q,i}^{WQ} = \widehat{B}_i(x_{q,i}^{WQ}) w_q^{CS}$. Because of the local support of the function $\widehat{B}_i$ only in three points the quadrature $\mathbb{Q}_i^{WQ}$ is non-zero and the weights are equal to $\frac{h}{3}$.

If we go to higher degree, we need more quadrature points in (1.120). For p-degree splines the integrand $\widehat{B}_i\widehat{B}_j$ is a piecewise polynomial of degree $2p$ and an element-wise integration requires $2p+1$ equispaced points, or $p+1$ Gauss points, or about $p/2$ points with generalized Gaussian integration (see [8**? ? ?**]). On the other hand, we can generalize (1.121) to higher degree still using as quadrature points only the knots and midpoints of the knot spans. Indeed this choice ensures that, for each basis function $\widehat{B}_i$, $i = 1, \ldots, n_{\mathsf{DOF}}$, there are $2p+1$ "active" quadrature points where $\widehat{B}_i$ is nonzero. Therefore we can compute the $2p+1$ quadrature weights by imposing conditions for the $2p+1$ B-splines $\widehat{B}_j$ that need to be exactly integrated. Clearly, the advantage of the weighted quadrature approach is that its computational complexity, i.e., the total number of quadrature points, is independent of p.

For the sake of clarity, we first consider the case $p = 1$ in detail. The exactness conditions are:

$$\int_0^1 \widehat{B}_i(\zeta)\,\widehat{B}_{i-1}(\zeta)\,d\zeta = \frac{h}{6} = \mathbb{Q}_i^{WQ}(\widehat{B}_{i-1}) = \frac{1}{2} w_{1,i}^{WQ}, \tag{1.122}$$

$$\int_0^1 \widehat{B}_i^2(\zeta)\,d\zeta = \frac{2h}{3} = \mathbb{Q}_i^{WQ}(\widehat{B}_i) = \frac{1}{2} w_{1,i}^{WQ} + w_{2,i}^{WQ} + \frac{1}{2} w_{3,i}^{WQ}, \tag{1.123}$$

$$\int_0^1 \widehat{B}_i(\zeta)\,\widehat{B}_{i+1}(\zeta)\,d\zeta = \frac{h}{6} = \mathbb{Q}_i^{WQ}(\widehat{B}_{i+1}) = \frac{1}{2} w_{3,i}^{WQ} \ . \tag{1.124}$$

Then it is easy to compute $w_{q,i}^{WQ} = h/3$, $\forall q = 1, 2, 3$.

In the case $p = 2$, five points are active and we have five exactness equations:

$$\int_0^1 \widehat{B}_i(\zeta)\,\widehat{B}_{i-2}(\zeta)\,d\zeta = \frac{h}{120} = \mathbb{Q}_i^{WQ}(\widehat{B}_{i-2}) = \frac{1}{8}w_{1,i}^{WQ}, \tag{1.125}$$

$$\int_0^1 \widehat{B}_i(\zeta)\,\widehat{B}_{i-1}(\zeta)\,d\zeta = \frac{26h}{120} = \mathbb{Q}_i^{WQ}(\widehat{B}_{i-1}) = \frac{3}{4}w_{1,i}^{WQ} + \frac{1}{2}w_{2,i}^{WQ} + \frac{1}{8}w_{3,i}^{WQ}, \tag{1.126}$$

$$\int_0^1 \widehat{B}_i^2(\zeta)\,d\zeta = \frac{66h}{120} = \mathbb{Q}_i^{WQ}(\widehat{B}_i) = \frac{1}{8}w_{1,i}^{WQ} + \frac{1}{2}w_{2,i}^{WQ} + \frac{3}{4}w_{3,i}^{WQ} + \frac{1}{2}w_{4,i}^{WQ} + \frac{1}{8}w_{5,i}^{WQ}, \tag{1.127}$$

$$\int_0^1 \widehat{B}_i(\zeta)\,\widehat{B}_{i+1}(\zeta)\,d\zeta = \frac{26h}{120} = \mathbb{Q}_i^{WQ}(\widehat{B}_{i+1}) = \frac{1}{8}w_{3,i}^{WQ} + \frac{1}{2}w_{4,i}^{WQ} + \frac{3}{4}w_{5,i}^{WQ}, \tag{1.128}$$

$$\int_0^1 \widehat{B}_i(\zeta)\,\widehat{B}_{i+2}(\zeta)\,d\zeta = \frac{h}{120} = \mathbb{Q}_i^{WQ}(\widehat{B}_{i+2}) = \frac{1}{8}w_{5,i}^{WQ}\ . \tag{1.129}$$

In the previous calculation we have used the usual properties of B-splines that can be found, e.g., in [? , Section 4.4]. The system can be solved and leads to the following solution $w_{q,i}^{WQ} = \frac{h}{30}\,[2, 7, 12, 7, 2]$.

When $p = 3$, the same approach gives $w_{q,i}^{WQ} = h\left[\frac{1}{105}, \frac{3}{35}, \frac{5}{21}, \frac{1}{3}, \frac{5}{21}, \frac{3}{35}, \frac{1}{105}\right]$.

In general case (arbitrary degree and non-uniform spacing, boundary functions, lower regularity ...) the rule can be computed numerically as solution of a linear system, see Section **??**.

Given a weighted quadrature rule of the kind above, we are then interested in using it for the approximate calculation of integrals as:

$$\int_0^1 c(\zeta)\widehat{B}_i(\zeta)\,\widehat{B}_j(\zeta)\,d\zeta \approx \mathbb{Q}_i^{WQ}\left(c(\cdot)\widehat{B}_j(\cdot)\right) = \sum_q w_{q,i}^{WQ} c(x_{q,i}^{WQ})\widehat{B}_j(x_{q,i}^{WQ})\ . \tag{1.130}$$

For a non-constant function $c(\cdot)$, (1.130) is in general just an approximation. In particular, the symmetry of the integral is not preserved, that is $\mathbb{Q}_i^{WQ}\left(c(\cdot)\widehat{B}_j(\cdot)\right)$ is different from $\mathbb{Q}_j^{WQ}\left(c(\cdot)\widehat{B}_i(\cdot)\right)$. Consider, for example, the case $p = 3$ derived above and apply the weighted quadrature rules to the linear function $c(\zeta) = \zeta$ in the case $j = i + 1$. For simplicity we take $h = 2$ so that the quadrature points are $x_{q,i}^{WQ} = [1 : 7]$. Then:

$$\begin{aligned}
\mathbb{Q}_i^{WQ}\left(c(\cdot)\widehat{B}_j(\cdot)\right) &= \frac{10}{21}3\frac{1}{48} + \frac{2}{3}4\frac{1}{6} + \frac{10}{21}5\frac{23}{48} + \frac{6}{35}6\frac{2}{3} + \frac{2}{105}7\frac{23}{48} \approx 2.3647,\\
\mathbb{Q}_j^{WQ}\left(c(\cdot)\widehat{B}_i(\cdot)\right) &= \frac{2}{105}3\frac{23}{48} + \frac{6}{35}4\frac{2}{3} + \frac{10}{21}5\frac{23}{48} + \frac{2}{3}6\frac{1}{6} + \frac{10}{21}7\frac{1}{48} \approx 2.3615.
\end{aligned}$$

A detailed mathematical analysis of the quadrature error of weighted quadrature is of key interest, especially in the context of isogeometric Galerkin methods. This is however beyond the scope of this paper and for its importance deserves future work.

We consider now the model reaction-diffusion problem

$$\begin{cases} -\nabla^2 u + u = f & \text{on} \quad \Omega,\\ \qquad\quad u = 0 & \text{on} \quad \partial\Omega, \end{cases} \tag{1.131}$$

Its Galerkin approximation on a discrete space V requires the formation of the stiffness matrix S and mass matrix M whose entries are

$$s_{\boldsymbol{i},\boldsymbol{j}} = \int_\Omega \nabla R_{\boldsymbol{i}}(\boldsymbol{x})\nabla R_{\boldsymbol{j}}(\boldsymbol{x})d\boldsymbol{x}, \tag{1.132}$$

$$m_{\boldsymbol{i},\boldsymbol{j}} = \int_\Omega R_{\boldsymbol{i}}(\boldsymbol{x})R_{\boldsymbol{j}}(\boldsymbol{x})d\boldsymbol{x}, \tag{1.133}$$

$R_{\boldsymbol{i}}$ and $R_{\boldsymbol{j}}$ being two basis functions in V. The dimension of the space V is $N_{\mathsf{DOF}} := \#V$. Non-constant coefficients could be included as well. Assume Ω is given by a d-dimensional single patch spline representation. Integrals are computed by change of variable. Summarizing, we are interested in the computation of (1.133) after change of variable, $\mathbb{M} = \{m_{\boldsymbol{i},\boldsymbol{j}}\} \in \mathbb{R}^{N_{\mathsf{DOF}} \times N_{\mathsf{DOF}}}$ where:

$$m_{\boldsymbol{i},\boldsymbol{j}} = \int_{\widehat{\Omega}} \widehat{B}_{\boldsymbol{i}}\, \widehat{B}_{\boldsymbol{j}} \det \widehat{\boldsymbol{D}}\boldsymbol{F}\, d\boldsymbol{\zeta}\ .$$

For notational convenience we write:

$$m_{\boldsymbol{i},\boldsymbol{j}} = \int_{\widehat{\Omega}} \widehat{B}_{\boldsymbol{i}}(\boldsymbol{\zeta})\, \widehat{B}_{\boldsymbol{j}}(\boldsymbol{\zeta})\, c(\boldsymbol{\zeta})\, d\boldsymbol{\zeta}\ . \tag{1.134}$$

In more general cases, the factor c incorporates the coefficient of the equation and, for NURBS functions, the polynomial denominator. Similarly for the stiffness matrix $\mathbb{S} = \{s_{\boldsymbol{i},\boldsymbol{j}}\} \in \mathbb{R}^{N_{\mathsf{DOF}} \times N_{\mathsf{DOF}}}$ we have:

$$\begin{aligned} s_{\boldsymbol{i},\boldsymbol{j}} &= \int_{\widehat{\Omega}} \left(\widehat{\boldsymbol{D}}\boldsymbol{F}^{-T} \widehat{\nabla} \widehat{B}_{\boldsymbol{i}}\right)^T \left(\widehat{\boldsymbol{D}}\boldsymbol{F}^{-T} \widehat{\nabla} \widehat{B}_{\boldsymbol{j}}\right) \det \widehat{\boldsymbol{D}}\boldsymbol{F}\, d\boldsymbol{\zeta} \\ &= \int_{\widehat{\Omega}} \widehat{\nabla} \widehat{B}_{\boldsymbol{i}}^T \left([\widehat{\boldsymbol{D}}\boldsymbol{F}^{-1} \widehat{\boldsymbol{D}}\boldsymbol{F}^{-T}] \det \widehat{\boldsymbol{D}}\boldsymbol{F} \right) \widehat{\nabla} \widehat{B}_{\boldsymbol{j}}\, d\boldsymbol{\zeta} \end{aligned}$$

which we write in compact form:

$$s_{\boldsymbol{i},\boldsymbol{j}} = \sum_{l,m=1}^{d} \int_{\widehat{\Omega}} \left(\widehat{\nabla} \widehat{B}_{\boldsymbol{i}}(\boldsymbol{\zeta})\right)_l c_{l,m}(\boldsymbol{\zeta}) \left(\widehat{\nabla} \widehat{B}_{\boldsymbol{j}}(\boldsymbol{\zeta})\right)_m d\boldsymbol{\zeta}. \tag{1.135}$$

Here we have denoted by $\left\{c_{l,m}(\boldsymbol{\zeta})\right\}_{l,m=1,\ldots,d}$ the following matrix:

$$c_{l,m}(\boldsymbol{\zeta}) = \left\{ [\widehat{\boldsymbol{D}}\boldsymbol{F}^{-1}(\boldsymbol{\zeta}) \widehat{\boldsymbol{D}}\boldsymbol{F}^{-T}(\boldsymbol{\zeta})] \det \widehat{\boldsymbol{D}}\boldsymbol{F}(\boldsymbol{\zeta}) \right\}_{l,m}. \tag{1.136}$$

The number of non-zero elements N_{NZ} of $\mathbb{M}$ and $\mathbb{S}$ (the same for simplicity) depends on the polynomial degree p and the required regularity r. We introduce the following sets, where the support is considered an open set:

$$\mathcal{K}_{l,i_l} = \left\{ k \in \{1, \ldots, n_{\mathsf{EL},l}\}\ s.t.\ ]\chi_{k-1}, \chi_k[\subset \operatorname{supp}\left(\widehat{B}_{i_l}\right) \right\}\ , \tag{1.137}$$

$$\mathcal{I}_{l,i_l} = \left\{ j_l \in \{1, \ldots, n_{\mathsf{DOF},l}\}\ s.t.\ \widehat{B}_{i_l} \cdot \widehat{B}_{j_l} \neq 0 \right\}\ ; \tag{1.138}$$

and the related multi-indexes as:

$$\mathcal{K}_{\boldsymbol{i}} = \prod_{l=1}^{d} \mathcal{K}_{l,i_l} \qquad \mathcal{I}_{\boldsymbol{i}} = \prod_{l=1}^{d} \mathcal{I}_{l,i_l}\ . \tag{1.139}$$

We have $\#\mathcal{I}_{l,i} \leq (2p+1)$ and $N_{\mathsf{NZ}} = O(N_{\mathsf{DOF}}\, p^d)$. In particular, with maximal regularity in the case $d = 1$ one has $N_{\mathsf{NZ}} = (2p+1)N_{\mathsf{DOF}} - p(p+1)$.

Consider the calculation of the mass matrix (1.133). The first step is to write the integral in a nested way, as done in [2]:

$$m_{\boldsymbol{i},\boldsymbol{j}} = \int_{\widehat{\Omega}} \widehat{B}_{\boldsymbol{i}}(\boldsymbol{\zeta})\widehat{B}_{\boldsymbol{j}}(\boldsymbol{\zeta})c(\boldsymbol{\zeta})\,d\boldsymbol{\zeta} = \\ \int_0^1 \widehat{B}_{i_1}(\zeta_1)\widehat{B}_{j_1}(\zeta_1)\left[\int_0^1 \widehat{B}_{i_2}(\zeta_2)\widehat{B}_{j_2}(\zeta_2)\cdots\left[\int_0^1 \widehat{B}_{i_d}(\zeta_d)\widehat{B}_{j_d}(\zeta_d)c(\boldsymbol{\zeta})\,d\zeta_d\right]\cdots d\zeta_2\right]d\zeta_1$$

The idea in is to isolate the *test function* $\widehat{B}_{i_l}$ univariate factors in each univariate integral and to consider it as a weight for the construction of the weighted quadrature (WQ) rule. This leads to a quadrature rule for each i_l that is:

$$\begin{aligned} m_{\boldsymbol{i},\boldsymbol{j}} \approx \widetilde{m}_{\boldsymbol{i},\boldsymbol{j}} &= \mathbb{Q}^{WQ}_{\boldsymbol{i}}\left(\widehat{B}_{\boldsymbol{j}}(\zeta)c(\boldsymbol{\zeta})\right) = \mathbb{Q}_{\boldsymbol{i}}\left(\widehat{B}_{\boldsymbol{j}}(\zeta)c(\boldsymbol{\zeta})\right) \\ &= \mathbb{Q}_{i_1}\left(\widehat{B}_{j_1}(\zeta_1)\mathbb{Q}_{i_2}\left(\cdots\mathbb{Q}_{i_d}\left(\widehat{B}_{j_d}(\zeta_d)c(\boldsymbol{\zeta})\right)\right)\right)\,. \end{aligned} \tag{1.140}$$

Notice that we drop from now on the label WQ used in the introduction in order to simplify notation. The key ingredients for the construction of the quadrature rules that preserve the optimal approximation properties are the exactness requirements. Roughly speaking, exactness means that in (1.140) we have $m_{\boldsymbol{i},\boldsymbol{j}} = \widetilde{m}_{\boldsymbol{i},\boldsymbol{j}}$ whenever c is a constant coefficient. When the stiffness term is considered, also terms with derivatives have to be considered.

We introduce the notation:

$$\begin{aligned} \mathbb{I}^{(0,0)}_{l,i_l,j_l} &:= \int_0^1 \widehat{B}_{i_l}(\zeta_l)\widehat{B}_{j_l}(\zeta_l)\,d\zeta_l \\ \mathbb{I}^{(1,0)}_{l,i_l,j_l} &:= \int_0^1 \widehat{B}'_{i_l}(\zeta_l)\widehat{B}_{j_l}(\zeta_l)\,d\zeta_l \\ \mathbb{I}^{(0,1)}_{l,i_l,j_l} &:= \int_0^1 \widehat{B}_{i_l}(\zeta_l)\widehat{B}'_{j_l}(\zeta_l)\,d\zeta_l \\ \mathbb{I}^{(1,1)}_{l,i_l,j_l} &:= \int_0^1 \widehat{B}'_{i_l}(\zeta_l)\widehat{B}'_{j_l}(\zeta_l)\,d\zeta_l \end{aligned} \tag{1.141}$$

For each integral in (1.141) we define a quadrature rule: we look for

- points $\widetilde{\boldsymbol{x}}_{\boldsymbol{q}} = (\widetilde{x}_{1,q_1}, \widetilde{x}_{2,q_2}, \dots, \widetilde{x}_{d,q_d})$ with $q_l = 1, \dots n_{\mathsf{QP},l}$, with N_{QP} is $\#\{\widetilde{\boldsymbol{x}}\} = \prod_{l=1}^{d} n_{\mathsf{QP},l}$;
- for each index $i_l = 1, \dots, n_{\mathsf{DOF},l}; l = 1, \dots, d$, four quadrature rules such that:

$$\begin{aligned} \mathbb{Q}^{(0,0)}_{i_l}(f) &:= \sum_{q_l=1}^{n_{\mathsf{QP},l}} w^{(0,0)}_{l,i_l,q_l} f(\widetilde{x}_{l,q_l}) \approx \int_0^1 f(\zeta_l)\widehat{B}_{i_l}(\zeta_l)d\zeta_l\,; \\ \mathbb{Q}^{(1,0)}_{i_l}(f) &:= \sum_{q_l=1}^{n_{\mathsf{QP},l}} w^{(1,0)}_{l,i_l,q_l} f(\widetilde{x}_{l,q_l}) \approx \int_0^1 f(\zeta_l)\widehat{B}_{i_l}(\zeta_l)d\zeta_l\,; \\ \mathbb{Q}^{(0,1)}_{i_l}(f) &:= \sum_{q_l=1}^{n_{\mathsf{QP},l}} w^{(0,1)}_{l,i_l,q_l} f(\widetilde{x}_{l,q_l}) \approx \int_0^1 f(\zeta_l)\widehat{B}'_{i_l}(\zeta_l)d\zeta_l\,; \\ \mathbb{Q}^{(1,1)}_{i_l}(f) &:= \sum_{q_l=1}^{n_{\mathsf{QP},l}} w^{(1,1)}_{l,i_l,q_l} f(\widetilde{x}_{l,q_l}) \approx \int_0^1 f(\zeta_l)\widehat{B}'_{i_l}(\zeta_l)d\zeta_l\,. \end{aligned} \tag{1.142}$$

fulfilling the exactness requirement:

$$\left.\begin{aligned} \mathbb{Q}^{(0,0)}_{i_l}(\widehat{B}_{j_l}) &= \mathbb{I}^{(0,0)}_{l,i_l,j_l} \\ \mathbb{Q}^{(1,0)}_{i_l}(\widehat{B}'_{j_l}) &= \mathbb{I}^{(1,0)}_{l,i_l,j_l} \\ \mathbb{Q}^{(0,1)}_{i_l}(\widehat{B}_{j_l}) &= \mathbb{I}^{(0,1)}_{l,i_l,j_l} \\ \mathbb{Q}^{(1,1)}_{i_l}(\widehat{B}'_{j_l}) &= \mathbb{I}^{(1,1)}_{l,i_l,j_l} \end{aligned}\right.\ ,\ \forall j_l \in \mathcal{I}_{l,i_l}\,. \tag{1.143}$$

For stability we also require that the quadrature rules $\mathbb{Q}_{i_l}^{(\cdot,\cdot)}$ have support included in the support of $\widehat{B}_{i_l}$, that is

$$q_l \notin \mathcal{Q}_{l,i_l} \Rightarrow w_{l,i_l,q_l}^{(\cdot,\cdot)} = 0 \ . \tag{1.144}$$

where $\mathcal{Q}_{l,i_l} := \left\{ q_l \in 1,\ldots,n_{\mathsf{QP},l} \text{ s.t. } \widetilde{x}_{l,q_l} \in \operatorname{supp}\left(\widehat{B}_{i_l}\right) \right\}$; recall that here the support of a function is considered an open set. Correspondingly, we introduce the set of multi-indexes $\mathcal{Q}_{\boldsymbol{i}} := \prod_{l=1}^{d} \mathcal{Q}_{l,i_l}$.

Once the points $\widetilde{\boldsymbol{x}}_{\boldsymbol{q}}$ are fixed, the quadrature rules have to be determined by the exactness requirements, that are a system of linear equations of the unknown weights (each of the (1.143)). For that we require

$$\#\mathcal{Q}_{l,i_l} \geq \#\mathcal{I}_{l,i_l} \ . \tag{1.145}$$

See Remark 1.6.2 for a discussion on the well-posedness of the linear systems for the weights.

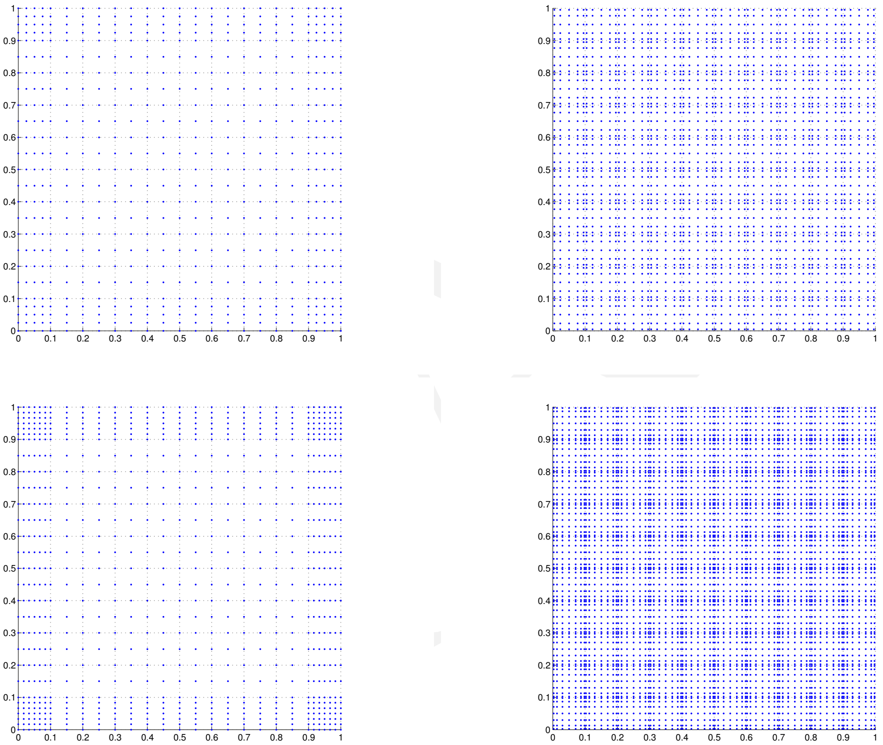


Fig. 1.29 Comparison between quadrature rules. On the first line, the quadrature points needed for the case $p = 4$, $n_{\mathsf{EL}} = 10$, $d = 2$, in the second line the case $p = 6$. On the left panel the proposed WQ rule, on the right the SGQ rule.

Remark 1.6.1 (The choice of quadrature points). The construction of a global grid of quadrature points is done in order to save computations. For the case of maximum C^{p-1} regularity considered here, the choice for quadrature points of [40] is endpoints (knots) and midpoints of all internal knot-spans, while for the boundary knot-spans (i.e. those that are adjacent to the boundary of the parameter domain $\widehat{\Omega}$) we take $p+1$ equally spaced points. Globally $N_{\mathsf{QP}} \approx 2^d N_{\mathsf{EL}} = O(N_{\mathsf{DOF}})$ considering only the dominant term (remember that $n_{\mathsf{EL},l} \gg p$). In Figure 1.29 we plot the quadrature points grid, and a comparison is made with respect to element-by-element standard Gaussian quadrature (SGQ) points.

Remark 1.6.2 (Computation of quadrature weights). Given the quadrature points, the quadrature weights are selected in order to fulfil (1.143)–(1.144). When $\#\mathcal{Q}_{l,i_l} > \#\mathcal{I}_{l,i_l}$ the quadrature weights are not uniquely given from (1.143)–(1.144) and are selected by a minimum norm condition. In all cases with the choice of quadrature points and thanks to the Schoenberg-Whitney interpolation theorem we can guarantee that the quadrature weights fulfilling the above conditions exist.

In order to simplify the FLOPs count, we assume $n_{\mathsf{EL}} := n_{\mathsf{EL},1} = n_{\mathsf{EL},2} = \cdots = n_{\mathsf{EL},d}$ and $n_{\mathsf{DOF}} := n_{\mathsf{DOF},1} = n_{\mathsf{DOF},2} = \cdots = n_{\mathsf{DOF},d}$. We then have $N_{\mathsf{EL}} = n_{\mathsf{EL}}^d$, $N_{\mathsf{DOF}} = n_{\mathsf{DOF}}^d$, etc. We consider the case of maximum regularity $r = p - 1$ and $n_{\mathsf{EL}} \gg p$ that implies $n_{\mathsf{EL}} \approx n_{\mathsf{DOF}}$. We recall that:

$$\begin{aligned} \#\mathcal{K}_{l,i_l} &\leq p+1 \\ \#\mathcal{I}_{l,i_l} &\leq 2p+1 \end{aligned} \tag{1.146}$$

With the choice above for the quadrature points, the previous two imply $\#\mathcal{Q}_{l,i_l} \leq 2p+1$.

We first collect all the initialisations needed in Algorithm 1. It is not necessary to precompute these quantities -and in most architectures access to stored data is costly- but this used here for FLOPs evaluation.

```
   Input: Quadrature points $\widetilde{\boldsymbol{x}}_{\boldsymbol{q}}$ as in Remark 1.6.1
 1 for $l = 1, \dots, d$ do
 2     for $i_l = 1, \dots, n_{\mathsf{DOF},l}$ do
 3         Evaluate $\mathbb{B}^{(0)}_{l,i_l,q} := B_{i_l}(\widetilde{x}_{l,q})\,\forall q \in \mathcal{Q}_{l,i_l}$, store $\mathbb{B}^{(0)}_{l,i_l} \in \mathbb{R}^{n_{\mathsf{QP},l}}$ ;
 4         Evaluate $\mathbb{B}^{(1)}_{l,i_l,q} := B'_{i_l}(\widetilde{x}_{l,q})\,\forall q \in \mathcal{Q}_{l,i_l}$, store $\mathbb{B}^{(1)}_{l,i_l} \in \mathbb{R}^{n_{\mathsf{QP},l}}$ ;
 5     end
 6     for $i_l = 1, \dots, n_{\mathsf{DOF},l}$ do
 7         for $j_l \in \mathcal{I}_{l,i_l}$ do
 8             Calculate $\mathbb{I}^{(0,0)}_{l,i_l,j_l}$, $\mathbb{I}^{(1,0)}_{l,i_l,j_l}$, $\mathbb{I}^{(0,1)}_{l,i_l,j_l}$, $\mathbb{I}^{(1,1)}_{l,i_l,j_l}$ as defined in (1.141) ;
 9         end
10     end
11     for $m = 1, \dots, d$ do
12         Evaluate $c_{l,m}(\zeta)$ of equation (1.136) on points $\widetilde{\boldsymbol{x}}_{\boldsymbol{q}}$ ;
13     end
14 end
15 Evaluate $c(\zeta)$ on points $\widetilde{\boldsymbol{x}}_{\boldsymbol{q}}$;
```

Algorithm 1: Initializations

Then we can count operations in Algorithm 1:

(i) Evaluations of B-splines reported on lines 3–4 can be done in $\frac{1}{2}p^2$ FLOPs each. They are repeated $d\, n_{\mathsf{DOF}}\, \#\mathcal{Q}_{l,i_l}$ times so that this part costs $O\left(p^3\, n_{\mathsf{DOF}}\right)$ FLOPs.

(ii) The calculation of integrals (1.141) on line 8 needs to be done in an exact manner. The calculation of the exact integral of products of B-splines has a vast literature [**?**], however, closed forms are available only in some particular cases. For this reason we consider here the usual element-wise Gaussian quadrature. The evaluation of B-splines and their derivatives cost, as reported before, $\frac{1}{2}p^2$ FLOPs for each point. Counting all Gaussian point, the cost is $\approx d\, p^2$ evaluations of each of the $d\, n_{\mathsf{DOF}}$ univariate basis functions, thus costs $O(p^4\, n_{\mathsf{DOF}})$ FLOPs.
The computation of each of the integrals has the cost of a summation on $\approx p^2$ terms; and the four calculations are done $d\, n_{\mathsf{DOF}}\, \#\mathcal{I}_{l,i_l}$ times so that this costs $O\left(p^3\, n_{\mathsf{DOF}}\right)$ FLOPs.

(iii) The evaluations of the (d^2+1) functions $c_{l,m}$ and c on lines 12 and 15 have to be performed at the $N_{\mathsf{QP}} = (n_{\mathsf{QP}})^d$ quadrature points. The actual cost depends on the evaluation cost of $c_{l,m}$ and c. If these coefficients are obtained by $O(p^d)$ linear combinations of B-spline values (or derivatives), and each multivariate B-spline value is computed from multiplications of univariate B-spline values, the total cost is $C(d)p^d$ per quadrature point and in total $O\left(p^d\, N_{\mathsf{QP}}\right)$ FLOPs.

```
   Input: Quadrature points $\widetilde{\boldsymbol{x}}_{\boldsymbol{q}}$, B-spline evaluations $\mathbb{B}^{(\cdot)}_{l,i_l,\boldsymbol{q}}$, Integrals $\mathbb{I}^{(\cdot,\cdot)}_{l,i_l,j_l}$
1  for $l = 1, \ldots, d$ do
2      for $i_l = 1, \ldots, n_{l,\mathsf{DOF}}$ do
3          Calculate $w^{(0,0)}_{l,i_l,\mathcal{Q}_{l,i_l}}$ as (minimum Euclidean norm) solution of $\mathbb{B}^{(0)}_{l,\mathcal{I}_{l,i_l},\mathcal{Q}_{l,i_l}} w^{(0,0)}_{l,i_l,\mathcal{Q}_{l,i_l}} = \mathbb{I}^{(0,0)}_{l,i_l,\mathcal{I}_{l,i_l}}$ ;
4          Calculate $w^{(1,0)}_{l,i_l,\mathcal{Q}_{l,i_l}}$ as (minimum Euclidean norm) solution of $\mathbb{B}^{(1)}_{l,\mathcal{I}_{l,i_l},\mathcal{Q}_{l,i_l}} w^{(1,0)}_{l,i_l,\mathcal{Q}_{l,i_l}} = \mathbb{I}^{(1,0)}_{l,i_l,\mathcal{I}_{l,i_l}}$ ;
5          Calculate $w^{(0,1)}_{l,i_l,\mathcal{Q}_{l,i_l}}$ as (minimum Euclidean norm) solution of $\mathbb{B}^{(0)}_{l,\mathcal{I}_{l,i_l},\mathcal{Q}_{l,i_l}} w^{(0,1)}_{l,i_l,\mathcal{Q}_{l,i_l}} = \mathbb{I}^{(0,1)}_{l,i_l,\mathcal{I}_{l,i_l}}$ ;
6          Calculate $w^{(1,1)}_{l,i_l,\mathcal{Q}_{l,i_l}}$ as (minimum Euclidean norm) solution of $\mathbb{B}^{(1)}_{l,\mathcal{I}_{l,i_l},\mathcal{Q}_{l,i_l}} w^{(1,1)}_{l,i_l,\mathcal{Q}_{l,i_l}} = \mathbb{I}^{(1,1)}_{l,i_l,\mathcal{I}_{l,i_l}}$ ;
7      end
8  end
```

Algorithm 2: Construction of univariate WQ rules

The leading cost of Algorithm 1 for $d \geq 2$ is $O\left(p^d\, N_{\mathsf{QP}}\right)$.

In Algorithm 2 we summarize the operations needed for the construction of the univariate WQ rules. Each calculation in lines 3–6 consists in the resolution of a linear system of dimension $\approx (2p)^2$ that is possibly under-determined. The cost of these computations in any case negligible since it is proportional to n_{DOF}.

1.7 Formation of the mass matrix

When all the quadrature rules are available we can write the computation of the approximate mass matrix following (1.140). Similar formulae and algorithms can be written for the stiffness matrix starting from equation (1.132).

The mass matrix formation algorithm is mainly a loop over all rows $\boldsymbol{i}$, for each $\boldsymbol{i}$ we consider the calculation of

$$\widetilde{m}_{\boldsymbol{i},\boldsymbol{j}} = \mathbb{Q}^{(0,0)}_{\boldsymbol{i}} \left(\widehat{B}_{\boldsymbol{j}}(\boldsymbol{\zeta}) c(\boldsymbol{\zeta})\right), \quad \forall \boldsymbol{j} \in \mathcal{I}_{\boldsymbol{i}}. \tag{1.147}$$

The computational cost of (1.147) is minimised by a sum factorization approach, which is explained below.

If we substitute (1.142) into (1.140) we obtain the following sequence of nested summations:

$$\widetilde{m}_{\boldsymbol{i},\boldsymbol{j}} = \sum_{q_1 \in \mathcal{Q}_{1,i_1}} w^{(0,0)}_{1,i_1,q_1} \widehat{B}_{j_1}(\widetilde{x}_{1,q_1}) \left(\sum_{q_2 \in \mathcal{Q}_{2,i_2}} \cdots \sum_{q_d \in \mathcal{Q}_{d,i_d}} w^{(0,0)}_{d,i_d,q_d} \widehat{B}_{j_d}(\widetilde{x}_{d,q_d}) c\left(x_{1,q_1}, \ldots, x_{d,q_d}\right) \right) \tag{1.148}$$

To write (1.148) in a more compact form, we introduce the notion of matrix-tensor product [**?**]. Let $\mathcal{X} = \{x_{k_1,\ldots,k_d}\} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ be a $d-$dimensional tensor, and let $m \in \{1, \ldots, d\}$. The $m-$mode product of $\mathcal{X}$ with a matrix $A = \{a_{i,j}\} \in \mathbb{R}^{t \times n_m}$, denoted with $\mathcal{X} \times_m A$, is a tensor of dimension $n_1 \times \ldots \times n_{m-1} \times t \times n_{m+1} \times \ldots \times n_d$, with components

$$\left(\mathcal{X} \times_m A\right)_{k_1,\ldots,k_d} = \sum_{j=1}^{n_m} a_{k_m,j}\; x_{k_1,\ldots k_{m-1},j,k_{m+1},\ldots k_d}.$$

We emphasize that such computation requires $2t \prod_{l=1}^{d} n_l$ FLOPs.

For $l = 1, \ldots, d$ and $i_l = 1, \ldots, n_{\mathsf{DOF},l}$ we define the matrices

$$\mathrm{B}^{(l,i_l)} = \left(\widehat{B}_{j_l}(x_{l,q_l})\right)_{j_l \in \mathcal{I}_{l,i_l}, q_l \in \mathcal{Q}_{l,i_l}}, \qquad \mathrm{W}^{(l,i_l)} = \mathrm{diag}\left(\left(w^{(0,0)}_{l,i_l,q_l}\right)_{q_l \in \mathcal{Q}_{l,i_l}}\right),$$

where diag(v) denotes the diagonal matrix obtained by the vector v. We also define, for each index $\boldsymbol{i}$, the $d-$dimensional tensor

$$\mathcal{C}_{\boldsymbol{i}} = c(\widetilde{\boldsymbol{x}}_{\mathcal{Q}_{\boldsymbol{i}}}) = \left(c(\widetilde{x}_{1,q_1}, \dots, \widetilde{x}_{d,q_d})\right)_{q_1 \in \mathcal{Q}_{1,i_1}, \dots, q_d \in \mathcal{Q}_{d,i_d}}$$

Using the above notations, we have

$$\widetilde{m}_{\boldsymbol{i},\mathcal{I}_{\boldsymbol{i}}} = \mathcal{C}_{\boldsymbol{i}} \times_d \left(\mathrm{B}^{(d,i_d)}\mathrm{W}^{(d,i_d)}\right) \times_{d-1} \dots \times_1 \left(\mathrm{B}^{(1,i_1)}\mathrm{W}^{(1,i_1)}\right). \tag{1.149}$$

Since with this choice of the quadrature points $\#\mathcal{Q}_{l,i_l}$ and $\#\mathcal{I}_{l,i_l}$ are both $O(p)$, the computational cost associated with (1.149) is $O(p^{d+1})$ FLOPs. Note that $\widetilde{m}_{\boldsymbol{i},\mathcal{I}_{\boldsymbol{i}}}$ includes all the nonzeros entries of the $\boldsymbol{i}-$th row of $\widetilde{\mathbb{M}}$. Hence if we compute it for each $\boldsymbol{i} = 1, \dots, N_{\mathsf{DOF}}$ the total cost amounts to $O(N_{\mathsf{DOF}}\, p^{d+1})$ FLOPs. This approach is summarized in Algorithm 3.

We remark that writing the sums in (1.148) in terms of matrix-tensor products as in (1.149) is very useful from an implementation viewpoint. Indeed, in interpreted languages like MATLAB (which is the one used in the experiments of the next section), it is crucial to avoid loops and vectorize/tensorize the operations, in order to obtain an efficient implementation of an algorithm; see also the discussion in [**?**]. In particular, each matrix-tensor product in (1.149) is computed via a simple matrix-matrix product, which is a BLAS level 3 operation and typically yields high efficiency on modern computers.

```
Input: Quadrature rules, evaluations of coefficients
for $\boldsymbol{i} = 1, \dots, N_{\mathsf{DOF}}$ do
    Set $\mathcal{C}_{\boldsymbol{i}}^{(0)} := c(\widetilde{\boldsymbol{x}}_{\mathcal{Q}_{\boldsymbol{i}}})$;
    for $l = d, d-1, \dots, 1$ do
        Load the quadrature rule $\mathbb{Q}_{i_l}^{(0,0)}$ and form the matrices $\mathrm{B}^{(l,i_l)}$ and $\mathrm{W}^{(l,i_l)}$;
        Compute $\mathcal{C}_{\boldsymbol{i}}^{(d+1-l)} = \mathcal{C}_{\boldsymbol{i}}^{(d-l)} \times_l \left(\mathrm{B}^{(l,i_l)}\mathrm{W}^{(l,i_l)}\right)$;
    end
    Store $\widetilde{m}_{\boldsymbol{i},\mathcal{I}_{\boldsymbol{i}}} = \mathcal{C}_{\boldsymbol{i}}^{(d)}$;
end
```

Algorithm 3: Construction of mass matrix by sum-factorization

In order to evaluate numerically the behavior of the proposed procedure we present some numerical tests. First, in Section 1.7.1 we consider the solution of a 1D problem where we see that the application of the row-loop WQ-based algorithm leads to optimal order of convergence. Then, in Section **??** we measure the performance of the algorithm. We consider there the formation of mass matrices in 3D. The results for all cases refers to a Linux workstation equipped with Intel i7-5820K processors running at 3.30GHz, and with 64 GB of RAM. The row-loop WQ-based algorithm is potentially better suited for a parallel implementation than the standart element-wise SGQ-based algorithm, however we benchmark here sequential execution and use only one core for the simulations.

1.7.1 Convergence of approximate solution in 1D

As a test with known solution we consider the following:

$$\begin{cases} u'' + u(x) = \dfrac{5\exp(2x) - 1}{4} \quad \text{on} \qquad [0, \pi/6] \\ u(0) = 0, u(\pi/6) = \exp(\pi/3)/2 \end{cases} \tag{1.150}$$

We compare the numerical solution in the parametric domain, using the geometric transformation $t = 2sin(x)$, with the exact one $u(x) = \frac{\exp(2x)-1}{4}$. Then we calculate point-wise absolute error, integral error and energy error - namely L^∞, L^2 and H^1 norms - with varying spline degree p. Figure 1.30 illustrates that the construction of the matrices with the proposed procedure does not effect the overall convergence properties of the Galerkin method, as it can be seen by comparing the convergence curves with those obtained using Gaussian quadrature.

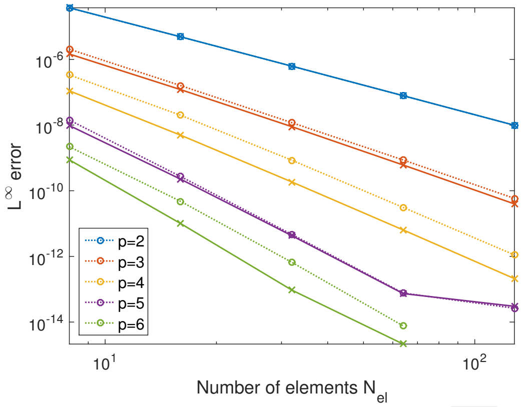


Fig. 1.30 Convergence history plot. We report errors in L^∞ norm for the solution of problem (1.150) by Galerkin based isogeometric analysis with WQ (Algorithm 1–3) for various degrees p in dotted lines. As reference, the solid lines refer to the same calculation made with element-wise SGQ. Optimal convergence rate is achieved in all cases. SGQ is sligtly more accurate for even degrees > 2 in L^2 and L^∞ norms.

We then report CPU time results for the formation on a single patch domain of mass matrices. Comparison is made with GeoPDEs, the optimized but SGQ-based MATLAB isogeometric library developed by Rafael Vázquez, see [113**?**]. In Figure 1.31 we plot the time needed for the mass matrix formation up to degree $p = 10$ with $N_{\mathsf{DOF}} = 20^3$. The tests confirm the superior performance of the proposed row-loop WQ-based algorithm vs SGQ. In the case $p = 10$ GeoPDEs takes more than 62 hours to form the mass matrix while the proposed algorithm needs only 27 sec, so that the use high degrees is possible with WQ.

In the last test, we experimentally study the growth order of the computational effort needed to form $\widetilde{\mathbb{M}}$, and we highlight which parts of the code mainly contributes to this effort.

In Figure 1.32, we plot in a log-log scale the total computation time spent by Algorithm 1–3 for 40^3 elements and spline degree up to 10. We also plot the time spent in the computations of the matrix-tensor products (i.e., line 5 of Algorithm 3, which is the dominant step with respect to the number of FLOPs of the whole procedure), and the time used by the MATLAB function `sparse`, which is responsible of allocating the memory for $\widetilde{\mathbb{M}}$ and copying the entries in the sparse matrix data structure. These timings were obtained using the profiler of MATLAB[3]. If we consider the products time, we can see that the its growth relative to p is significantly milder than what is indicated by the theoretical FLOP counting, i.e., $O(N_{\mathsf{DOF}}\, p^4)$. This is probably related to the small dimension of the matrices and tensors involved. On the other hand, the times spent by the `sparse` function is clearly proportional to p^3, as highlighted in the plot by a reference triangle with slope 3. This is expected, as the number of nonzero entries of $\widetilde{\mathbb{M}}$ is $O(N_{\mathsf{DOF}}\, p^3)$. What is surprising is that, for $p > 5$ the time of the `sparse` call dominates the total time of the algorithm. This indicates that the approach of [40] is in practice giving the best possible performance at least for degree high enough, since the `sparse` call is unavoidable and well optimised in MATLAB.

Furthermore, the computing time depends linearly on N_{DOF}, as expected, but for brevity we do not show the results.

Parallelization is another key ingredient to boost performance (see [75] for details).

[3] The same timings were also computed using the commands `tic` and `toc`, yielding similar results

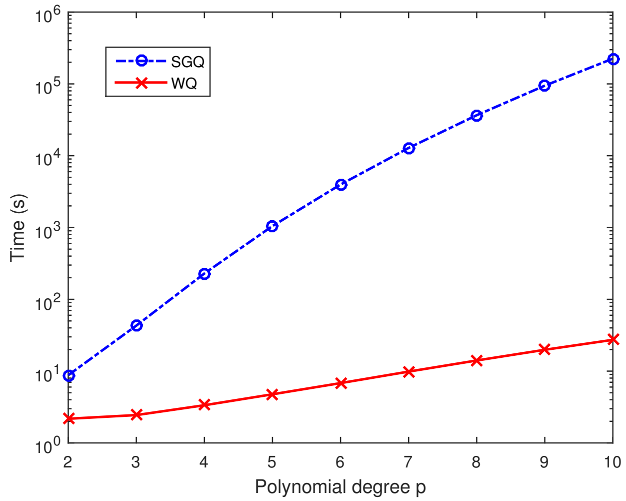


Fig. 1.31 Time for mass matrix assembly in the framework of isogeometric-Galerkin method with maximal regularity on a single patch domain of 20^3 elements. The comparison is between the WQ approach proposed (Algorithm 1–3) and SGQ as implemented in GeoPDEs 3.0.

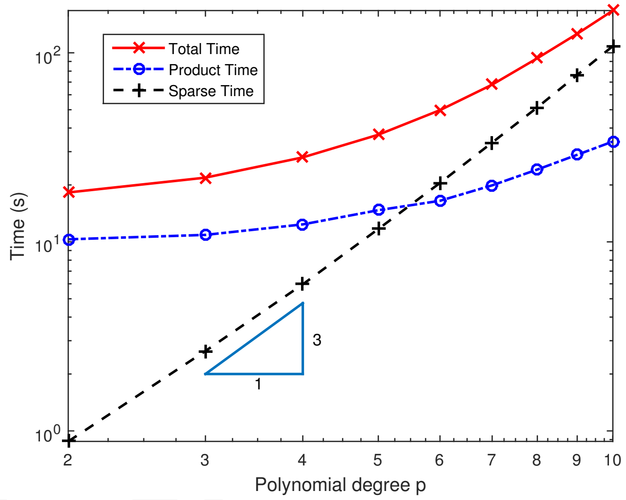


Fig. 1.32 Time for mass matrix formation with the WQ approach proposed in this paper. In this case $d = 3$, $n_{el} = 40^3$. Reference slope is p^3. Along with the total time, we show the time spent by the product (1.149) and by the function `sparse`, which represent the single most relevant computational efforts of the code. Other timings, which become negligible for large p, are not shown.

1.7.2 Linear solvers and preconditioners

The study of the computational efficiency of linear solvers for isogeometric discretization has been initiated in the papers [44, 45], where it has been shown that the algorithms used with the finite element method experiences performance degradation when used to solve isogeometric linear systems. Consider, for example, a Lagrangian finite element method with polynomial degree p and N degrees-of-freedom, in 3D: the linear system matrix $\mathcal{A}$ has a storage cost of $O(Np^3)$ non-zero terms and a solving cost by a direct solver of $O(N^2)$ FLOPs, (see [45, Section 2.3], under the assumption $N > p^6$). If, instead, we consider the isogeometric k-method with C^{p-1} p-degree splines and N degrees-of-freedom, still the linear system matrix $\mathcal{A}$ has $O(Np^3)$ non-zero entries but a standard direct solver costs $O(N^2p^3)$ FLOPs, i.e., p^3 times more than a finite element approximation.

Iterative solvers have attracted more attention in the isogeometric community. The effort has been primarily on the development of preconditioners for the Poisson model problem, for arbitrary degree and continuity splines. As reported in [44], standard algebraic preconditioners (Jacobi, SSOR, incomplete factorization) commonly adopted for finite elements exhibit reduced performance when used in the context of the isogeometric k-method. Standard multilevel and multigrid approaches are studied respectively in [37] and [59], while advances in the theory of domain-decomposition based solvers are given in, e.g., [49], [24], [28]. These papers also confirm the difficulty in achieving both robustness and computational efficiency for the high-degree k-method.

More sophisticated multigrid preconditioners have been proposed in the recent papers [53] and [66]. The latter, in particular, contains a proof of robustness, based on the theory of [106]. The two works are based on the following common ingredients: specific spectral properties of the discrete operator of the isogeometric k-method and the tensor-product structure of isogeometric spaces.

The tensor-product structure of multivariate spline space is exploited in [60, 91], based on approaches that have been developed for the so-called Sylvester equation. The tensor product structure of splines spaces yields to a Kronecker sctructure of isogeometric matrices.

We first recall the notation and basic properties of the Kronecker product of matrices. Let $A \in \mathbb{R}^{n_a \times n_a}$, and $B \in \mathbb{R}^{n_b \times n_b}$. The Kronecker product between A and B is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n_a}B \\ \vdots & \ddots & \vdots \\ a_{n_a1}B & \dots & a_{n_an_a}B \end{bmatrix} \in \mathbb{R}^{n_an_b \times n_an_b},$$

where a_{ij}, $i, j = 1, \dots n_a$, denote the entries of A. The Kronecker product is an associative operation, and it is bilinear with respect to matrix sum and scalar multiplication. Some properties of the Kronecker product that will be useful in the following.

- It holds
$$(A \otimes B)^T = A^T \otimes B^T. \tag{1.151}$$
- If C and D are matrices of conforming order, then
$$(A \otimes B)\,(C \otimes D) = (AC \otimes BD). \tag{1.152}$$
- If A and B are nonsingular, then
$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}. \tag{1.153}$$
- If λ_i, $i = 1, \dots, n_a$, denote the eigenvalues of A and μ_b, $j = 1, \dots, n_2$, denote the eigenvalues of B, then the n_an_b eigenvalues of $A \otimes B$ have the form
$$\lambda_i\mu_j, \qquad i = 1, \dots, n_a,\ j = 1, \dots, n_b. \tag{1.154}$$

Property (1.151) implies that if A and B are both symmetric, then $A \otimes B$ is also symmetric. Moreover, if A and B are both positive definite, then according to (1.154) $A \otimes B$ is also positive definite.

For any matrix $X \in \mathbb{R}^{n_a \times n_b}$ we denote with $\mathrm{vec}(X)$ the vector of $\mathbb{R}^{n_a n_b}$ obtained by "stacking" the columns of X. Then if A, B and X are matrices of conforming order, and $x = \mathrm{vec}(X)$, it holds

$$(A \otimes B)\, x = \mathrm{vec}(BXA^T). \tag{1.155}$$

This property can be used to cheaply compute matrix-vector products with a matrix having Kronecker structure. Indeed, it shows that computing $(A \otimes B)\, x$ is equivalent to computing n_b matrix-vector products with A and n_a matrix-vector products with B. Note in particular that $(A \otimes B)$ does not have to be formed. If A and B are nonsingular, then (1.155) is equivalent to

$$(A \otimes B)^{-1}\, x = \mathrm{vec}(A^{-1}XB^{-T}), \tag{1.156}$$

which, in a similar way, shows that the problem of solving a linear system having $(A \otimes B)$ as coefficient matrix is equivalent to solve n_b linear systems involving A and n_a linear systems involving B.

Consider the Laplace operator with constant coefficients, on the square $[0,1]^2$, then the tensor-product spline Galerkin discretization leads to the system

$$(K_1 \otimes M_2 + M_1 \otimes K_2)u = b \tag{1.157}$$

where K_ℓ and M_ℓ denote the univariate stiffness and mass matrices in the ℓ direction, $\ell = 1, 2$, and $\otimes$ is the Kronecker product. Equation (1.157), when reformulated as a matrix equation, takes the form of a (generalized) Sylvester equation. This is a well studied problem in the numerical linear algebra literature (see the recent survey [104]). Two popular algorithms to solve the Sylvester equation are the direct solver proposed by Bartels and Stewart in [10] and the alternating direction implicit (ADI) iterative solver, first introduced in [85] and further developed in a number of papers, among which is [117]. These approaches are benchmarked, in the isogeometric setting, in [91].

Observe that in general, for variable coefficients, general elliptic problems, non-trivial and possibly multipatch geometry parametrization, the isogeometric system is not as in (1.157). In this case, a fast solver for (1.157) plays the role of a preconditioner. At each iterative step, the preconditioner takes the form

$$(K_1 \otimes M_2 + M_1 \otimes K_2)\, s = r. \tag{1.158}$$

Using relation (1.155), we can rewrite this equation in matrix form

$$M_2SK_1 + K_2SM_1 = R, \tag{1.159}$$

where $\mathrm{vec}(S) = s$ and $\mathrm{vec}(R) = r$. Equation (1.159) takes the name of (generalized) Sylvester equation. Due to its many applications, the literature dealing with Sylvester equation (and its variants) is vast, and a number of methods have been proposed for its numerical solution. We refer to [104] for a recent survey on this subject.

In the last two decades, the research on Sylvester equation has mainly focused on methods which require that the right-hand side matrix R has low rank. Such methods are nor considered in this work. Indeed, even if there are cases where R is low-rank or can be approximated efficiently by a low-rank matrix, this is not the general case. Furthermore, and perhaps more important, low-rank methods are designed for solving very large problems, where even storing the solution S might be unfeasible. This is not the case of problems of practical interest in isogeometric analysis.

In this paper, we consider two among the most studied methods, which in the authors' perspective seem the most suited for the particular features of IGA problems. The fast diagonalization (FD) method is a direct solver, which means that $s = \mathcal{P}^{-1}r$ is computed exactly. The alternating direction implicit (ADI) method is an iterative solver, which means that s is computed only approximately. We remark that ADI was first applied to IGA problems in [60].

To keep the notation consistent with the rest of the paper, in this section we will favor the Kronecker formulation (1.158) with respect to the matrix equation form (1.159).

We describe a direct method for (1.158) that was first presented in 1964 by Lynch, Rice and Thomas [**?**] as a method for solving elliptic partial differential equations discretized with finite differences. Following [**?**], we refer to it as the fast diagonalization (FD) method. We remark that this approach was extended to a general Sylvester equation involving nonsymmetric matrices by Bartels and Stewart in 1972 [10], although this is not considered here.

We consider the generalized eigendecomposition of the matrix pencils (K_1, M_1) and (K_2, M_2), namely

$$K_1 U_1 = M_1 U_1 D_1 \qquad K_2 U_2 = M_2 U_2 D_2, \tag{1.160}$$

where D_1 and D_2 are diagonal matrices whose entries are the eigenvalues of $M_1^{-1} K_1$ and $M_2^{-1} K_2$, respectively, while U_1 and U_2 satisfy

$$U_1^T M_1 U_1 = I, \qquad U_2^T M_2 U_2 = I,$$

which implies in particular $U_1^{-T} U_1^{-1} = M_1$ and $U_2^{-T} U_2^{-1} = M_2$, and also, from (1.160), $U_1^{-T} D_1 U_1^{-1} = K_1$ and $U_2^{-T} D_2 U_2^{-1} = K_2$. Therefore we factorize $\mathcal{P}$ in (1.158) as follows:

$$(U_1 \otimes U_2)^{-T} (D_1 \otimes I + I \otimes D_2) (U_1 \otimes U_2)^{-1} s = r,$$

and adopt the following strategy:

- Compute the generalized eigendecompositions (1.160)
- Compute $\widetilde{r} = (U_1 \otimes U_2)^T r$
- Compute $\widetilde{s} = (D_1 \otimes I + I \otimes D_2)^{-1} \widetilde{r}$
- Compute $s = (U_1 \otimes U_2) \widetilde{s}$

Algorithm 4: FD direct method (2D)

The exact cost of the eigendecompositions in line 1 depends on the algorithm employed. A simple approach is to first compute the Cholesky factorization $M_1 = LL^T$ and the symmetric matrix $\widetilde{K_1} = L^{-1} K_1 L^{-T}$. Since M_1 and K_1 are banded, the cost of these computations is $\mathrm{O}(pn^2)$ FLOPs. The eigenvalues of $\widetilde{K_1}$ are the same of (1.160), and once the matrix $\widetilde{U_1}$ of orthonormal eigenvectors is computed then one can compute $U_1 = L^{-T} \widetilde{U_1}$, again at the cost of $\mathrm{O}(pn^2)$ FLOPs. Being $\widetilde{U_1}$ orthogonal, then $U_1^T M_1 U_1 = I_n$. If the eigendecomposition of $\widetilde{K_1}$ is computed using a divide-and-conquer method, the cost of this operation is roughly $4n^3$ FLOPs. We remark that the divide-and-conquer approach is also very suited for parallelization. In conclusion, by this approach, line 1 requires roughly $8n^3$ FLOPs.

Lines 2 and 4 each involve a matrix-vector product with a matrix having Kronecker structure, and each step is equivalent (see (1.155)) to 2 matrix products involving dense $n \times n$ matrices. The total computational cost of both steps is $8n^3$ FLOPs. Line 3 is just a diagonal scaling, and its $O(n^2)$ cost is negligible. We emphasize that the overall computational cost of Algorithm 4 is independent of p.

If we apply Algorithm 4 as a preconditioner, then Step 1 may be performed only once, since the matrices involved do not change throughout the CG iteration. In this case the main cost can be quantified in approximately $8n^3$ FLOPs per CG iteration. The other main computational effort of each CG iteration is the residual computation, that is the product of the system matrix $\mathcal{A}$ by a vector, whose cost in FLOPs is twice the number of nonzero entries of $\mathcal{A}$, that is approximately $2(2p+1)^2 n^2$. In conclusion, the cost ratio between the preconditioner application and the residual computation is about $4n/(2p+1)^2 \approx n/p^2$.

The ultimate goal is the solution of 3D isogeometric systems, especially when high resolution is needed. On a unit cube, still considering the Poisson problem, thanks to the tensor-product structure the linear system takes the form

$$(K_1 \otimes M_2 \otimes M_3 + M_1 \otimes K_2 \otimes M_3 + M_1 \otimes M_2 \otimes K_3)u = b.$$

In this case, extensions of Bartels-Stewart and ADI methods can efficiently solve such problems [91].

When $d = 3$, equation (**??**) takes the form

$$(K_1 \otimes M_2 \otimes M_3 + M_1 \otimes K_2 \otimes M_3 + M_1 \otimes M_2 \otimes K_3)\, s = r. \tag{1.161}$$

The direct method above admits a straightforward generalization to the 3D case. We consider the generalized eigendecompositions

$$K_1 U_1 = M_1 U_1 D_1, \qquad K_2 U_2 = M_2 U_2 D_2, \qquad K_3 U_3 = M_3 U_3 D_3, \tag{1.162}$$

with D_1, D_2, D_3 diagonal matrices and

$$U_1^T M_1 U_1 = I, \qquad U_2^T M_2 U_2 = I, \qquad U_3^T M_3 U_3 = I.$$

Then, (1.161) can be factorized as

$$(U_1 \otimes U_2 \otimes U_3)^{-1} (D_1 \otimes I \otimes I + I \otimes D_2 \otimes I + I \otimes I \otimes D_3) (U_1 \otimes U_2 \otimes U_3)^{-T} s = r,$$

which suggests the following algorithm.

```
• Compute the generalized eigendecompositions (1.162)
• Compute $\widetilde{r} = (U_1 \otimes U_2 \otimes U_3) r$
• Compute $\widetilde{s} = (D_1 \otimes I \otimes I + I \otimes D_2 \otimes I + I \otimes I \otimes D_3)^{-1} \widetilde{r}$
• Compute $s = (U_1 \otimes U_2 \otimes U_3)^T \widetilde{s}$
```

Algorithm 5: FD direct method (3D)

Lines 1 and 3 require $O(n^3)$ FLOPs. Lines 2 and 4, as can be seen by nested applications of formula (1.155), are equivalent to performing a total of 6 products between dense matrices of size $n \times n$ and $n \times n^2$. Thus, neglecting lower order terms the overall computational cost of Algorithm 5 is $12n^4$ FLOPs.

The direct method is even more appealing in the 3D case than it was in the 2D case, for at least two reasons. First, the computational cost associated with the preconditioner setup, that is the eigendecomposition, is negligible. This means that the main computational effort of the method consists in a few (dense) matrix-matrix products, which are level 3 BLAS operations and typically yield high efficiency thanks to a dedicated implementation on modern computers by optimized usage of the memory cache hierarchy Second, in a preconditioned CG iteration the cost for applying the preconditioner has to be compared with the cost of the residual computation (a matrix-vector product with $\mathcal{A}$) which can be quantified in approximately $2(2p+1)^3 n^3$ for 3D problems, resulting in a FLOPs ratio of the preconditioner application to residual computation of $(3n)/(4p^3) \approx n/p^3$. For example, if $N = 256^3$ and $p = 4$, the preconditioner requires only 3 times more FLOPs than the residual computation, while for degree $p = 6$ the matrix-vector product is even more costly than the preconditioner itself. However in numerical tests we will see that, for all cases of practical interest in 3D, the computational time used by the preconditioner application is far lower that the residual computation itself. This is because the computational time depends not only on the FLOPs count but also on the memory usage and, as mentioned above, dense matrix-matrix multiplications greatly benefit of modern computer architecture. This approach will show largely higher performance than the alternative ADI approach we have considered in [91].

We repor some 3D single-patch numerical tests from [91].

We consider a two domains: the first one is a thick quarter of ring; note that this solid has a trivial geometry on the third direction. The second one is the solid of revolution obtained by the 2D quarter of ring. Specifically, we performed a $\pi/2$ revolution around the axis having direction $(0, 1, 0)$ and passing through $(-1, -1, -1)$. We emphasize that here the geometry is nontrivial along all directions.

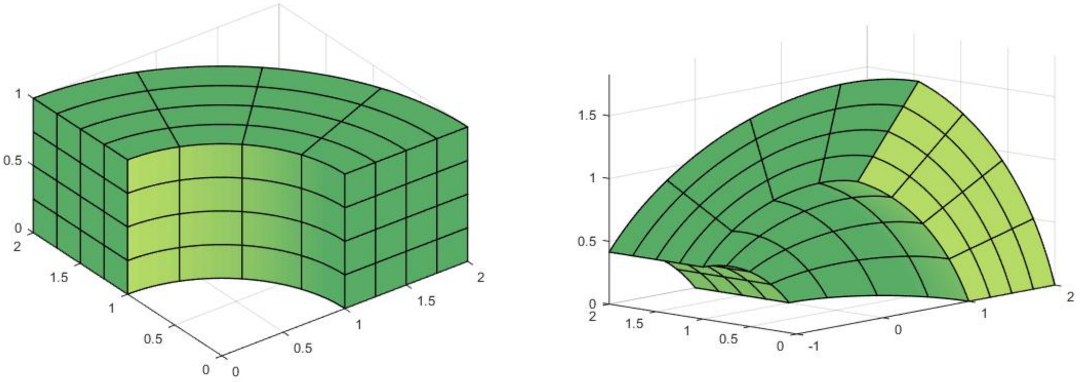


Fig. 1.33 Thick ring and revolved ring domains

We consider a standard Incomplete Cholesky (IC) preconditioner (no reordering is used in this case, as the resulting performance is better than when using the standard reorderings available in MATLAB).

In Table 1.1 we report the results for the thick quarter ring while in Table 1.2 we report the results for the revolved ring. The symbol "*" denotes the cases in which even assembling the system matrix $\mathcal{A}$ was unfeasible due to memory limitations. From these results, we infer that most of the conclusions drawn for the 2D case still hold in 3D. In particular, both Sylvester-based preconditioners yield a better performance than the IC preconditioner, especially for small h.

	CG + $\mathcal{P}$ Iterations / Time (sec)				
h^{-1}	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
32	26 / 0.19	26 / 0.38	26 / 0.75	26 / 1.51	26 / 2.64
64	27 / 1.43	27 / 3.35	27 / 6.59	27 / 12.75	27 / 21.83
128	28 / 14.14	28 / 32.01	28 / 61.22	*	*

	CG + $\mathcal{P}_J$ Iterations / Time (sec)				
h^{-1}	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
32	26 (7) / 0.88	26 (7) / 1.20	26 (7) / 1.71	26 (7) / 2.62	27 (8) / 4.08
64	27 (7) / 7.20	27 (8) / 10.98	27 (8) / 14.89	27 (8) / 21.81	27 (8) / 30.56
128	28 (8) / 99.01	28 (8) / 98.39	28 (8) / 143.45	*	*

	CG + IC Iterations / Time (sec)				
h^{-1}	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
32	21 / 0.37	15 / 1.17	12 / 3.41	10 / 9.43	9 / 24.05
64	37 / 4.26	28 / 13.23	22 / 33.96	18 / 88.94	16 / 215.31
128	73 / 65.03	51 / 163.48	41 / 385.54	*	*

Table 1.1 Thick quarter of ring domain. Performance of CG preconditioned by the direct method (upper table), by ADI (middle table) and by Incomplete Cholesky (lower table).

	CG + $\mathcal{P}$ Iterations / Time (sec)				
h^{-1}	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
32	40 / 0.27	41 / 0.63	41 / 1.24	42 / 2.38	42 / 4.13
64	44 / 2.30	44 / 5.09	45 / 10.75	45 / 20.69	45 / 35.11
128	47 / 23.26	47 / 55.34	47 / 101.94	*	*

	CG + $\mathcal{P}_J$ Iterations / Time (sec)				
h^{-1}	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
32	40 (7) / 1.39	41 (7) / 1.93	41 (7) / 2.67	42 (7) / 4.17	42 (8) / 6.25
64	44 (7) / 11.82	44 (8) / 16.96	45 (8) / 24.31	45 (8) / 35.76	45 (8) / 49.89
128	47 (8) / 170.69	47 (8) / 168.45	47 (9) / 239.07	*	*

	CG + IC Iterations / Time (sec)				
h^{-1}	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
32	24 / 0.44	18 / 1.28	15 / 3.61	12 / 9.63	11 / 24.57
64	47 / 5.19	35 / 14.95	28 / 37.33	24 / 94.08	20 / 222.09
128	94 / 81.65	71 / 211.53	57 / 464.84	*	*

Table 1.2 Revolved quarter of ring domain. Performance of CG preconditioned by the direct method (upper table), by ADI (middle table) and by Incomplete Cholesky (lower table).

Somewhat surprisingly, however, the CPU times show a stronger dependence on p than in the 2D case, and the performance gap between the ADI and the FD approach is not as large as for the cube domain. This is due to the cost of the residual computation in the CG iteration (a sparse matrix-vector product, costing $O(p^3n^3)$ FLOPs). This step represents now a significant computational effort in the overall CG performance. In fact, our numerical experience shows that the 3D FD method is so efficient that the time spent in the preconditioning step is often negligible w.r.t. the time required for the residual computation. This effect is clearly shown in Table 1.3, where we we report the percentage of time spent in the application of the preconditioner when compared with the overall time of CG, in the case of the revolved ring domain. Interestingly, this percentage is almost constant w.r.t. h up to the finest discretization level, corresponding to about 2 million degrees-of-freedom.

h^{-1}	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
32	25.60	13.34	7.40	4.16	2.44
64	22.69	11.26	5.84	3.32	1.88
128	25.64	13.09	6.92	*	*

Table 1.3 Percentage of time spent in the application of the 3D FD preconditioner with respect to the overall CG time. Revolved ring domain.

To enhance flexibility in geometry representation, typically multi-patch parametrizations are adopted in isogeometric analysis. This means that the domain of interest $\varOmega$ is the union of patches $\varOmega_i$ such that $\varOmega_i = \mathbf{F}_i([0,1]^d)$, and each $\mathbf{F}_i$ is a spline (or NURBS) parametrization. Typically $\varOmega_i \cap \varOmega_j$ is an empty set, or a vertex, or the full common edge or the full common face (when $d = 3$) of the patches. Furthermore we assume that the meshes are conforming, that is for each patch interface $\varOmega_i \cap \varOmega_j$ the isogeometric functions on $\varOmega_i$ and the ones on $\varOmega_j$ generate the same trace space. See, e.g., [**?**] for more details.

For such a configuration, we can easily combine the approaches discussed in the previous sections with an overlapping Schwarz preconditioner. For that, we need to further split $\varOmega$ into overlapping subdomains. We choose the subdomains as pairs of neighboring patches merged together. Precisely, let N_s denotes the total number of interfaces between neighboring patches. We define

$$\varTheta_i = \varOmega_{i_1} \cup \varOmega_{i_2}, \qquad i = 1, \dots, N_s,$$

where Ω_{i_1} and Ω_{i_2} are the patches which share the $i-$th interface. It holds $\Omega = \bigcup_i \Theta_i$.

Now let R_i be the rectangular restriction matrix on the degrees-of-freedom associated with the $i-$th subdomain, and let $\mathcal{A}_i = R_i \mathcal{A} R_i^T$.The (exact) additive Schwarz preconditioner is

$$\mathcal{P}_{EAS} = \sum_{i=1}^{N_s} R_i^T \mathcal{A}_i^{-1} R_i, \tag{1.163}$$

and its inexact variant

$$\mathcal{P}_{IAS} = \sum_{i=1}^{N_s} R_i^T \widetilde{\mathcal{A}}_i^{-1} R_i, \tag{1.164}$$

where each $\widetilde{A}_i^{-1}$ is a suitable approximation of $\mathcal{A}_i^{-1}$. Each $\mathcal{A}_i$ represents the system matrix of a discretized Poisson problem on Θ_i. A crucial observation is that, under the conforming assumption, Θ_i can be considered a single-patch domain. Thus, it is possible to construct a preconditioner of the form (**??**) for $\mathcal{A}_i$, which we denote with $\mathcal{P}_i$. Then $\mathcal{P}_i$ can be used to construct $\widetilde{\mathcal{A}}_i^{-1}$ (we can have $\widetilde{\mathcal{A}}_i = \mathcal{P}_i$, or $\widetilde{\mathcal{A}}_i^{-1}$ may represent a fixed number of iteration of some iterative method preconditioned by $\mathcal{P}_i$). The proposed approach is somewhat unusual in the context of domain decomposition methods. Indeed, it is more common to split the domain Ω into a large number of small subdomains, so that local problems can be efficiently solved by parallel architectures. Here, on the other hand, the subdomains are chosen so that the basis functions of the local problems have a tensor structure that can be exploited by our preconditioner. The efficiency of such preconditioners, which is demonstrated numerically in the next section, makes it feasible to work with local problems whose size is comparable with that of the whole system. Finally, we remark that the large overlap between neighboring subdomains ensures that the outer iteration converges fast, independently of p and h.

Extension of this approach to nonconforming discretizations would require the use of nonconforming DD preconditioners (e.g., [78]) instead of an overlapping Schwarz preconditioner.

1.8 Application examples

In this section we present some numerical benchmarks of model problems. The first example, from [14], is the one of linear elasticity. The second one, from [57], is a fluid benchmark and utilizes the divergence-free isogeometric vector fields defined in Section 1.3.2.

1.8.1 Linear elasticity

We start by considering the classical elliptic linear elastic problem. First we introduce some notation. The body occupies a two-dimensional domain $\Omega \subset \mathbb{R}^2$. We assume that the boundary $\partial\Omega$ is decomposed into a Dirichlet part Γ_D and a Neumann part Γ_N. Moreover, let $\mathbf{f} : \Omega \to \mathbb{R}^d$ be the given body force and $\mathbf{g} : \Omega : \Gamma_N \to \mathbb{R}^d$ the given traction on Γ_N.

Then, the mixed boundary-value problem reads

$$\begin{cases} \operatorname{div} \mathbb{C}\boldsymbol{\varepsilon}(\mathbf{u}) + \mathbf{f} = 0 & \text{in } \Omega \\ \mathbf{u} = 0 & \text{on } \Gamma_D \\ \mathbb{C}\boldsymbol{\varepsilon}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{g} & \text{on } \Gamma_N, \end{cases} \tag{1.165}$$

where $\mathbf{u}$ is the body displacement and $\boldsymbol{\varepsilon}(\mathbf{u})$ its symmetric gradient, $\mathbf{n}$ is the unit outward normal at each point of the boundary and the fourth-order tensor $\mathbb{C}$ satisfies

$$\mathbb{C}\mathbf{w} = 2\mu \left[\mathbf{w} + \frac{\nu}{1-2\nu} \mathrm{tr}(\mathbf{w})\mathbf{I} \right] \tag{1.166}$$

for all second-order tensors $\mathbf{w}$, where tr represents the trace operator and $\mu > 0,\ 0 \leq \nu < 1/2$ are, respectively, the shear modulus and Poisson's ratio. The stress, $\boldsymbol{\sigma}$, is given by Hooke's law, $\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon}$.

Assuming for simplicity a regular loading $\mathbf{f} \in [L^2(\Omega)]^2$ and $\mathbf{g} \in [L^2(\Gamma_N)]^2$, we introduce also

$$< \boldsymbol{\psi}, \mathbf{v} >= (\mathbf{f}, \mathbf{v})_\Omega + (\mathbf{g}, \mathbf{v})_{\Gamma_N} \qquad \forall \mathbf{v} \in [H^1(\Omega)]^d, \tag{1.167}$$

where $(\,,\,)_\Omega$, $(\,,\,)_{\Gamma_N}$ indicate, as usual, the L^2 scalar products on Ω and Γ_N, respectively. The variational form of problem (1.165) then reads: find $\mathbf{u} \in [H^1_{\Gamma_D}(\Omega)]^d$ such that

$$(\mathbb{C}\boldsymbol{\varepsilon}(\mathbf{u}), \boldsymbol{\varepsilon}(\mathbf{v}))_\Omega =< \boldsymbol{\psi}, \mathbf{v} > \qquad \forall \mathbf{v} \in [H^1_{\Gamma_D}(\Omega)]^d \tag{1.168}$$

To solve (1.165), we introduce an isogeometric vector space V_h as defined in Section 1.3.1 and look for the Galerkin isogeometric approximation $\mathbf{u}_h \in \boldsymbol{V}_h$ such that

$$(\mathbb{C}\boldsymbol{\varepsilon}(\mathbf{u}), \boldsymbol{\varepsilon}(\mathbf{v}))_\Omega =< \boldsymbol{\psi}, \mathbf{v} > \qquad \forall \mathbf{v} \in \boldsymbol{V}_h, \tag{1.169}$$

where

$$\boldsymbol{V}_h = [V_h]^d \cap [H^1_{\Gamma_D}(\Omega)]^d. \tag{1.170}$$

This is an elliptic problem, then a Galerkin method returns the best approximation in the energy norm. The order of convergence of the numerical error $\mathbf{u}-\mathbf{u}_h$ follows from the approximation properties of isogeometric spaces, see Section 1.4.

We will see this for the model of an infinite plate with a hole, modeled by a finite quarter plate. The exact solution (Gould[63], pp. 120-123), evaluated at the boundary of the finite quarter plate, is applied as a Neumann boundary condition. The setup is illustrated in Figure 1.34. T_x is the magnitude of the applied stress at infinity, R is the radius of the traction-free hole, L is the length of the finite quarter plate, E is Young's modulus, and ν is Poisson's ratio. The rational quadratic basis is the minimum order capable of exactly representing a circle.

The first six meshes used in the analysis are shown in Figure 1.35. The cubic and quartic NURBS are obtained by order elevation of the quadratic NURBS on the coarsest mesh (for details of the geometry and mesh construction, see [68]). Continuity of the basis is C^{p-1} everywhere, except along the line which joins the center of the circular edge with the upper left-hand corner of the domain. There it is C^1 as is dictated by the coarsest mesh employing rational quadratic parametrization. In this example, the geometry parametrization is singular at the the upper left-hand corner of the domain. Convergence results in the L^2-norm of stresses (which is equivalent to the H^1-seminorm of the displacements) are shown in Figure 1.36. As can be seen, the L^2-convergence rates of stress for quadratic, cubic, and quartic NURBS are 2, 3, and 4, respectively, hence optimal in all cases, consistent with the approximation estimates described in Section 1.4.

1.8.2 Steady Navier-Stokes Problem

We consider now the steady Navier-Stokes Problem. The fluid occupies the domain $\Omega \subset \mathbb{R}^3$. We assume that the boundary $\partial\Omega = \Gamma_D$ for simplicity and take $\mathbf{f} : \Omega \to \mathbb{R}^3$ as the external driving force. Then, the problem reads

$$\begin{cases} \mathrm{div}\,(\mathbf{u} \otimes \mathbf{u}) - \ \mathrm{div}\,(2\nu\boldsymbol{\varepsilon}(\mathbf{u})) + \nabla p = \mathbf{f} & \text{in } \Omega \\ \mathrm{div}\,\mathbf{u} = 0 & \text{in } \Omega \\ \mathbf{u} \cdot \mathbf{n} & \text{on } \partial\Omega, \end{cases} \tag{1.171}$$

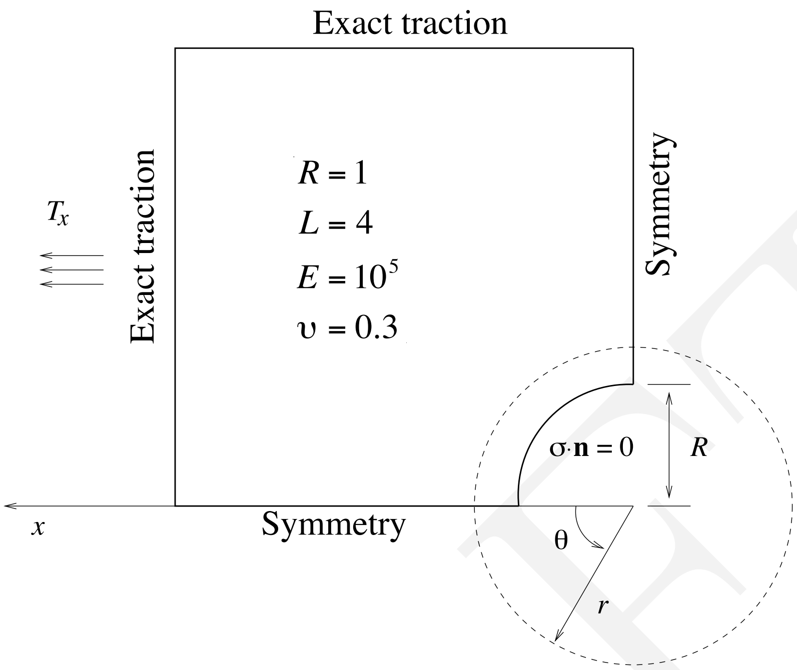


Fig. 1.34 Elastic plate with a circular hole: problem definition.

where $\mathbf{u}$ is the fluid velocity, p is the pressure, ν is the kinematic viscosity and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the symmetric gradient operator.

The variational form of (1.171) reads as follows: find $\mathbf{u} \in [H_0^1(\Omega)]^d$ and $p \in L_0^2(\Omega)$ such that

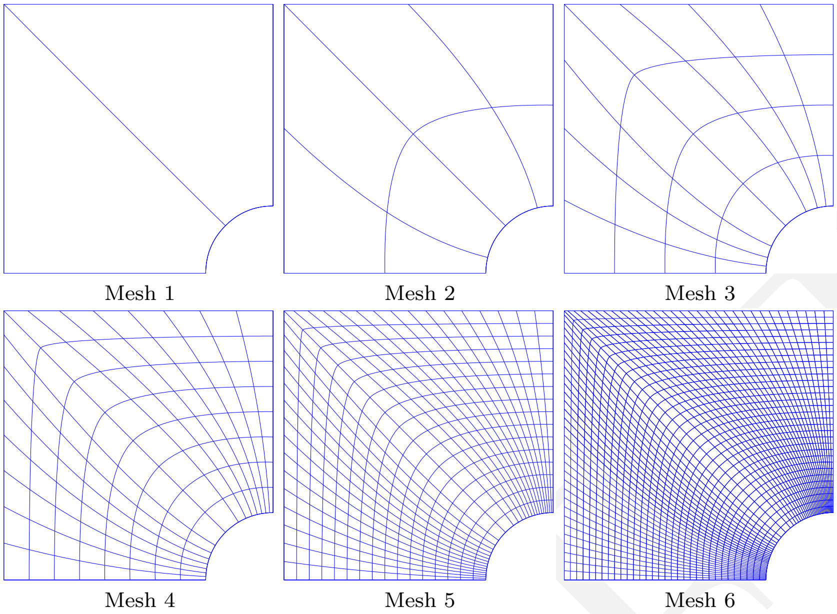
$$(2\nu\boldsymbol{\varepsilon}(\mathbf{u}), \boldsymbol{\varepsilon}(\mathbf{v}))_\Omega - (\mathbf{u} \otimes \mathbf{u}, \nabla\mathbf{v})_\Omega - (p, \operatorname{div}\mathbf{u})_\Omega + (q, \operatorname{div}\mathbf{v})_\Omega = (\mathbf{f}, \mathbf{v})_\Omega, \, \forall \mathbf{u} \in [H_0^1(\Omega)]^d, q \in L_0^2(\Omega), \tag{1.172}$$

where $L_0^2(\Omega)$ is the subspace of $L^2(\Omega)$ functions having zero average on Ω. At the discrete level, we are going to adopt a divergence-free (X_h^2, X_h^3) isogeometric discretization for the velocity-pressure pair, as defined in Section 1.3.2. In this case, only the Dirichlet boundary condition on the normal velocity component (i.e., no-penetration condition) can be imposed strongly (see [38]) while the other boundary conditions, including the Dirichlet boundary condition on the tangential velocity component, have to be imposed weakly, for example by Nitsche's method, as studied in [57]. For that, we introduce the space $\boldsymbol{H}_{\mathbf{n}}^1(\Omega) = \{\mathbf{w} \in [H^1(\Omega)]^d$ such that $\mathbf{w} \cdot \mathbf{n} = 0$ on $\partial\Omega\}$, and the discrete variational formulation is: find $\mathbf{u}_h \in X_h^2 \cap \boldsymbol{H}_{\mathbf{n}}^1(\Omega)$ and $p_h \in X_h^3 \cap L_0^2(\Omega)$ such that

$$\begin{aligned}
&(2\nu\boldsymbol{\varepsilon}(\mathbf{u}_h), \boldsymbol{\varepsilon}(\mathbf{v}_h))_\Omega - (\mathbf{u}_h \otimes \mathbf{u}_h, \nabla\mathbf{v}_h)_\Omega - (p_h, \operatorname{div}\mathbf{u}_h)_\Omega + (q_h, \operatorname{div}\mathbf{v}_h)_\Omega \\
&- \sum_{F \subset \partial\Omega} \int_F \left((\boldsymbol{\varepsilon}(\mathbf{u}_h)\,\mathbf{n}) \cdot \mathbf{v}_h + (\boldsymbol{\varepsilon}(\mathbf{v}_h)\,\mathbf{n}) \cdot \mathbf{u}_h - \frac{C_{pen}}{h_F}\mathbf{u}_h \cdot \mathbf{v}_h \right) d\boldsymbol{s} \\
&= (\mathbf{f}, \mathbf{v}_h)_\Omega, \qquad\qquad \forall \mathbf{v}_h \in X_h^2 \cap \boldsymbol{H}_{\mathbf{n}}^1(\Omega), q_h \in X_h^3 \cap L_0^2(\Omega),
\end{aligned} \tag{1.173}$$

where $F \subset \partial\Omega$ denote the faces (in three dimensions) of the Bézier elements that are on the boundary of Ω and $C_{pen} > 0$ is a suitable penalty constant.

We consider a simple configuration (see [57, Section 8.2]), with $\Omega = [0, 1]^3$ and select in (1.32) the polynomial degrees $p_1 = p_2 = p_3 = 2$ and $p_1 = p_2 = p_3 = 3$. Selecting knots with single multiplicity, the former choice X_h^2 is formed by linear-quadratic splines and X_h^3 is formed by trilinear splines, which is the minimum degree required to have $X_h^2 \in [H^1(\Omega)]^d$. The right-hand side is set up in order to give



Mesh 1

Mesh 2

Mesh 3

Mesh 4

Mesh 5

Mesh 6

Fig. 1.35 Elastic plate with circular hole. Meshes produced by h-refinement (knot insertion).

the exact solution:

$$\mathbf{u}_h = \mathbf{curl} \begin{bmatrix} x(x-1)y^2(y-1)^2z^2(z-1)^2 \\ 0 \\ x^2(x-1)^2y^2(y-1)^2z(z-1) \end{bmatrix}; \quad p = \sin(\pi x)\sin(\pi y) - \frac{4}{\pi^2}.$$

Streamlines associated with the exact solution are plotted in Figure 1.37. The convergence rates are shown in Figures 1.38-1.39 for Reynolds number $Re = 1$. Optimal convergence is obtained for both velocity and pressure. We remark that the discrete velocity is point-wise divergence-free, because of (1.38).

Acknowledgments

This chapter is a review of the mathematical results available on IGA. The presentation follows in particular [47], [18], [46], [70], [69], [91], [57]. We thank our colleagues and coauthors: Yuri Bazilevs, Lourenco Beirão da Veiga, Annalisa Buffa, Francesco Calabrò, Annabelle Collin, Austin Cottrell, John Evans, Alessandro Reali, Thomas Takacs, and Rafael Vázquez. Thomas J.R. Hughes is partially supported by the Office of Naval Research under contract number N00014-08-0992. Giancarlo Sangalli and Mattia Tani are partially supported by the European Research Council through the FP7 Ideas Starting Grant *HIGEOM*. This support is gratefully acknowledged.

References

[1] C. Adam, T.J.R. Hughes, S. Bouabdallah, M. Zarroug, and H. Maitournam. Selective and reduced numerical integrations for NURBS-based isogeometric analysis. *Computer Methods in*

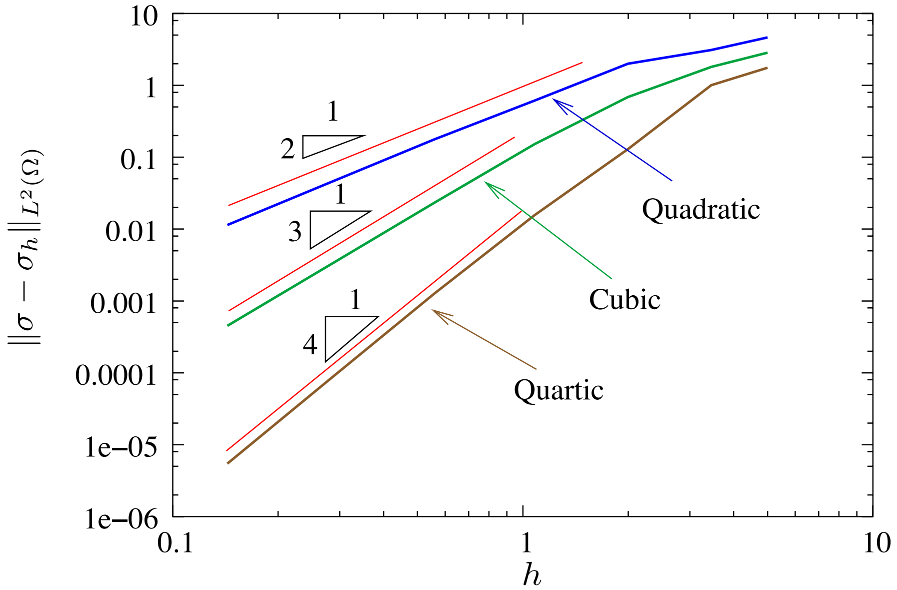


Fig. 1.36 Elasticity: error measured in the L^2-norm of stress vs. mesh parameter (optimal convergence rates in red).

Applied Mechanics and Engineering, 284:732–761, 2015.

[2] 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*, 285:817–828, 2015.

[3] Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numer.*, 15:1–155, 2006.

[4] Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. Finite element exterior calculus: from Hodge theory to numerical stability. *Bull. Amer. Math. Soc. (N.S.)*, 47(2):281–354, 2010.

[5] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation for elastostatics and explicit dynamics. *Comput. Meth. Appl. Mech. Engrg.*, 249–252:2–14, 2012.

[6] F. Auricchio, L. Beirão da Veiga, A. Buffa, C. Lovadina, A. Reali, and G. Sangalli. A fully ”locking-free” isogeometric approach for plane linear elasticity problems: a stream function formulation. *Comput. Methods Appl. Mech. Engrg.*, 197(1-4):160–172, 2007.

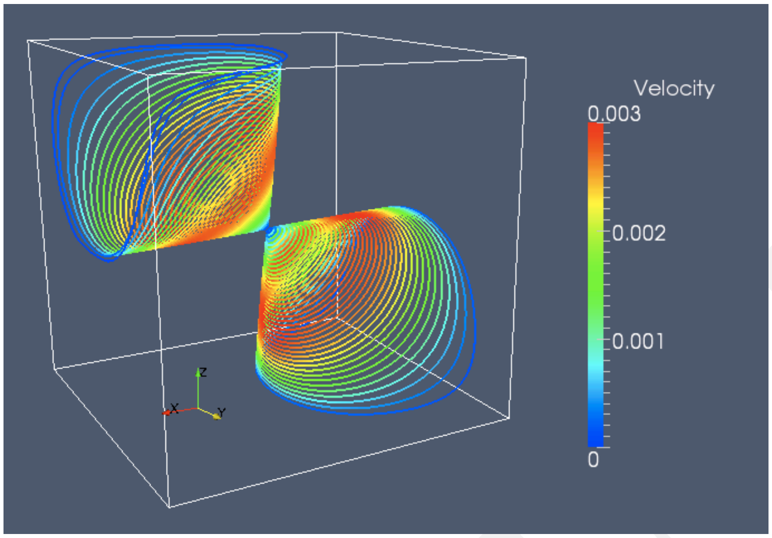


Fig. 1.37 Vortex manufactured solution: Flow velocity streamlines colored by velocity magnitude.

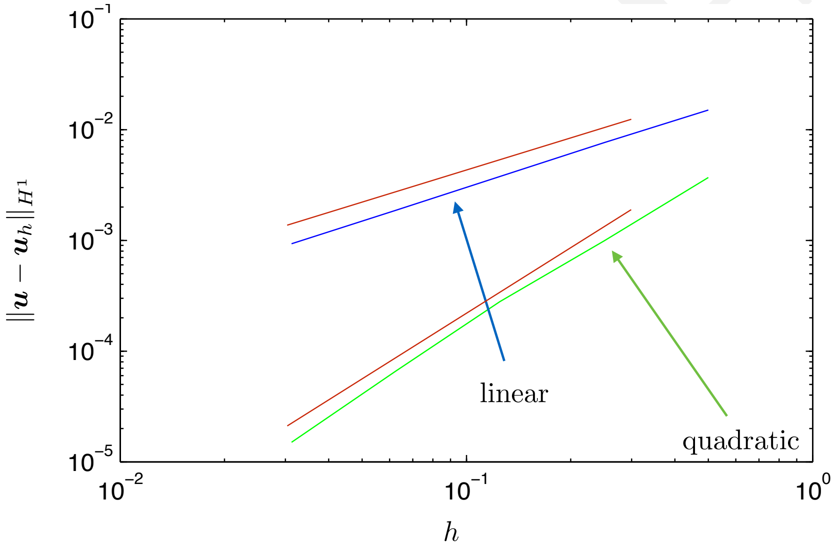


Fig. 1.38 Navier-Stokes: error measured in the H^1-norm of velocity vs. mesh-size h (the optimal convergence rates in red).

[7] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation methods. *Math. Models Methods Appl. Sci.*, 20(11):2075–2107, 2010.

[8] F. Auricchio, F. Calabrò, T.J.R. Hughes, A. Reali, and G. Sangalli. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 249:15–27, 2012.

[9] I Babuska and W Gui. The h, p and hp versions of the finite element method in 1 dimension. part ii. the error analysis of the h-and hp versions. *Numerische Mathematik*, 49:613–658, 1986.

[10] R. H. Bartels and G. W. Stewart. Solution of the matrix equation AX+XB=C. *Communications of the ACM*, 15(9):820–826, 1972.

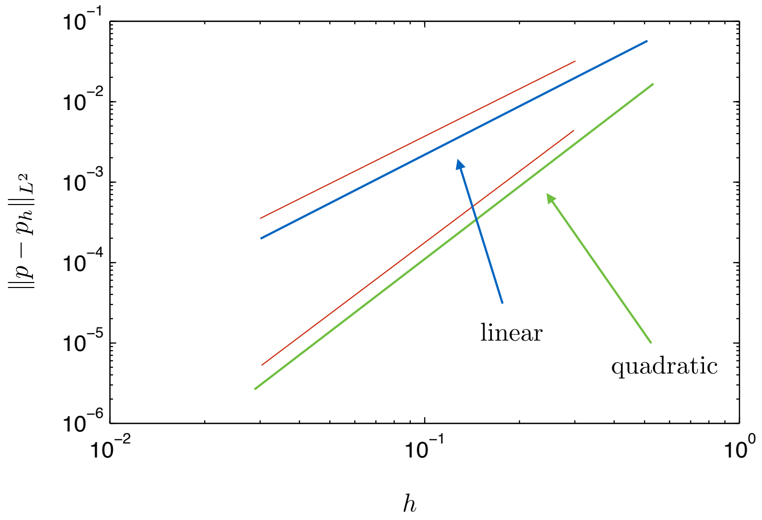


Fig. 1.39 Navier-Stokes: error measured in the L^2-norm of pressure vs. mesh-size h (optimal convergence rates in red).

[11] Andrea Bartezzaghi, Luca Dedè, and Alfio Quarteroni. Isogeometric analysis of high order partial differential equations on surfaces. *Computer Methods in Applied Mechanics and Engineering*, 295:446–469, 2015.

[12] Michael Bartoň and Victor Manuel Calo. Gaussian quadrature for splines via homotopy continuation: rules for C^2 cubic splines. *Journal of Computational and Applied Mathematics*, 296:709–723, 2016.

[13] Michael Bartoň and Victor Manuel Calo. Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 305:217–240, 2016.

[14] Y. Bazilevs, L. Beirão da Veiga, J. A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Math. Models Methods Appl. Sci.*, 16(7):1031–1090, 2006.

[15] Y. Bazilevs, V.M. Calo, J. A. Cottrell, J. A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Comput. Methods Appl. Mech. Engrg.*, 199(5-8):229–263, 2010.

[16] Y Bazilevs, M-C Hsu, and MA Scott. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Comput. Methods Appl. Mech. Engrg.*, 249:28–41, 2012.

[17] E. Beeker. Smoothing of shapes designed with free-form surfaces. *Computer-aided design*, 18(4):224–232, 1986.

[18] L. Beirão da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Mathematical analysis of variational isogeometric methods. *Acta Numerica*, 23:157–287, 5 2014.

[19] L. Beirão da Veiga, A. Buffa, D. Cho, and G. Sangalli. Analysis-Suitable T-splines are Dual-Compatible. *Comput. Methods Appl. Mech. Engrg.*, 249–252:42–51, 2012.

[20] L. Beirão da Veiga, A. Buffa, C. Lovadina, M. Martinelli, and G. Sangalli. An isogeometric method for the Reissner-Mindlin plate bending problem. *Comput. Methods Appl. Mech. Engrg.*, 209-212:45–53, 2012.

[21] L. Beirão da Veiga, A. Buffa, J. Rivas, and G. Sangalli. Some estimates for h-p-k-refinement in isogeometric analysis. *Numer. Math.*, 118(2):271–305, 2011.

[22] L. Beirão da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Analysis-suitable T-splines of arbitrary degree: Definition, linear independence and approximation properties. *Math. Models Methods Appl. Sci.*, 23(11):1979–2003, 2013.

[23] L. Beirao Da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Mathematical analysis of variational isogeometric methods. *Acta Numerica*, 23:157–287, 2014.

[24] L. Beirão da Veiga, D. Cho, L. F. Pavarino, and S. Scacchi. BDDC preconditioners for isogeometric analysis. *Mathematical Models and Methods in Applied Sciences*, 23(06):1099–1142, 2013.

[25] L. Beirão da Veiga, D. Cho, and G. Sangalli. Anisotropic NURBS approximation in Isogeometric Analysis. *Comput. Methods Appl. Mech. Engrg.*, 209-212:1–11, 2012.

[26] D.J. Benson, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes. A large deformation, rotation-free, isogeometric shell. *Comput. Methods Appl. Mech. Engrg.*, 200(13-16):1367–1378, 2011.

[27] M. Bercovier and T. Matskewich. Smooth Bezier surfaces over arbitrary quadrilateral meshes. *arXiv preprint arXiv:1412.1125*, 2014.

[28] M. Bercovier and I. Soloveichik. Overlapping non matching meshes domain decomposition method in isogeometric analysis. *arXiv preprint arXiv:1502.03756*, 2015.

[29] Michael J. Borden, Thomas J.R. Hughes, Chad M. Landis, and Clemens V. Verhoosel. A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework. *Computer Methods in Applied Mechanics and Engineering*, 273:100–118, 2014.

[30] Michael J. Borden, Clemens V. Verhoosel, Michael A. Scott, Thomas J.R. Hughes, and Chad M. Landis. A phase-field description of dynamic brittle fracture. *Comput. Meth. Appl. Mech. Engrg.*, 217–220(0):77–95, 2012.

[31] Robin Bouclier, Thomas Elguedj, and Alain Combescure. Efficient isogeometric NURBS-based solid-shell elements: Mixed formulation and-method. *Computer Methods in Applied Mechanics and Engineering*, 267:86–110, 2013.

[32] Andrea Bressan. Isogeometric regular discretization for the Stokes problem. *IMA J. Numer. Anal.*, 31(4):1334–1356, 2011.

[33] Andrea Bressan and Giancarlo Sangalli. Isogeometric discretizations of the Stokes problem: stability analysis by the macroelement technique. *IMA J. Numer. Anal.*, 33(2):629–651, 2013.

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

[35] Florian Buchegger, Bert Jüttler, and Angelos Mantzaflaris. Adaptively refined multi-patch B-splines with enhanced smoothness. *Applied Mathematics and Computation*, page in press, 2015.

[36] A. Buffa, C. de Falco, and G. Sangalli. Isogeometric Analysis: stable elements for the 2D Stokes equation. *Internat. J. Numer. Methods Fluids*, 65(11-12):1407–1422, 2011.

[37] A. Buffa, H. Harbrecht, A. Kunoth, and G. Sangalli. BPX-preconditioning for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 265:63–70, 2013.

[38] A. Buffa, J. Rivas, G. Sangalli, and R. Vázquez. Isogeometric discrete differential forms in three dimensions. *SIAM J. Numer. Anal.*, 49(2):818–844, 2011.

[39] Annalisa Buffa, Giancarlo Sangalli, and Christoph Schwab. Exponential convergence of the hp version of isogeometric analysis in 1d. In Mejdi Azaiez, Henda El Fekih, and Jan S. Hesthaven, editors, *Spectral and High Order Methods for Partial Differential Equations–ICOSAHOM 2012*, volume 95 of *Lecture Notes in Computational Science and Engineering*, pages 191–203. Springer International Publishing, 2014.

[40] F. Calabrò, G. Sangalli, and M. Tani. Fast formation of isogeometric Galerkin matrices by weighted quadrature. *ArXiv e-print n.1605.01238*, May 2016.

[41] F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, and P. Schröder. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Computer-Aided Design*, 34(2):137–148, 2002.

[42] E. Cohen, T. Martin, R. M. Kirby, T. Lyche, and R. F. Riesenfeld. Analysis-aware modeling: understanding quality considerations in modeling for isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 199(5-8):334–356, 2010.

[43] E. Cohen, R.F. Riesenfeld, and G. Elber. *Geometric modeling with splines: an introduction*, volume 1. AK Peters Wellesley, MA, 2001.

[44] N. Collier, L. Dalcin, D. Pardo, and V. M. Calo. The cost of continuity: performance of iterative solvers on isogeometric finite elements. *SIAM Journal on Scientific Computing*, 35(2):A767–A784, 2013.

[45] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, and V. M. Calo. The cost of continuity: a study of the performance of isogeometric finite elements using direct solvers. *Computer Methods in Applied Mechanics and Engineering*, 213:353–361, 2012.

[46] Annabelle Collin, Giancarlo Sangalli, and Thomas Takacs. Analysis-suitable G^1 multi-patch parametrizations for C^1 isogeometric spaces. *Computer Aided Geometric Design*, 2016. To Appear.

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

[48] J. A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Comput. Methods Appl. Mech. Engrg.*, 195(41-43):5257–5296, 2006.

[49] L Beirao da Veiga, D Cho, Luca F Pavarino, and Simone Scacchi. Overlapping Schwarz methods for isogeometric analysis. *SIAM Journal on Numerical Analysis*, 50(3):1394–1416, 2012.

[50] Carl de Boor. *A practical guide to splines*, volume 27 of *Applied Mathematical Sciences*. Springer-Verlag, New York, revised edition, 2001.

[51] R. Dimitri, L. De Lorenzis, M. Scott, P. Wriggers, R. Taylor, and G. Zavarise. Isogeometric large deformation frictionless contact using T-splines. *Comput. Methods Appl. Mech. Engrg.*, 269:394–414, 2014.

[52] Tor Dokken, Tom Lyche, and Kjell Fredrik Pettersen. Polynomial splines over locally refined box-partitions. *Comput. Aided Geom. Design*, 30(3):331–356, 2013.

[53] M. Donatelli, C. Garoni, C. Manni, S. Serra-Capizzano, and H. Speleers. Robust and optimal multi-iterative techniques for IgA Galerkin linear systems. *Computer Methods in Applied Mechanics and Engineering*, 284:230–264, 2015.

[54] M. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Comput. Methods Appl. Mech. Engrg.*, 199(5-8):264–275, 2010.

[55] J. A. Evans, Y. Bazilevs, I. Babuška, and T.J.R. Hughes. n-widths, sup-infs, and optimality ratios for the k-version of the isogeometic finite element method. *Comput. Methods Appl. Mech. Engrg.*, 198:1726–1741, 2009.

[56] J. A. Evans and T.J.R. Hughes. Isogeometric divergence-conforming B-splines for the Darcy-Stokes-Brinkman equations. *Math. Models Methods Appl. Sci.*, 23(04):671–741, 2013.

[57] J. A. Evans and T.J.R. Hughes. Isogeometric divergence-conforming B-splines for the Steady Navier-Stokes Equations. *Math. Models Methods Appl. Sci.*, 23(08):1421–1478, 2013.

[58] J. A. Evans and T.J.R. Hughes. Isogeometric Divergence-conforming B-splines for the Unsteady Navier-Stokes Equations. *J. Comput. Phys.*, 241:141–167, 2013.

[59] K. P. S. Gahalaut, J. K. Kraus, and S. K. Tomar. Multigrid methods for isogeometric discretization. *Computer methods in applied mechanics and engineering*, 253:413–425, 2013.

[60] L. Gao. *Kronecker Products on Preconditioning.* PhD thesis, King Abdullah University of Science and Technology, 2013.

[61] H. Gómez, V. Calo, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Comput. Methods Appl. Mech. Engrg.*, 197(49-50):4333–4352, 2008.

[62] Hector Gómez, Thomas J.R. Hughes, Xesús Nogueira, and Victor M. Calo. Isogeometric analysis of the isothermal Navier-Stokes-Korteweg equations. *Comput. Methods Appl. Mech. Engrg.*, 199(25-28):1828–1840, 2010.

[63] Phillip L Gould. *Introduction to linear elasticity.* Springer, 1994.

[64] M. Hillman, J.S. Chen, and Y. Bazilevs. Variationally consistent domain integration for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:521–540, 2015.

[65] R. Hiptmair. Finite elements in computational electromagnetism. *Acta Numer.*, 11:237–339, 2002.

[66] C. Hofreither, S. Takacs, and W. Zulehner. A robust multigrid method for isogeometric analysis using boundary correction. Technical Report 33, NFN, 2015.

[67] Saman Hosseini, Joris JC Remmers, Clemens V Verhoosel, and René Borst. An isogeometric solid-like shell element for nonlinear analysis. *Internat. J. Numer. Methods Engrg.*, 95(3):238–256, 2013.

[68] T. J.R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Engrg.*, 194(39-41):4135–4195, 2005.

[69] T.J.R. Hughes, J.A. Evans, and A. Reali. Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems. *Comput. Meth. Appl. Mech. Engrg.*, 272:290–320, 2014.

[70] T.J.R. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method NURBS. *Comput. Methods Appl. Mech. Engrg.*, 197(49-50):4104–4124, 2008.

[71] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 199(5-8):301–313, 2010.

[72] Kjetil André Johannessen. Optimal quadrature for univariate and tensor product splines. *Comput. Methods Appl. Mech. Engrg.*, 2016. To appear, DOI: 10.1016/j.cma.2016.04.030.

[73] B. Jüttler, A. Mantzaflaris, R. Perl, and M. Rumpf. On isogeometric subdivision methods for PDEs on surfaces. 2015.

[74] Mario Kapl, Vito Vitrih, Bert Jüttler, and Katharina Birner. Isogeometric analysis with geometrically continuous functions on two-patch geometries. *Computers & Mathematics with Applications*, 2015.

[75] A. Karatarakis, P. Karakitsios, and M. Papadrakakis. Gpu accelerated computation of the isogeometric analysis stiffness matrix. *Computer Methods in Applied Mechanics and Engineering*, 269:334–355, 2014.

[76] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Comput. Methods Appl. Mech. Engrg.*, 199(37-40):2403–2416, 2010.

[77] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wuchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Comput. Methods Appl. Mech. Engrg.*, 198(49–52):3902–3914, 2009.

[78] S. K. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. IETI–isogeometric tearing and interconnecting. *Computer Methods in Applied Mechanics and Engineering*, 247-248(0):201–215, 2012.

[79] D Liu and J Hoschek. $GC1$ continuity conditions between adjacent rectangular and triangular Bézier surface patches. *Computer-Aided Design*, 21(4):194–200, 1989.

[80] A. Mantzaflaris, B. Jüttler, B. Khoromskij, and U. Langer. Matrix generation in isogeometric analysis by low rank tensor approximation. 2014.

[81] P. Monk. *Finite element methods for Maxwell's equations.* Oxford University Press, Oxford, 2003.

[82] Bernard Mourrain, Raimundas Vidunas, and Nelly Villamizar. Geometrically continuous splines for surfaces of arbitrary topology. *arXiv preprint arXiv:1509.03274*, 2015.

[83] T. Nguyen, K. Karčiauskas, and J. Peters. A comparative study of several classical, discrete differential and isogeometric methods for solving Poisson's equation on the disk. *Axioms*, 3(2):280–299, 2014.

[84] Thien Nguyen and Jörg Peters. Refinable C^1 spline elements for irregular quad layout. *Computer Aided Geometric Design*, 43:123–130, 2016.

[85] D. W. Peaceman and H. H. Rachford. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial & Applied Mathematics*, 3(1):28–41, 1955.

[86] J. Peters. Smooth interpolation of a mesh of curves. *Constructive Approximation*, 7(1):221–246, 1991.

[87] L.A. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, 1997.

[88] David F. Rogers. *An Introduction to NURBS: With Historical Perspective*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[89] T. Rüberg and F. Cirak. Subdivision-stabilised immersed B-spline finite elements for moving boundary flows. *Comput. Methods Appl. Mech. Engrg.*, 209-212:266–283, 2012.

[90] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank. Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method. *Internat. J. Numer. Methods Engrg.*, 95(10):811–846, 2013.

[91] G. Sangalli and M. Tani. Isogeometric preconditioners based on fast solvers for the Sylvester equation. *ArXiv e-print 1602.01636*, February 2016.

[92] Giancarlo Sangalli, Thomas Takacs, and Rafael Vázquez. Unstructured spline spaces for isogeometric analysis based on spline manifolds. *arXiv preprint arXiv:1507.08477*, 2015. To appear on CAGD.

[93] D. Schillinger, J.A. Evans, A. Reali, M.A. Scott, and T.J.R. Hughes. Isogeometric collocation: cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Comput. Methods Appl. Mech. Engrg.*, 267:170–232, 2013.

[94] D. Schillinger, S.J. Hossain, and T.J.R. Hughes. Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 277:1–45, 2014.

[95] Dominik Schillinger, Luca Dedè, Michael A. Scott, John A. Evans, Michael J. Borden, Ernst Rank, and Thomas J.R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Comput. Methods Appl. Mech. Engrg.*, 249-252:116–150, 2012.

[96] Larry L. Schumaker. *Spline functions: basic theory*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, third edition, 2007.

[97] Ch. Schwab. *p- and hp-finite element methods*. Numerical Mathematics and Scientific Computation. The Clarendon Press Oxford University Press, New York, 1998. Theory and applications in solid and fluid mechanics.

[98] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. A. Bordas, T. JR Hughes, and T. W. Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197–221, 2013.

[99] M. A. Scott, D. C. Thomas, and E. J. Evans. Isogeometric spline forests. *Computer Methods in Applied Mechanics and Engineering*, 269:222–264, 2014.

[100] M.A. Scott, X. Li, T.W. Sederberg, and T.J.R. Hughes. Local refinement of analysis-suitable T-splines. *Comput. Methods Appl. Mech. Engrg.*, 213–216:206–222, 2012.

[101] Thomas W. Sederberg, G. Thomas Finnigan, Xin Li, Hongwei Lin, and Heather Ipson. Watertight trimmed NURBS. *ACM Transactions on Graphics (TOG)*, 27(3):79, 2008.

[102] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. *ACM Trans. Graph.*, 23(3):276–283, 2004.

[103] T.W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCSs. *ACM Trans. Graph.*, 22(3):477–484, 2003.

[104] V. Simoncini. Computational methods for linear matrix equations. *to appear on SIAM Review*, 2013.

[105] Gilbert Strang and George J Fix. *An analysis of the finite element method*, volume 212. Prentice-Hall Englewood Cliffs, NJ, 1973.

[106] Stefan Takacs and Thomas Takacs. Approximation error estimates and inverse inequalities for b-splines of maximum smoothness. *arXiv preprint arXiv:1502.03733*, 2015.

[107] T. Takacs and B. Jüttler. Existence of stiffness matrix integrals for singularly parameterized domains in isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 200(49–52):3568–3582, 2011.

[108] T. Takacs and B. Jüttler. H^2 regularity properties of singular parameterizations in isogeometric analysis. *Graph. Models*, 74(6):361–372, 2012.

[109] T. Takacs, B. Jüttler, and O. Scherzer. Derivatives of isogeometric functions on n-dimensional rational patches in R^d. *Computer Aided Geometric Design*, 31(7-8):567–581, 2014. Recent Trends in Theoretical and Applied Geometry.

[110] Thomas Takacs. Construction of smooth isogeometric function spaces on singularly parameterized domains. In *Curves and Surfaces*, pages 433–451. Springer, 2014.

[111] Thomas Takacs. Approximation properties of isogeometric function spaces on singularly parameterized domains. *arXiv preprint arXiv:1507.08095*, 2015.

[112] R.L. Taylor. Isogeometric analysis of nearly incompressible solids. *International Journal for Numerical Methods in Engineering*, 87(1-5):273–288, 2011.

[113] R. Vázquez. A new design for the implementation of isogeometric analysis in Octave and Matlab: GeoPDEs 3.0. Technical Report 16-02, IMATI Report Series, April 2016.

[114] Clemens V. Verhoosel, Michael A. Scott, René De Borst, and Thomas J.R. Hughes. An isogeometric approach to cohesive zone modeling. *International Journal for Numerical Methods in Engineering*, 87(1-5):336–360, 2011.

[115] Clemens V. Verhoosel, Michael A. Scott, Thomas J.R. Hughes, and René de Borst. An isogeometric analysis approach to gradient damage models. *Internat. J. Numer. Methods Engrg.*, 86(1):115–134, 2011.

[116] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 200(49-52):3554–3567, 2011.

[117] E. L. Wachspress. Extended application of alternating direction implicit iteration model problem theory. *Journal of the Society for Industrial & Applied Mathematics*, 11(4):994–1016, 1963.

[118] Meng Wu, Bernard Mourrain, André Galligo, and Boniface Nkonga. Hermite type spline spaces over rectangular meshes with arbitrary topology. Technical report, 2016. HAL Id: hal-01196428.