# Adaptive isogeometric analysis based on locally refined Tchebycheffian B-splines

Krunal Raval[a,*], Carla Manni[a], Hendrik Speleers[a]

[a]*Department of Mathematics, University of Rome Tor Vergata, 00133 Rome, Italy*

**Abstract**

We introduce locally refined (LR) Tchebycheffian B-splines as a generalization of LR B-splines from the algebraic polynomial setting to the broad Tchebycheffian setting. We focus on the particularly interesting class of Tchebycheffian splines whose pieces belong to null-spaces of constant-coefficient linear differential operators. They offer the freedom of combining algebraic polynomials with exponential and trigonometric functions with any number of individual shape parameters and have been recently equipped with efficient evaluation and manipulation procedures. We consider their application in the context of isogeometric analysis and discuss related adaptive refinement, adopting the so-called structured mesh refinement strategy, widely used and analyzed in the classical polynomial case.

*Keywords:* Isogeometric analysis, Tchebycheffian B-splines, Locally refined meshes

## 1. Introduction

Tensor-product B-splines are probably the most well-known multivariate spline basis functions. They have been profitably applied in different contexts including geometric modeling, approximation theory, and numerical simulation. Their popularity roots in their simple, elegant, and efficient construction: they are nothing but tensor products of univariate B-splines; see, e.g., [21, 32] and references therein.

The tensor structure of the underlying mesh, however, is the major weakness of tensor-product B-splines as it hinders adequate local refinement, forcing the use of unnecessarily large discrete spaces and leading to a significant loss in efficiency. This has been seen as a severe limitation in the context of isogeometric analysis, which aims to simplify the interoperability between geometric modeling and numerical simulation by constructing a fully integrated framework for computer-aided design and finite element analysis; see, e.g., [10, 15]. Adaptive local refinement strategies are essential in isogeometric analysis, in order to achieve small approximation error while keeping the computational cost low.

To overcome this limitation, in the last decades many alternative spline technologies have been developed for so-called T-meshes. Such meshes are still axis-aligned but T-vertices are allowed in the interior of the domain, in order to support local refinement, while preserving locally the simplicity of the tensor approach; see [35] and references therein. T-splines [36], (truncated) hierarchical B-splines [13], PHT-splines [11], and locally refined (LR) B-splines [12] are popular examples of such spline technologies. All these

*Corresponding author

*Email addresses:* raval@mat.uniroma2.it (Krunal Raval), manni@mat.uniroma12.it (Carla Manni), speleers@mat.uniroma2.it (Hendrik Speleers)

approaches have their own strengths (and weaknesses) depending on the context they are intended to be used.

The definition of LR B-splines is inspired by the knot insertion refinement process of univariate (and tensor-product) B-splines. Their formulation bears a large similarity to classical tensor-product B-splines and this makes them one of the most elegant extensions of univariate B-splines towards T-meshes. Since their introduction in [12], LR B-splines have found interesting applications in several contexts ranging from data approximation [20, 37] to numerical simulations [17, 18], also considering their rational version [41]. More theoretical aspects, mainly related to the issue of linear and local linear independence and related adaptive refinement strategies, have been investigated in [7, 28, 29, 30]. A comparison between LR B-splines, hierarchical, and truncated hierarchical B-splines can be found in [19], while combinations of the LR B-spline framework with the hierarchical approach have been explored in [8, 40].

Univariate Tchebycheffian splines are smooth piecewise functions whose pieces are drawn from (possibly different) extended Tchebycheff (ET-) spaces. Any non-trivial element of an ET-space of dimension $p+1$ has at most $p$ zeros, counting multiplicity. ET-spaces are natural generalizations of algebraic polynomial spaces [22, 26, 34]. Extended complete Tchebycheff (ECT-) spaces [22, 31] are a particularly interesting subclass of ET-spaces enjoying some additional useful properties, such as the possibility of defining so-called generalized power functions, which can be regarded as a natural extension of the monomial basis functions for algebraic polynomials. Relevant examples are null-spaces of linear differential operators on suitable intervals [22, 31]. From now on we will focus only on ECT-spaces.

Most of the results known for polynomial splines extend in a natural way to the Tchebycheffian setting. In particular, under suitable assumptions on the involved ECT-spaces, Tchebycheffian splines admit a representation in terms of basis functions, called Tchebycheffian B-splines (TB-splines), which enjoy almost all the properties that make polynomial B-splines popular and successful, such as local linear independence, minimal support, non-negativity, and partition of unity. Unfortunately, TB-splines in their most generality still lack practical algorithms for their evaluation and manipulation. This almost nullifies their great applicative potential.

On the contrary, for the subclass of TB-splines with pieces belonging to ECT-spaces that are null-spaces of constant-coefficient linear differential operators, manipulation routines have been recently developed and made publicly available in a Matlab toolbox [38], based on the results in [14]. As a consequence, thanks to their structural similarity and plug-to-plug compatibility with classical polynomial B-splines, they can be easily incorporated in any software library supporting polynomial B-splines. This subclass of TB-splines provides a large variety of combinations of algebraic polynomial, exponential, and trigonometric functions equipped with a wide spectrum of shape parameters. They allow for an exact representation of profiles of interest in applications (such as conic sections), they behave nicely with respect to differentiation and integration and, by construction, they include fundamental solutions of certain differential operators. Therefore, they offer a valid alternative to classical polynomial B-splines (and their rational extension NURBS) in isogeometric Galerkin methods. As illustrated in the analysis presented in [31], it turns out that tensor-product TB-splines can outperform tensor-product polynomial B-splines in isogeometric Galerkin methods whenever appropriate problem-driven selection strategies for the underlying ECT-spaces are applied.

The structural similarity between ECT-spaces and algebraic polynomial spaces also enables us to extend popular local refinement technologies, based on local tensor products, towards the Tchebycheffian setting. Tchebycheffian spline spaces over T-meshes have been introduced in their full generality in [5]. The structure of ECT-spaces has been exploited in [6] to fully extend the dimension study carried out in the polynomial case in [27]. Some earlier generalizations of the polynomial setting towards particular Tchebycheffian spline spaces or peculiar T-meshes have been considered in [2, 3, 4, 25]. In particular, [4]

outlines the construction of so-called generalized splines (a special instance of Tchebycheffian splines) on LR-meshes.

In this paper, we define LR TB-splines as a generalization of LR B-splines and we analyze their performance in the context of adaptive isogeometric analysis. The definition of LR TB-splines is driven by the knot insertion refinement process of tensor-product TB-splines, in complete analogy to the polynomial setting. In the bivariate tensor case, inserting a new knot in a pair of (global) knot vectors results in inserting a line segment in the mesh crossing the entire domain, thus refining all the TB-splines whose supports are crossed. On the contrary, LR TB-splines are defined on local knot vectors, and consequently the insertion of a new knot is always performed with respect to a particular LR TB-spline and results in refining only few basis functions. The theoretical construction of LR TB-splines is independent of the particular ECT-spaces where the various pieces are drawn from. However, in the applicative context we confine ourselves to Tchebycheffian splines identified by ECT-spaces that are null-spaces of constant-coefficient linear differential operators containing constants because:

- they already grant the freedom of combining algebraic polynomials with exponential and trigonometric functions with any number of individual shape parameters;
- when the various pieces are drawn from a single ECT-space, the existence of TB-splines is always ensured, possibly with some restriction on the partition; see [31, Section 2.4];
- the corresponding TB-splines are supported by the Matlab toolbox available in [38].

The remainder of the paper is divided into five sections. Section 2 summarizes the definition and main properties of Tchebycheffian splines, with a particular focus on TB-splines and knot insertion. Section 3 details and illustrates the construction of LR TB-splines. In Section 4 we consider their use in the isogeometric Galerkin method and discuss related adaptive refinement. Section 5 collects various numerical results, showcasing the performance of adaptive isogeometric analysis based on LR TB-splines for the solution of few classical benchmark differential problems. We end in Section 6 with some concluding remarks.

Throughout the paper, we assume the reader to be familiar with the definition and main properties of univariate polynomial B-splines, in particular with the knot insertion procedure; see, e.g., [21].

## 2. Tchebycheffian splines

Tchebycheffian splines are smooth piecewise functions with pieces drawn from (possibly different) extended Tchebycheff (ET-) spaces glued together with prescribed smoothness. ET-spaces, and more precisely their important subclass of extended complete Tchebycheff (ECT-) spaces, are natural generalizations of algebraic polynomial spaces. The focus of this paper will be on a large class of ECT-spaces given by the null-spaces of linear differential operators with real constant coefficients on suitable intervals; see [34, 38, 31]. In this section, following the presentation and the notation from [31, Section 2], we briefly recollect the definition and some properties of Tchebycheffian splines. In particular, we discuss their representation, whenever it exists, in terms of Tchebycheffian B-splines (TB-splines), which are B-spline like basis functions for Tchebycheffian spline spaces. We refer the reader to [34, 22] for in-depth information about Tchebycheffian splines.

### *2.1. Extended Tchebycheff spaces from null-spaces of linear differential operators*

We start by defining our Tchebycheff spaces of interest.

**Definition 1** (Extended Tchebycheff space)**.** Given an interval $J$ and an integer $p \geq 0$, a space $\mathbb{T}_p(J) \subset C^p(J)$ of dimension $p+1$ is an extended Tchebycheff (ET-) space on $J$ if any non-trivial element of $\mathbb{T}_p$ has at most $p$ roots in $J$, counting multiplicity.

This definition shows a strong connection between ET-spaces $\mathbb{T}_p$ and the space $\mathbb{P}_p$ of algebraic polynomials of degree less than or equal to $p$. This relation is even further strengthened when considering the following important subclass of ET-spaces.

**Definition 2** (Extended complete Tchebycheff space)**.** An ET-space $\mathbb{T}_p(J)$ is an extended complete Tchebycheff (ECT-) space on $J$, if there exist functions $g_0, \ldots, g_p$ such that $\mathbb{T}_p(J) = \langle g_0, \ldots, g_p \rangle$ and every subspace $\langle g_0, \ldots, g_k \rangle$ for $k = 0, \ldots, p$ is an ET-space on $J$. The basis $\{g_0, \ldots, g_p\}$ is called an ECT-system.

The functions $g_0, \ldots, g_p$ in Definition 2 can be regarded as a natural extension of the monomial basis functions for algebraic polynomials. Due to the above similarities, we will refer to the integer $p \geq 0$ as the degree of an ECT-space $\mathbb{T}_p$, in analogy to the space $\mathbb{P}_p$ of algebraic polynomials.

In this paper we are focusing on the subclass of ECT-spaces given by the null-spaces of linear differential operators with real constant coefficients. Such a differential operator is defined by

$$\mathcal{L}_p f := D^{p+1} f + \sum_{j=0}^{p} a_j D^j f, \quad f \in C^{p+1}(\mathbb{R}), \quad a_j \in \mathbb{R}, \quad j = 0, \ldots, p, \tag{1}$$

and its characteristic polynomial by

$$\mathfrak{p}_p(\omega) := \omega^{p+1} + \sum_{j=0}^{p} a_j \omega^j. \tag{2}$$

The null-space of the differential operator in (1) is fully identified by the roots of $\mathfrak{p}_p$ in (2). We assume that $\omega = 0$ is a root, in order to ensure that the constants belong to the null-space. Let us denote all the different roots of $\mathfrak{p}_p$ by $\omega_k = \alpha_k + \mathrm{i}\beta_k$ ($\alpha_k, \beta_k \in \mathbb{R}$ and $\mathrm{i} := \sqrt{-1}$), with $\omega_0 = 0$, and their multiplicity by $\mu_k \geq 1$ for $k = 0, \ldots, M$ such that $\sum_{k=0}^{M} \mu_k = p+1$. Then, the corresponding null-space is uniquely characterized by the following vector with $p + 1 - \mu_0$ components:

$$\mathcal{W} := \Big( \underbrace{\omega_1, \ldots, \omega_1}_{\mu_1 \text{ times}}, \underbrace{\omega_2, \ldots, \omega_2}_{\mu_2 \text{ times}}, \ldots, \underbrace{\omega_M, \ldots, \omega_M}_{\mu_M \text{ times}} \Big), \tag{3}$$

and can be denoted by

$$\mathbb{P}_p^{\mathcal{W}} := \mathbb{P}_p^{(\alpha_1+\mathrm{i}\beta_1, \ldots, \alpha_1+\mathrm{i}\beta_1, \alpha_2+\mathrm{i}\beta_2, \ldots, \alpha_2+\mathrm{i}\beta_2, \ldots, \alpha_M+\mathrm{i}\beta_M)}. \tag{4}$$

If the polynomial in (2) has solely real roots, then the null-space is an ECT-space on the whole real line, and in particular on any bounded interval $[a, b]$. On the other hand, if the characteristic polynomial has also complex roots, then it is only an ECT-space on sufficiently small intervals.

**Remark 3.** *Let $\omega = \alpha + \mathrm{i}\beta$ be a root of multiplicity $\mu \geq 1$ of the polynomial in* (2)*. This root gives rise to the following fundamental subspace:*

- *if $\beta = 0$, then*
$$\left\langle x^k e^{\alpha x} : k = 0, \ldots, \mu - 1 \right\rangle \subseteq \mathbb{P}_p^{\mathcal{W}};$$
- *if $\beta \neq 0$, then the complex conjugate of $\omega$ is also a root of multiplicity $\mu$, and*
$$\left\langle x^k e^{\alpha x} \cos(\beta x), x^k e^{\alpha x} \sin(\beta x) : k = 0, \ldots, \mu - 1 \right\rangle \subseteq \mathbb{P}_p^{\mathcal{W}}.$$

*The collective subspaces mentioned above, corresponding to all the roots, constitute the null-space $\mathbb{P}_p^{\mathcal{W}}$ of the considered differential operator.*

In this paper we are considering the comprehensive subclass of ECT-spaces described by

$$\mathbb{T}_p = \mathbb{P}_p^{(\alpha_1,\dots,\alpha_\ell,\mathrm{i}\beta_1,-\mathrm{i}\beta_1,\dots,\mathrm{i}\beta_q,-\mathrm{i}\beta_q)} = \Big\langle 1, x, \dots, x^{p-\ell-2q}, e^{\alpha_1 x}, \dots, e^{\alpha_\ell x}, \cos(\beta_1 x), \sin(\beta_1 x), \dots, \cos(\beta_q x), \sin(\beta_q x)\Big\rangle, \tag{5}$$

where $0 \le \ell \le p$ and $\alpha_1, \dots, \alpha_\ell \in \mathbb{R}$ with $\alpha_i \ne \alpha_j$ for each $i \ne j$; and $0 \le 2q \le p - \ell$ with $\beta_1, \dots, \beta_q \in \mathbb{R}$ and $\beta_i \ne \beta_j$ for each $i \ne j$. The class in (5) grants the freedom of combining algebraic polynomials with exponential and trigonometric functions by choosing a wide range of shape parameters in (3). An extensive study on the selection of the individual shape parameters, according to an automatic problem-driven strategy, can be found in [31]. Here we limit ourselves to recall few particular selections of these shape parameters, which result into familiar ECT-spaces.

- Algebraic polynomial spaces $\mathbb{P}_p = \langle 1, x, \dots, x^p \rangle$ are the most established ECT-spaces.
- Generalized polynomial spaces enrich algebraic polynomial spaces by a pair of functions [22, 23]. Important cases of generalized polynomials can be obtained by taking $M = 2$ and $\mu_1 = \mu_2 = 1$ in (3). In particular, when selecting $\omega_1 = \alpha$, $\omega_2 = -\alpha$ we get the following null-space,

$$\mathbb{P}_p^{(\alpha,-\alpha)} = \Big\langle 1, x, \dots, x^{p-2}, e^{\alpha x}, e^{-\alpha x}\Big\rangle = \Big\langle 1, x, \dots, x^{p-2}, \cosh(\alpha x), \sinh(\alpha x)\Big\rangle, \quad p \ge 2; \tag{6}$$

  while setting $\omega_1 = \mathrm{i}\beta$, $\omega_2 = -\mathrm{i}\beta$ results in the following null-space,

$$\mathbb{P}_p^{(\mathrm{i}\beta,-\mathrm{i}\beta)} = \Big\langle 1, x, \dots, x^{p-2}, \cos(\beta x), \sin(\beta x)\Big\rangle, \quad p \ge 2. \tag{7}$$

  We note that the space (6) is an ECT-space on any interval. On the contrary, the space (7) is only an ECT-space on a sufficiently small interval. It can be shown that any closed interval of length less than $\frac{2\pi}{\beta}$ is valid; see [31, Section 2.1] and references therein for further details.

### 2.2. Tchebycheffian spline spaces

Analogous to polynomial splines, Tchebycheffian splines are piecewise functions with pieces belonging to ECT-spaces and ensuring specific prescribed smoothness. Let $\mathcal{M}$ be a mesh of the interval $[a, b] \subset \mathbb{R}$ defined by

$$\mathcal{M} := \{a =: x_0 < x_1 < \cdots < x_{m-1} < x_m := b\}. \tag{8}$$

For $i = 1, \dots, m$ let $\mathbb{T}_{p,i}$ be an ECT-space of dimension $p + 1$ on the closed interval $[x_{i-1}, x_i]$ and consider the sequence of $m - 1$ integers

$$\mathbf{r} := \Big\{r_i \in \mathbb{Z}_{\ge 0} : 0 \le r_i \le p - 1,\ i = 1, \dots, m - 1\Big\}.$$

Then, the Tchebycheffian spline space $\mathbb{S}_p^{\mathbf{r}}(\mathcal{M})$ of degree $p$ and smoothness $\mathbf{r}$ on the mesh $\mathcal{M}$, associated with the ECT-spaces $\mathbb{T}_{p,i}$, is defined by

$$\mathbb{S}_p^{\mathbf{r}}(\mathcal{M}) := \Big\{f : [a, b] \to \mathbb{R} : f|_{[x_{i-1},x_i]} \in \mathbb{T}_{p,i},\ i = 1, \dots, m;\\ D_-^l f(x_i) = D_+^l f(x_i),\ l = 0, \dots, r_i,\ i = 1, \dots, m - 1\Big\}.$$

For the sake of simplicity, in this paper we confine ourselves to select all the pieces of the spline space from a single space $\mathbb{T}_p$ as in (5). We assume that $\mathbb{T}_p$ (and its derivative space) is an ECT-space on each

interval $[x_{i-1}, x_i]$ separately, but not necessarily on the entire interval $[a, b]$. As it will be shown in the rest of the paper, also in this simplified setting, the framework we are dealing with is very rich and offers enough flexibility to allow a proper treatment for a wide class of problems. Moreover, we assume maximal smoothness at the internal breakpoints in the partition, i.e., $\mathbf{r} := \{p-1, \dots, p-1\}$, and we drop the smoothness $\mathbf{r}$ from the notation of the Tchebycheffian spline space. Thus, the Tchebycheffian spline space $\mathbb{S}_p(\mathcal{M})$ of degree $p$ where all the pieces are drawn from the space $\mathbb{T}_p$ and glued with maximal smoothness on a partition $\mathcal{M}$ is given as

$$\mathbb{S}_p(\mathcal{M}) := \Big\{ f : [a,b] \to \mathbb{R} : f|_{[x_{i-1},x_i]} \in \mathbb{T}_p,\ i = 1, \dots, m; \\ D^l_- f(x_i) = D^l_+ f(x_i),\ l = 0, \dots, p-1,\ i = 1, \dots, m-1 \Big\}. \tag{9}$$

### *2.3. Tchebycheffian B-splines*

In the particular case $\mathbb{T}_p = \mathbb{P}_p$, the space in (9) admits a special basis formed by the so-called B-splines. This basis allows for an efficient representation, evaluation, and manipulation of the elements in the space. Under suitable assumptions on the ECT-space $\mathbb{T}_p$, the spline space in (9) admits a set of basis functions which enjoy all the main properties of B-splines and therefore are called Tchebycheffian B-splines (TB-splines). In the rest of the paper we assume that the space (9) admits a TB-spline basis. We refer the reader to [31, Section 2.4], and references therein, for a detailed discussion about the existence and definition of such a basis. Here we limit ourselves to recall the main properties of TB-splines.

Like in the polynomial B-spline case, the set of TB-splines of the space in (9) can be defined using a vector of non-decreasing (open) knots,

$$\Xi := (\xi_k)_{k=1}^{n+p+1} := \Big( \underbrace{x_0, \dots, x_0}_{p+1 \text{ times}}, x_1, \dots, x_{m-1}, \underbrace{x_m, \dots, x_m}_{p+1 \text{ times}} \Big), \tag{10}$$

with $n := m + p$ being the dimension of the Tchebycheffian spline space $\mathbb{S}_p(\mathcal{M})$, and any TB-spline $N_{\Xi_k,p} : \mathbb{R} \to \mathbb{R}$ of degree $p$ can be uniquely identified by a local knot vector of length $p + 2$,

$$\Xi_k := (\xi_{k,1}, \dots, \xi_{k,p+2}),$$

where $\xi_{k,1}, \dots, \xi_{k,p+2}$ are consecutive knots in (10). TB-splines possess a complete structural similarity to polynomial B-splines. The next proposition collects some properties related to our Tchebycheffian spline spaces of interest. The listed properties, however, remain valid under much more general assumptions; see [14, 22, 34] and references therein.

**Proposition 4** (TB-spline properties)**.** *Let $\mathcal{S}_p(\mathcal{M}) := \{N_{\Xi_k,p} : k = 1, \dots, n\}$ be the set of all TB-splines defined on the sequence of knots* (10) *and related to the space $\mathbb{S}_p(\mathcal{M})$ in* (9)*, with $\mathbb{T}_p$ as in* (5)*. The following properties hold for $N_{\Xi_k,p}$ (see Figure 1).*

- *Non-negativity: $N_{\Xi_k,p}(x) > 0$ for all $x \in (\xi_{k,1}, \xi_{k,p+2})$.*
- *Local support:* $\operatorname{supp}(N_{\Xi_k,p}) = [\xi_{k,1}, \xi_{k,p+2}]$*, hence $N_{\Xi_k,p}(x) = 0$ for all $x \notin [\xi_{k,1}, \xi_{k,p+2}]$.*
- *Partition of unity: $\sum_{k=1}^{n} N_{\Xi_k,p}(x) = 1$ for all $x \in [a, b]$.*
- *Local linear independence: the set $\{N_{\Xi_k,p}\}_{k=i}^{i+p}$ forms a basis of $\mathbb{T}_p$ on $[x_{i-1}, x_i]$ for all $1 \le i \le m$.*

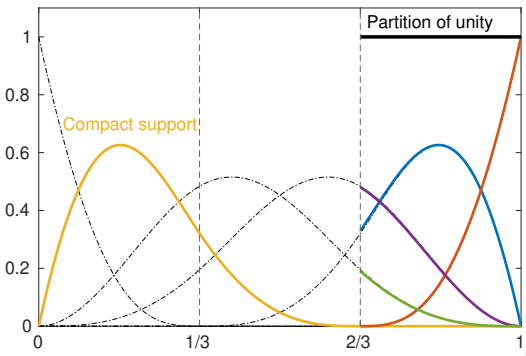



Figure 1: Illustration of TB-spline properties.

- *Interpolation at end-points:*

$$N_{\Xi_1,p}(a) = 1, \quad N_{\Xi_k,p}(a) = 0, \quad k = 2, \dots, n;$$
$$N_{\Xi_n,p}(b) = 1, \quad N_{\Xi_k,p}(b) = 0, \quad k = 1, \dots, n-1.$$

When $m = 1$, the mesh $\mathcal{M}$ has no interior breakpoints and the corresponding TB-splines reduce to the Tchebycheffian counterpart of Bernstein polynomials, called Tchebycheffian Bernstein functions. Some examples of Tchebycheffian Bernstein functions for different ECT-spaces are depicted in Figure 2.

### 2.4. Knot insertion

Similar to polynomial splines, we can represent a Tchebycheffian spline on a refined knot sequence by inserting a single knot at a time. This follows from the following proposition; see, e.g., [22, Section 3.4] for a proof. We use the notation $(\xi_1, \dots, \hat{\xi}, \dots, \xi_q)$ to indicate the knot vector that is obtained after adding the knot $\hat{\xi}$ to the knot vector $(\xi_1, \dots, \xi_q)$ and this implicitly assumes that the resulting vector is in non-decreasing order. In particular, if $\hat{\xi} < \xi_1$ then $\hat{\xi}$ will be the first item of the vector, or if $\hat{\xi} > \xi_q$ then it will be the last item of the vector.

**Proposition 5** (Knot insertion)**.** *Let $N_{\Xi_k,p}$ be a TB-spline as in Proposition 4, identified by the local knot vector $\Xi_k := (\xi_{k,1}, \dots, \xi_{k,p+2})$. Adding a knot $\hat{\xi} \in [\xi_{k,1}, \xi_{k,p+2}]$ results in two TB-splines*

$$N_{(\xi_{k,1},\dots,\hat{\xi},\dots,\xi_{k,p+1}),p}, \quad N_{(\xi_{k,2},\dots,\hat{\xi},\dots,\xi_{k,p+2}),p},$$

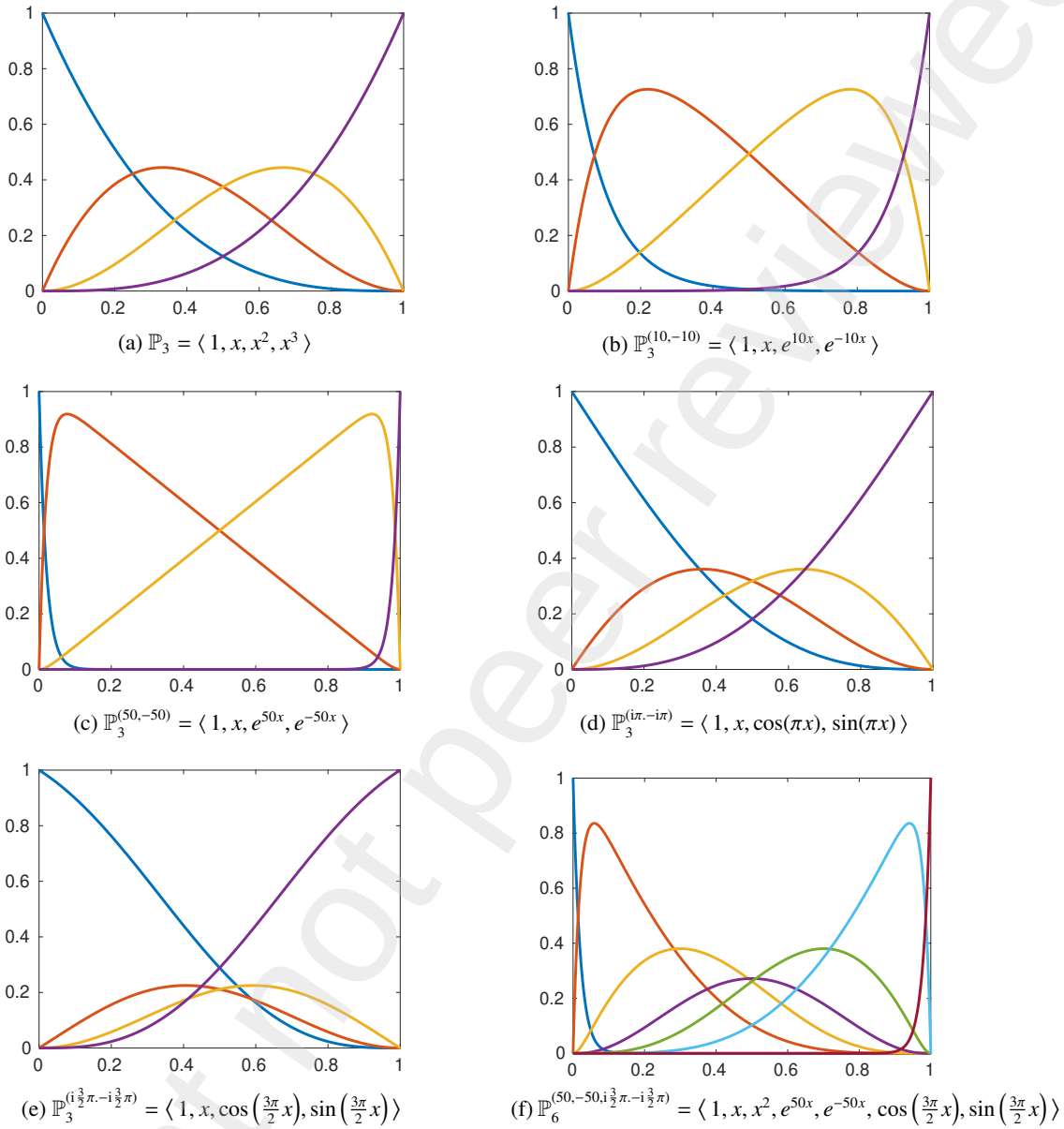
*such that*

$$N_{(\xi_{k,1},\dots,\xi_{k,p+2}),p}(x) = \nu_k^{(1)} N_{(\xi_{k,1},\dots,\hat{\xi},\dots,\xi_{k,p+1}),p}(x) + \nu_k^{(2)} N_{(\xi_{k,2},\dots,\hat{\xi},\dots,\xi_{k,p+2}),p}(x), \tag{11}$$

*with*

$$\begin{cases} \nu_k^{(1)} = 0, \ \nu_k^{(2)} = 1, & \text{if } \hat{\xi} = \xi_{k,1}, \\ \nu_k^{(1)} > 0, \ \nu_k^{(2)} > 0, & \text{if } \xi_{k,1} < \hat{\xi} < \xi_{k,p+2}, \\ \nu_k^{(1)} = 1, \ \nu_k^{(2)} = 0, & \text{if } \hat{\xi} = \xi_{k,p+2}. \end{cases}$$

The coefficients $\nu_k^{(1)}, \nu_k^{(2)} \in [0, 1]$, connecting two different scales of TB-splines, can be explicitly expressed in an analytical form involving the underlying space $\mathbb{T}_p$; we refer the reader to [22, Section 3.4]



(a) $\mathbb{P}_3 = \langle\, 1, x, x^2, x^3 \,\rangle$ (b) $\mathbb{P}_3^{(10,-10)} = \langle\, 1, x, e^{10x}, e^{-10x} \,\rangle$

(c) $\mathbb{P}_3^{(50,-50)} = \langle\, 1, x, e^{50x}, e^{-50x} \,\rangle$ (d) $\mathbb{P}_3^{(\mathrm{i}\pi,-\mathrm{i}\pi)} = \langle\, 1, x, \cos(\pi x), \sin(\pi x) \,\rangle$

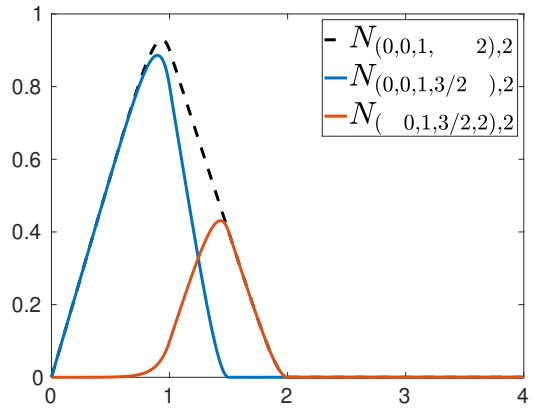
(e) $\mathbb{P}_3^{(\mathrm{i}\frac{3}{2}\pi,-\mathrm{i}\frac{3}{2}\pi)} = \langle\, 1, x, \cos\left(\frac{3\pi}{2}x\right), \sin\left(\frac{3\pi}{2}x\right) \,\rangle$ (f) $\mathbb{P}_6^{(50,-50,\mathrm{i}\frac{3}{2}\pi,-\mathrm{i}\frac{3}{2}\pi)} = \langle\, 1, x, x^2, e^{50x}, e^{-50x}, \cos\left(\frac{3\pi}{2}x\right), \sin\left(\frac{3\pi}{2}x\right) \,\rangle$

Figure 2: Tchebycheffian Bernstein basis on the interval $[0, 1]$ for different ECT-spaces.

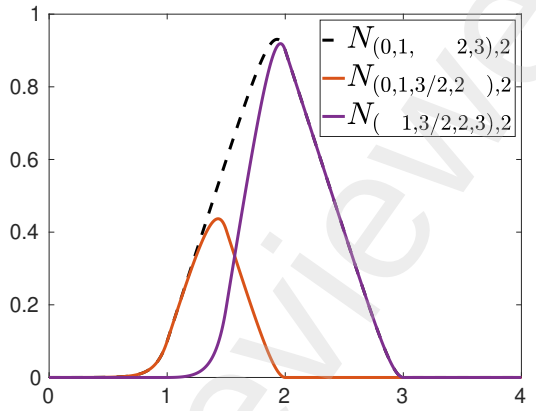
for details. Alternatively, for practical purposes, they can be simply computed by solving the linear system obtained from (11) by evaluating the two sides at any pair of distinct points $\hat{x}_1$, $\hat{x}_2$ such that

$$\hat{x}_j \in (\xi_{k,j}, \xi_{k,p+j}), \quad j = 1, 2.$$

**Example 6** (Knot insertion)**.** *Let us consider a Tchebycheffian spline space* $\mathbb{S}_p(\mathcal{M})$ *on the mesh* $\mathcal{M} = \{0, 1, 2, 3\}$ *related to the ECT-space* $\mathbb{P}_2^{(10)} = \left\langle 1, x, e^{10x} \right\rangle$ *of degree* $p = 2$*. We refine the TB-spline* $N_{(0,0,1,2),2}$




(a) $N_{(0,0,1,2),2} = N_{(0,0,1,3/2),2} + \left(\frac{e^{\alpha/2}}{2e^{\alpha/2}+2}\right) N_{(0,1,3/2,2),2}$




(b) $N_{(0,1,2,3),2} = \left(\frac{e^{\alpha/2}+2}{2e^{\alpha/2}+2}\right) N_{(0,1,3/2,2),2} + \left(\frac{2e^{\alpha/2}+1}{2e^{\alpha/2}+2}\right) N_{(1,3/2,2,3),2}$

Figure 3: Inserting a knot $\hat{\xi} = 3/2$ for different TB-splines related to the ECT-space $\mathbb{P}_2^{(10)} = \langle 1, x, e^{10x} \rangle$. Each existing spline is described as a combination of the refined TB-splines with their corresponding scaling factors as given in (11).

*by inserting the knot* $\hat{\xi} = 3/2$. *The knot insertion formula* (11) *gives*

$$N_{(0,0,1,2),2} = N_{(0,0,1,3/2),2} + \left(\frac{e^{\alpha/2}}{2e^{\alpha/2}+2}\right) N_{(0,1,3/2,2),2}.$$

*Figure 3(a) illustrates the original TB-spline split into two new TB-splines after knot insertion. Figure 3(b) exhibits another example where, as a result of knot insertion at* $\hat{\xi} = 3/2$, *the TB-spline* $N_{(0,1,2,3),2}$ *splits as*

$$N_{(0,0,1,2),2} = \left(\frac{e^{\alpha/2}+2}{2e^{\alpha/2}+2}\right) N_{(0,0,1,3/2),2} + \left(\frac{2e^{\alpha/2}+1}{2e^{\alpha/2}+2}\right) N_{(0,1,3/2,2),2}.$$

Since we are interested in spline spaces of maximal smoothness, from now on we only consider the case where the inserted knot $\hat{\xi}$ does not coincide with any existing knot in $\Xi$.

**Remark 7.** *Let* $\mathcal{S}_p(\mathcal{M})$ *be the set of TB-splines spanning the Tchebycheffian spline space* $\mathbb{S}_p(\mathcal{M})$ *on the mesh* $\mathcal{M}$. *From* (10) *it follows that the cardinality of* $\mathcal{S}_p(\mathcal{M})$ *equals n. Let* $\hat{\mathcal{M}}$ *be the refined mesh obtained from* $\mathcal{M}$ *by adding* $\hat{\xi}$ *as additional breakpoint in the interior of the interval* $[a, b]$. *On this refined mesh, the new set of TB-splines can be constructed by the following procedure.*

1. *Initialize the set by* $\mathcal{S}_p(\hat{\mathcal{M}}) \leftarrow \mathcal{S}_p(\mathcal{M})$.
2. *As long as there exists* $N_{(\xi_{k,1},\ldots,\xi_{k,p+2}),p} \in \mathcal{S}_p(\hat{\mathcal{M}})$ *with* $\hat{\xi} \in [\xi_{k,1}, \xi_{k,p+2}]$ *but* $\hat{\xi} \notin \{\xi_{k,1}, \ldots, \xi_{k,p+2}\}$ *:*
   (a) *Apply knot insertion as in* (11)*:*

$$N_{(\xi_{k,1},\ldots,\xi_{k,p+2}),p}(x) = \nu_k^{(1)} N_{(\xi_{k,1},\ldots,\hat{\xi},\ldots,\xi_{k,p+1}),p}(x) + \nu_k^{(2)} N_{(\xi_{k,2},\ldots,\hat{\xi},\ldots,\xi_{k,p+2}),p}(x).$$

   (b) *Update the set:*

$$\mathcal{S}_p(\hat{\mathcal{M}}) \leftarrow \left(\mathcal{S}_p(\hat{\mathcal{M}}) \setminus \{N_{(\xi_{k,1},\ldots,\xi_{k,p+2}),p}\}\right) \cup \left\{N_{(\xi_{k,1},\ldots,\hat{\xi},\ldots,\xi_{k,p+1}),p}, N_{(\xi_{k,2},\ldots,\hat{\xi},\ldots,\xi_{k,p+2}),p}\right\}.$$

*It can be verified that the obtained set* $\mathcal{S}_p(\hat{\mathcal{M}})$ *spans the* $(n+1)$*-dimensional Tchebycheffian spline space* $\mathbb{S}_p(\hat{\mathcal{M}})$ *on the refined mesh* $\hat{\mathcal{M}}$ *and forms actually the TB-spline basis of this space. The above procedure is a simplified version of the LR TB-spline construction that will be discussed in Section 3 (see Definition 13).*

### 2.5. Multivariate setting

Let $\boldsymbol{x} := (x_1, \dots, x_d)$. The local representation of univariate TB-splines can be easily extended to the multivariate case by taking their tensor product. More precisely, given degrees $\boldsymbol{p} := (p_1, \dots, p_d)$ and root vectors $\mathcal{w}_1, \mathcal{w}_2, \dots, \mathcal{w}_d$ as in (3), we define the multivariate TB-spline $N_{\boldsymbol{\Xi}_k, \boldsymbol{p}} : \mathbb{R}^d \to \mathbb{R}$, with local knot vectors

$$\boldsymbol{\Xi}_k := \left( \Xi^1_{k_1}, \Xi^2_{k_2}, \dots, \Xi^d_{k_d} \right),$$

as

$$N_{\boldsymbol{\Xi}_k, \boldsymbol{p}}(\boldsymbol{x}) := \prod_{i=1}^{d} N_{\Xi^i_{k_i}, p_i}(x_i), \tag{12}$$

where $N_{\Xi^i_{k_i}, p_i}$ is a univariate TB-spline related to the ECT-space $\mathbb{P}^{\mathcal{W}_i}_{p_i}$ of degree $p_i$ and root vector $\mathcal{w}_i$ as in (4) for $i = 1, \dots, d$. The function $N_{\boldsymbol{\Xi}_k, \boldsymbol{p}}$ belongs piecewisely to the space $\mathbb{P}^{\mathcal{W}_1}_{p_1} \otimes \mathbb{P}^{\mathcal{W}_2}_{p_2} \otimes \cdots \otimes \mathbb{P}^{\mathcal{W}_d}_{p_d}$ and its support is given by the Cartesian product of the supports of the local knot vectors in each direction as

$$\operatorname{supp}(N_{\boldsymbol{\Xi}_k, \boldsymbol{p}}) := [\xi^1_{k_1,1}, \xi^1_{k_1, p_1+2}] \times \cdots \times [\xi^d_{k_d,1}, \xi^d_{k_d, p_d+2}].$$

In the following we restrict our attention to the bivariate case $d = 2$. Expanding the concept of knot insertion from a univariate to a bivariate TB-spline can be intuitively compared to inserting an axis-aligned line. The sole necessary condition for the inserted line is that it must split the complete support of the TB-spline undergoing the refinement process. The bivariate TB-spline $N_{\boldsymbol{\Xi}_k, \boldsymbol{p}}$, with local knot vector

$$\boldsymbol{\Xi}_k = \left( \Xi^1_{k_1}, \Xi^2_{k_2} \right) = \left( (\xi^1_{k_1,1}, \dots, \xi^1_{k_1, p_1+2}), (\xi^2_{k_2,1}, \dots, \xi^2_{k_2, p_2+2}) \right),$$

can be refined by inserting, e.g., a vertical line $\{\hat{\xi}\} \times [\xi^2_{k_2,1}, \xi^2_{k_2, p_2+2}]$, with $\hat{\xi} \in (\xi^1_{k_1,1}, \xi^1_{k_1, p_1+2})$. Analogous to knot insertion, the line insertion results in splitting the given TB-spline in two new TB-splines as

$$\begin{aligned} N_{\boldsymbol{\Xi}_k, \boldsymbol{p}}(\boldsymbol{x}) &= N_{(\xi^1_{k_1,1}, \dots, \xi^1_{k_1, p_1+2}), p_1}(x_1) N_{(\xi^2_{k_2,1}, \dots, \xi^2_{k_2, p_2+2}), p_2}(x_2) \\ &= \left( \nu^{(1)}_k N_{(\xi^1_{k_1,1}, \dots, \hat{\xi}, \dots, \xi^1_{k_1, p_1+1}), p_1}(x_1) + \nu^{(2)}_k N_{(\xi^1_{k_1,2}, \dots, \hat{\xi}, \dots, \xi^1_{k_1, p_1+2}), p_1}(x_1) \right) N_{(\xi^2_{k_2,1}, \dots, \xi^2_{k_2, p_2+2}), p_2}(x_2) \\ &=: \nu^{(1)}_k N_{\boldsymbol{\Xi}_{l_1}, \boldsymbol{p}}(\boldsymbol{x}) + \nu^{(2)}_k N_{\boldsymbol{\Xi}_{l_2}, \boldsymbol{p}}(\boldsymbol{x}). \end{aligned} \tag{13}$$

The non-negative scaling factors $\nu^{(1)}_k, \nu^{(2)}_k \in [0, 1]$ are computed in the same manner as in (11). It is clear that the scaling relation for insertion of a horizontal line is alike.

## 3. TB-splines on LR-meshes

In this section, we discuss the construction of TB-splines on locally refined meshes (LR-meshes), which are particular axis-aligned box-partitions. First, we explore how these meshes can be obtained by means of successive line insertion. Subsequently, we define LR TB-splines as an extension of polynomial LR B-splines. We refer the reader to [12] for in-depth information about splines over locally refined box-partitions. Our presentation closely follows [30, Section 2], extending to the Tchebycheff setting the concepts presented therein.

### 3.1. LR-meshes

Throughout this paper we keep the description of LR-meshes limited to the bivariate case. Note that several definitions we are going to state are simplified for an easier reading because we confine ourselves to the case of maximal smoothness. However, the Tchebycheff setting allows for full generality as in the algebraic polynomial case.

**Definition 8** (Box-partition and mesh)**.** Given an axis-aligned rectangle $\Omega \subseteq \mathbb{R}^2$, a box-partition of $\Omega$ is a finite collection $\mathcal{E}$ of axis-aligned rectangles, called elements $\sigma$, such that

- $\bigcup_{\sigma\in\mathcal{E}} \sigma = \Omega$,
- $\mathring{\sigma}_1 \cap \mathring{\sigma}_2 = \emptyset$ for any $\sigma_1, \sigma_2 \in \mathcal{E}$, with $\sigma_1 \neq \sigma_2$,

where the interior of an element $\sigma$ is denoted by $\mathring{\sigma}$. We define the set $\mathcal{V}$ consisting of all vertices from all the elements of $\mathcal{E}$ as the vertices of $\mathcal{E}$. A meshline $\gamma$ of $\mathcal{E}$ is an axis-aligned segment contained in an element edge, connecting two and only two vertices of $\mathcal{V}$ at its end-points. The set $\mathcal{M}$ of all the meshlines $\gamma$ of $\mathcal{E}$ is called the mesh of $\mathcal{E}$.

We can represent a meshline $\gamma$ as the Cartesian product of a point in $\mathbb{R}$ and a finite interval. For example, a vertical meshline can be described as $\{\xi_i^1\} \times [\xi_j^2, \xi_{j+1}^2]$, while a horizontal meshline can be described as $[\xi_i^1, \xi_{i+1}^1] \times \{\xi_j^2\}$.

**Definition 9** ($\mu$-extended mesh)**.** Given a mesh $\mathcal{M}$ and bi-degree $\boldsymbol{p} = (p_1, p_2)$, the meshline multiplicity function $\mu : \mathcal{M} \mapsto \mathbb{Z}_{>0}$ is a function that associates a positive integer with every meshline, called multiplicity of the meshline. A mesh that has an assigned multiplicity function $\mu$ is called $\mu$-extended mesh. The meshline multiplicity is assumed to be maximally $p_k + 1$ in its corresponding direction $k$. A mesh $\mathcal{M}$ is considered an open mesh if every boundary meshline has maximal multiplicity. Moreover, when all the meshlines of the box-partition $\mathcal{E}$ have the same multiplicity $\mu$ except on the boundary of the domain, we can say that the mesh $\mathcal{M}$ has multiplicity $\mu$.

In this paper we fix the multiplicity of all meshlines to one except on the boundary, where it is fixed to maximum. Hence, all the considered meshes will be of multiplicity $\mu = 1$ and on the boundaries they will be open. Figure 4 illustrates an example of a box-partition $\mathcal{E}$ of $\Omega$ and related $\mu$-extended mesh $\mathcal{M}$.

A vertex of $\mathcal{E}$ is called T-vertex if it is the intersection of two collinear meshlines and a single orthogonal meshline. A tensor mesh is a special case of $\mu$-extended mesh, where the T-vertices occur only on the boundaries of the domain and all collinear meshlines have the same multiplicity $\mu$. We can describe any bivariate tensor mesh on two knot vectors $\boldsymbol{\Xi} = ((\xi_1^1, \ldots, \xi_s^1), (\xi_1^2, \ldots, \xi_t^2))$ as

$$\begin{aligned}\mathcal{M}(\boldsymbol{\Xi}) \coloneqq \Big\{ &\{\xi_i^1\} \times [\xi_j^2, \xi_{j+1}^2] : i = 1, \ldots, s;\ j = 1, \ldots, t-1\} \\ &\cup [\xi_i^1, \xi_{i+1}^1] \times \{\xi_j^2\} : i = 1, \ldots, s-1;\ j = 1, \ldots, t\} \Big\}.\end{aligned}$$

The multiplicities of the meshlines in $\mathcal{M}(\boldsymbol{\Xi})$ are given by the multiplicities of the knots in $\boldsymbol{\Xi}$.

**Definition 10** (Support)**.** Given a mesh $\mathcal{M}$ and a bivariate TB-spline $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}$, we say that $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}$ has support on $\mathcal{M}$ if

- the meshlines in $\mathcal{M}(\boldsymbol{\Xi}_k)$ can be obtained as the union of meshlines in $\mathcal{M}$, and
- the multiplicities of the meshlines in $\mathcal{M}(\boldsymbol{\Xi}_k)$ are less than or equal to the multiplicities of the corresponding meshlines in $\mathcal{M}$.

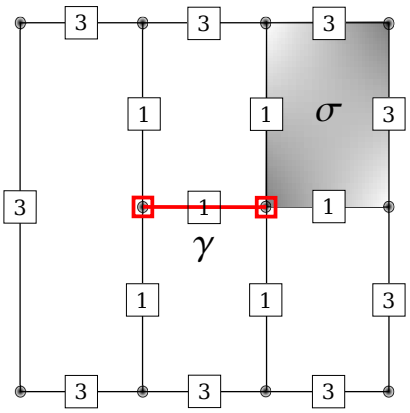
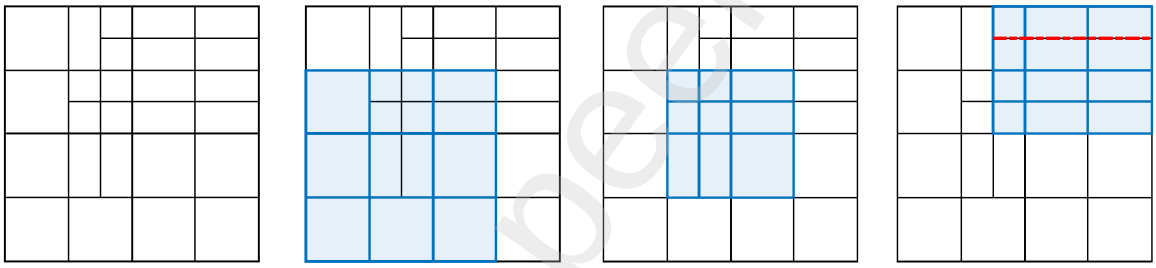



Figure 4: A box-partition $\mathcal{E}$ of the domain $\Omega$ with elements $\sigma$ and meshlines $\gamma$. The vertices of $\mathcal{V}$ are marked with circles. The $\mu$-extended mesh $\mathcal{M}$ of $\mathcal{E}$ for bi-degree $\boldsymbol{p} = (2, 2)$ assumes that the boundary meshlines have maximal multiplicity ($\mu = 3$) and all internal meshlines have a fixed multiplicity equal to $\mu = 1$. The multiplicities associated with all meshlines are depicted in the squares along the lines.



(a) (b) (c) (d)

Figure 5: Some supports of TB-splines of bi-degree $\boldsymbol{p} = (2, 2)$ on the given mesh $\mathcal{M}$ in (a) with multiplicity $\mu = 1$. The TB-splines in (b) and (c) have minimal support on $\mathcal{M}$, with their knots highlighted by thicker lines. The TB-spline in (d) does not have minimal support on $\mathcal{M}$, as the dashed line disconnects its support.

Furthermore, we say that $N_{\Xi_k,\boldsymbol{p}}$ has minimal support if

- the multiplicities of the interior meshlines in $\mathcal{M}(\Xi_k)$ are equal to the multiplicities of the corresponding meshlines in $\mathcal{M}$, and
- there is no collection of collinear meshlines $\gamma$ in $\mathcal{M} \setminus \mathcal{M}(\Xi_k)$ such that $\text{supp}(N_{\Xi_k,\boldsymbol{p}}) \setminus \gamma$ is not connected.

Figure 5 illustrates some examples of TB-splines of bi-degree $\boldsymbol{p} = (2, 2)$ with minimal support on a mesh $\mathcal{M}$ with fixed multiplicity $\mu = 1$. Note that not all TB-splines that have support on $\mathcal{M}$ have minimal support on $\mathcal{M}$.

**Definition 11** (Split)**.** Given a box-partition $\mathcal{E}$ and an axis-aligned segment $\gamma$, we say that $\gamma$ traverses $\sigma \in \mathcal{E}$ if $\gamma \subset \sigma$ and the interior of $\sigma$ is divided into two parts by $\gamma$, i.e., $\sigma \setminus \gamma$ is not connected. A split is defined as a finite union of contiguous and collinear axis-aligned segments $\gamma = \cup_i \gamma_i$ such that every $\gamma_i$ is either a meshline of $\mathcal{E}$ or it traverses some $\sigma \in \mathcal{E}$.

LR-meshes are meshes obtained from a tensor mesh by successively inserting (sufficiently long) splits and LR TB-splines are defined as a specific set of TB-splines with minimal support on an LR-mesh. When

a split $\gamma$ is inserted into a box-partition $\mathcal{E}$, any traversed rectangle $\sigma \in \mathcal{E}$ is replaced with two sub-rectangles given by the closures of the connected components of $\sigma \setminus \gamma$. The resulting new box-partition will be denoted by $\mathcal{E}+\gamma$ and its corresponding mesh by $\mathcal{M}+\gamma$. We focus only on meshes of multiplicity $\mu = 1$ that are open on the boundaries. Hence, when we refine such a mesh $\mathcal{M}$ with a split $\gamma$, the multiplicities of the meshlines shared by both $\mathcal{M}$ and $\mathcal{M}+\gamma$ remain the same, while multiplicity one is assigned to the new meshlines in $\gamma$.

**Definition 12** (Traversal of a TB-spline)**.** Given an open mesh $\mathcal{M}$ of multiplicity $\mu = 1$, a TB-spline $N_{\Xi_k,\boldsymbol{p}}$ with support on $\mathcal{M}$, and a split $\gamma$, we say that $\gamma$ traverses $N_{\Xi_k,\boldsymbol{p}}$ if the interior of $\text{supp}(N_{\Xi_k,\boldsymbol{p}})$ is divided in two disjoint parts by $\gamma$, i.e., $\text{supp}(N_{\Xi_k,\boldsymbol{p}}) \setminus \gamma$ is not connected, and $\gamma$ is in $\mathcal{M} \setminus \mathcal{M}(\Xi_k)$.

### *3.2. LR TB-splines*

At this point we have all fundamentals to construct LR TB-splines. We start with an open tensor mesh and the corresponding set of tensor-product TB-splines. Then, we refine the mesh by inserting splits, one at a time, and whenever a TB-spline in the considered set has no longer minimal support during the mesh refinement process, we refine it by using the knot insertion procedure.

**Definition 13** (LR TB-splines)**.** Let $\mathcal{M}_0$ be an open tensor mesh of multiplicity $\mu = 1$ and let $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_0)$ be the corresponding set of tensor-product TB-splines of bi-degree $\boldsymbol{p}$ (and maximal smoothness) on $\mathcal{M}_0$. We then define a sequence of meshes $\mathcal{M}_1, \mathcal{M}_2, \ldots$ of multiplicity $\mu = 1$ and a sequence of sets of TB-splines $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_1), \mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_2), \ldots$ as follows. For $j = 1, 2, \ldots$, let $\gamma_j$ be a split such that $\mathcal{M}_j = \mathcal{M}_{j-1} + \gamma_j$ and such that at least one TB-spline in $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_{j-1})$ is traversed by a split in $\mathcal{M}_j$. On this refined mesh $\mathcal{M}_j$ the new set of TB-splines is constructed by the following procedure.

1. Initialize the set by $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j) \leftarrow \mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_{j-1})$.
2. As long as there exists $N_{\Xi_k,\boldsymbol{p}} \in \mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)$ with no minimal support on $\mathcal{M}_j$ :
   (a) Apply knot insertion as in (13) : $N_{\Xi_k,\boldsymbol{p}} = \nu_k^{(1)} N_{\Xi_{l_1},\boldsymbol{p}} + \nu_k^{(2)} N_{\Xi_{l_2},\boldsymbol{p}}$.
   (b) Update the set: $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j) \leftarrow \left(\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j) \setminus N_{\Xi_k,\boldsymbol{p}}\right) \cup \left\{N_{\Xi_{l_1},\boldsymbol{p}}, N_{\Xi_{l_2},\boldsymbol{p}}\right\}$.

The generated mesh $\mathcal{M}_j$ is referred to as an LR-mesh and the corresponding set $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)$ is designated as a set of LR TB-splines.

**Remark 14.** *The splitting procedure described in Step 2 of Definition 13 to construct a set of TB-splines on the refined mesh $\mathcal{M}_j$ (obtained by inserting split $\gamma_j$ into mesh $\mathcal{M}_{j-1}$) can be subdivided into two steps.*

- *Primary split: Refine any existing TB-spline whose support is traversed by $\gamma_j$ or by the union of $\gamma_j$ and any existing split in $\mathcal{M}_{j-1}$ that intersects and is collinear with $\gamma_j$.*
- *Secondary split: Refine any newly created TB-spline whose support is traversed by any existing split in $\mathcal{M}_{j-1}$.*

*After the primary split step, the supports of the newly created TB-splines are expected to be smaller than the original ones. This, in turn, opens up the possibility that existing splits in $\mathcal{M}_{j-1}$ might traverse these new TB-splines. Therefore, the secondary split step is designed to check for this possibility. If there is such a split, the secondary split step is executed once more, and this process continues until there are no remaining new TB-splines that are traversed by any split in $\mathcal{M}_{j-1}$. This implementation of LR TB-splines is summarized in Algorithm 1 in Appendix A.*

**Remark 15.** *Any refinement sequence producing a given LR-mesh $\mathcal{M}_j$ is not inherently unique, as the order of split insertions can often be altered. Nonetheless, the set of LR TB-splines $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)$ is well defined on $\mathcal{M}_j$ because it is independent of such insertion ordering, as proved in [12, Theorem 3.4] for polynomial LR B-splines but the proof also extends to LR TB-splines. Therefore, once the set of LR TB-splines on $\mathcal{M}_j$ has been constructed, it is safe to discard any data linked to the previous iteration, including the mesh $\mathcal{M}_{j-1}$.*

As in the polynomial case, due to the local refinement in new LR TB-splines, the partition of unity property is lost. This property is essential for interpreting the LR TB-spline coefficients as control points and ensuring the convex hull property. Restoration of the partition of unity in the new set of LR TB-splines can be achieved by a proper scaling, resulting in the so-called weighted LR TB-splines.

**Definition 16** (Weighted LR TB-splines)**.** Given a set of LR TB-splines $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)$ constructed according to Definition 13, the set of weighted LR TB-splines satisfies

$$\sum_{N_{\boldsymbol{\Xi}_k,\boldsymbol{p}} \in \mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)} N^{\mathcal{S}}_{\boldsymbol{\Xi}_k,\boldsymbol{p}}(\boldsymbol{x}) := \sum_{N_{\boldsymbol{\Xi}_k,\boldsymbol{p}} \in \mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)} \varsigma_k N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}(\boldsymbol{x}) = 1, \quad \boldsymbol{x} \in \Omega,$$

for some positive weights $\varsigma_k \in \mathbb{R}$, defined as follows for $j = 0, 1, 2, \ldots$. The initial weights corresponding to the tensor-product TB-splines on the tensor mesh $\mathcal{M}_0$ are set to $\varsigma_k = 1$. When we advance refining $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}$ (with weight $\varsigma_k$) as in Step 2 of Definition 13, the weights $\varsigma_{l_1}$ and $\varsigma_{l_2}$ corresponding to the refined TB-splines $N_{\boldsymbol{\Xi}_{l_1},\boldsymbol{p}}$ and $N_{\boldsymbol{\Xi}_{l_2},\boldsymbol{p}}$ on $\mathcal{M}_j$ are updated as

$$\varsigma_{l_i} \leftarrow \varsigma_k^{(i)} := \begin{cases} \nu_k^{(i)} \varsigma_k, & \text{if } N_{\boldsymbol{\Xi}_{l_i},\boldsymbol{p}} \notin \mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j), \\ \varsigma_{l_i} + \nu_k^{(i)} \varsigma_k, & \text{if } N_{\boldsymbol{\Xi}_{l_i},\boldsymbol{p}} \in \mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j), \end{cases} \quad i = 1, 2. \tag{14}$$

Due to the structural similarity between the Tchebycheff and the polynomial setting (see Proposition 4), the LR TB-splines considered in Definitions 13 and 16 enjoy the same properties as polynomial LR B-splines; see [7, 12]. In particular, from their construction it follows immediately that they are non-negative, have minimal support, and the weighted LR TB-splines sum up to one. Unfortunately, LR (T)B-splines are not always linearly independent. Extensive studies have been conducted to investigate the linear dependency for polynomial LR B-splines, and several refinement strategies have been proposed to address this issue; see, e.g., [28, 29, 30]. Similar strategies are also applicable for TB-splines on LR-meshes.

Figure 6 shows some bi-quadratic LR TB-splines obtained from the local space $\mathbb{P}_2^{(50)} \otimes \mathbb{P}_2^{(50)} = \left\langle 1, x_1, e^{50x_1} \right\rangle \otimes \left\langle 1, x_2, e^{50x_2} \right\rangle$ on a given LR-mesh $\mathcal{M}$.

### *3.3. Constructive example*

We replicate the example presented in [18, Section 2.2.4] for the local space $\mathbb{P}_2^{(10)} \otimes \mathbb{P}_2^{(10)} = \left\langle 1, x_1, e^{10x_1} \right\rangle \otimes \left\langle 1, x_2, e^{10x_2} \right\rangle$ of bi-degree $\boldsymbol{p} = (2, 2)$. We start from the open tensor mesh $\mathcal{M}_0$ given by the global knot vectors $\boldsymbol{\Xi} = ((0, 0, 0, 1, 2, 3, 4, 5, 5, 5), (0, 0, 0, 1, 2, 3, 4, 5, 5, 5))$; see Figure 7. All the tensor-product TB-splines in the set $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_0)$ have weight $\varsigma_k = 1$. In the example we insert two splits and we monitor the evolution of the weights corresponding to each TB-spline refinement according to Definition 16. Note that the associated knot insertion procedure in the univariate case has been elaborated in Example 6.

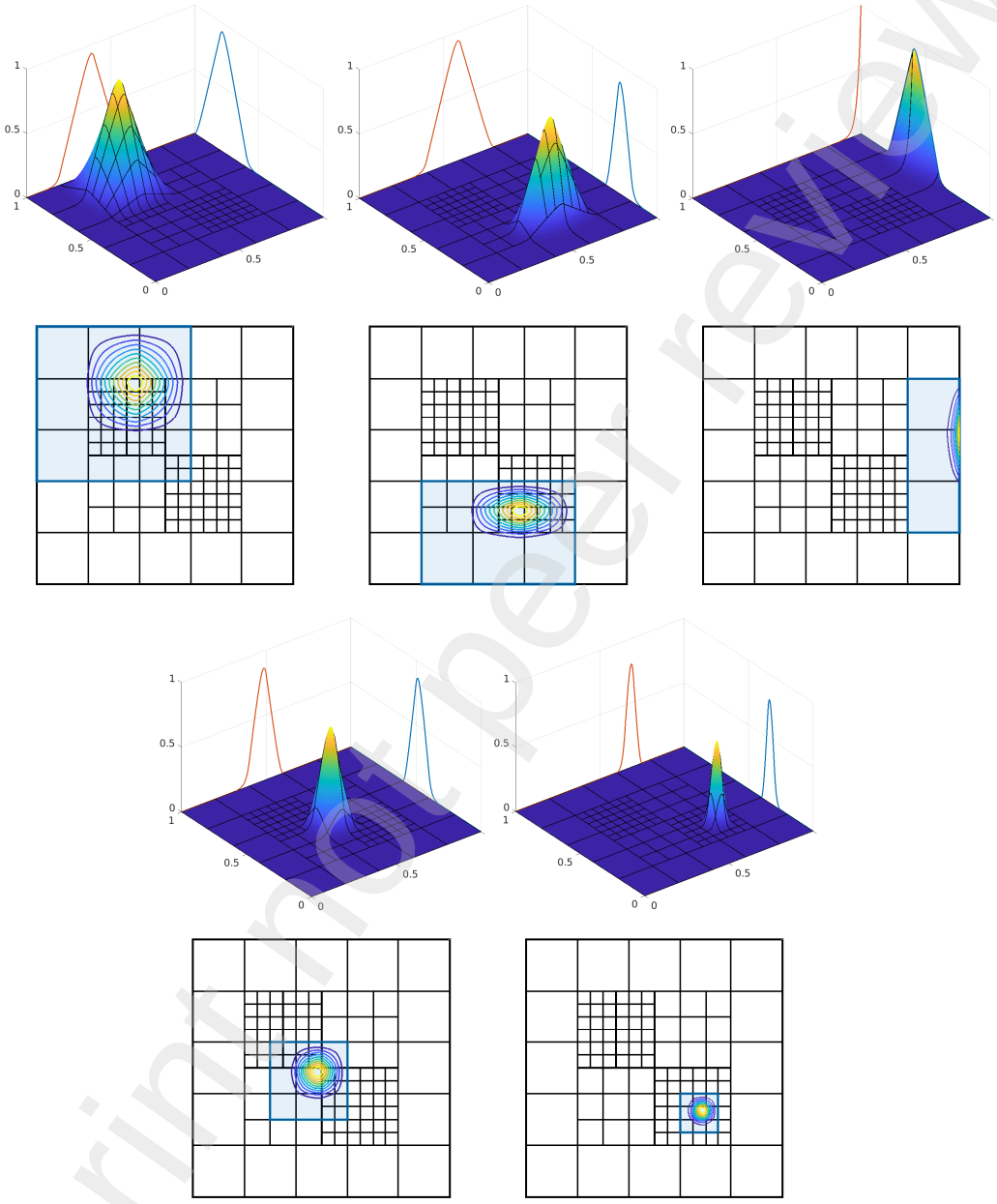


Figure 6: Some bi-quadratic LR TB-splines related to the local space $\mathbb{P}_2^{(50)} \otimes \mathbb{P}_2^{(50)}$ on a given LR-mesh $\mathcal{M}$. These splines, visualized along with their contour plots, are tensor products of univariate TB-splines, which are depicted on the coordinate planes.

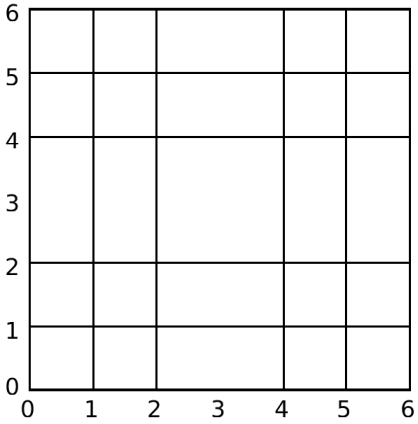



Figure 7: Initial tensor mesh $\mathcal{M}_0$ used to generate the TB-spline set $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_0)$ related to the local space $\mathbb{P}_2^{(10)} \otimes \mathbb{P}_2^{(10)}$ of bi-degree $\boldsymbol{p} = (2, 2)$.

First, we introduce a vertical split $\gamma_1 = \{3\} \times [1, 5]$ into $\mathcal{M}_0$, resulting in $\mathcal{M}_1$. This affects three TB-splines as illustrated in Figure 8, creating four new refined TB-splines,

$$
\begin{aligned}
N_{((0,1,2,4),(1,2,4,5)),(2,2)} &= N_{((0,1,2,3),(1,2,4,5)),(2,2)} + \left(\frac{e^\alpha}{2e^\alpha + 1}\right) N_{((1,2,3,4),(1,2,4,5)),(2,2)}, \\
N_{((1,2,4,5),(1,2,4,5)),(2,2)} &= \left(\frac{e^\alpha + 1}{2e^\alpha + 1}\right) N_{((1,2,3,4),(1,2,4,5)),(2,2)} + \left(\frac{e^\alpha + 1}{e^\alpha + 2}\right) N_{((2,3,4,5),(1,2,4,5)),(2,2)}, \\
N_{((2,4,5,6),(1,2,4,5)),(2,2)} &= \left(\frac{1}{e^\alpha + 2}\right) N_{((2,3,4,5),(1,2,4,5)),(2,2)} + N_{((3,4,5,6),(1,2,4,5)),(2,2)}.
\end{aligned}
\tag{15}
$$

As a result of the split of $N_{((0,1,2,4),(1,2,4,5)),(2,2)}$ in (15), we achieve two new TB-splines $N_{((0,1,2,3),(1,2,4,5)),(2,2)}$ and $N_{((1,2,3,4),(1,2,4,5)),(2,2)}$ with the scaling coefficients 1 and $\left(\frac{e^\alpha}{2e^\alpha+1}\right)$, respectively. Since both TB-splines are new and the weight of $N_{((0,1,2,4),(1,2,4,5)),(2,2)}$ from which they split is one, the resulting weight according to (14) is the same as their scaling coefficient as shown in Column 2 of Table 1. Next, the split of $N_{((1,2,4,5),(1,2,4,5)),(2,2)}$ in (15) results in one of the existing TB-splines $N_{((1,2,3,4),(1,2,4,5)),(2,2)}$ with scaling coefficient $\left(\frac{e^\alpha+1}{2e^\alpha+1}\right)$. Here the weight of the existing spline is updated according to (14) as presented in Column 3 in Table 1,

$$
\varsigma_2^{(1)} = \varsigma_1^{(2)} + \left(\frac{e^\alpha + 1}{2e^\alpha + 1}\right) \cdot \varsigma_2 = \left(\frac{e^\alpha}{2e^\alpha + 1}\right) + \left(\frac{e^\alpha + 1}{2e^\alpha + 1}\right) \cdot 1 = 1,
$$

where the existing weight of the TB-spline is $\left(\frac{e^\alpha}{2e^\alpha+1}\right)$. Similarly, with the split of $N_{((2,4,5,6),(1,2,4,5)),(2,2)}$ we complete the primary split step (see Remark 14), resulting in four new TB-splines and their corresponding weights as tabulated in Column 5 in Table 1.

Proceeding with the secondary split step (see Remark 14) in the refinement process, every new TB-spline listed in Column 1 in Table 1 is tested against traversal by any existing split in $\mathcal{M}_0$. However, there are no TB-splines traversed here, hence the set of all TB-splines after the primary split step is the final set $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_1)$.

Second, we insert a horizontal split $\gamma_2 = [1, 5] \times \{3\}$ into $\mathcal{M}_1$, resulting in $\mathcal{M}_2$. This affects four TB-

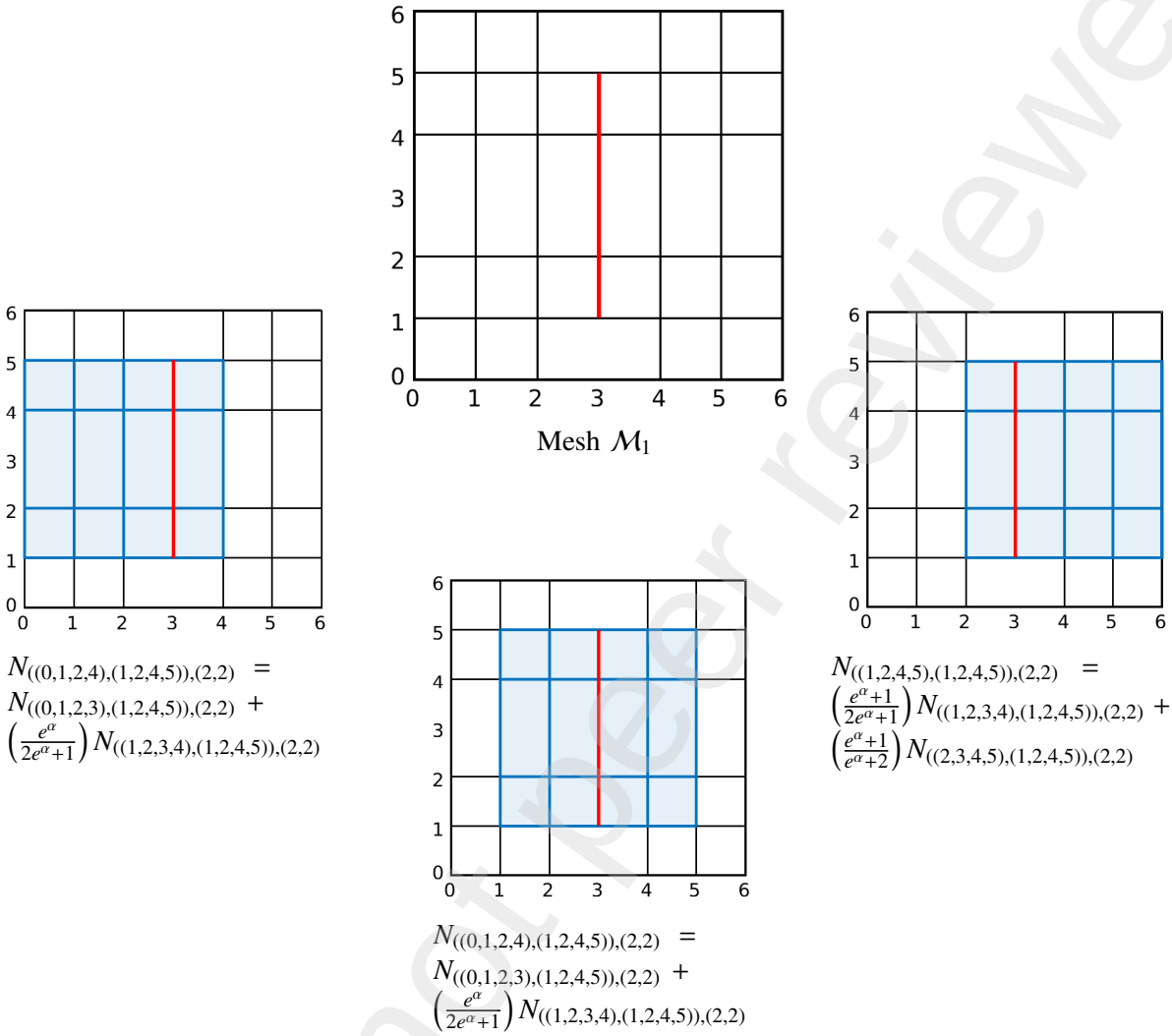



Figure 8: Split insertion (Iteration 1): The mesh $\mathcal{M}_1$ obtained by inserting a vertical split $\gamma_1 = \{3\} \times [1, 5]$ into the tensor mesh $\mathcal{M}_0$ illustrated in Figure 7, together with all three TB-splines traversed by $\gamma_1$. The supports of the TB-splines are highlighted and annotated with the corresponding scaling relation between the original TB-spline and the two new TB-splines into which it has been refined, with their scaling factors for the local space $\mathbb{P}_2^{(10)} \otimes \mathbb{P}_2^{(10)}$ of bi-degree $\boldsymbol{p} = (2, 2)$.

splines as illustrated in Figure 9, creating eight new refined TB-splines,

$$
\begin{aligned}
N_{((1,2,3,4),(1,2,4,5)),(2,2)} &= \left(\frac{e^\alpha+1}{2e^\alpha+1}\right) N_{((1,2,3,4),(1,2,3,4)),(2,2)} + \left(\frac{e^\alpha+1}{e^\alpha+2}\right) N_{((1,2,3,4),(2,3,4,5)),(2,2)}, \\
N_{((1,2,4,5),(0,1,2,4)),(2,2)} &= N_{((1,2,4,5),(0,1,2,3)),(2,2)} + \left(\frac{e^\alpha}{2e^\alpha+1}\right) N_{((1,2,4,5),(1,2,3,4)),(2,2)}, \\
N_{((1,2,4,5),(2,4,5,6)),(2,2)} &= \left(\frac{1}{e^\alpha+2}\right) N_{((1,2,4,5),(2,3,4,5)),(2,2)} + N_{((1,2,4,5),(3,4,5,6)),(2,2)}, \\
N_{((2,3,4,5),(1,2,4,5)),(2,2)} &= \left(\frac{e^\alpha+1}{2e^\alpha+1}\right) N_{((2,3,4,5),(1,2,3,4)),(2,2)} + \left(\frac{e^\alpha+1}{e^\alpha+2}\right) N_{((2,3,4,5),(2,3,4,5)),(2,2)}.
\end{aligned}
\tag{16}
$$

| | Primary split | | | Final |
|---|---|---|---|---|
| | $N^{\mathcal{S}}_{\Xi_1,\mathbf{2}} = N^{\mathcal{S}}_{((0,1,2,4),(1,2,4,5)),(2,2)}$<br>$\varsigma_1 = 1$ | $N^{\mathcal{S}}_{\Xi_2,\mathbf{2}} = N^{\mathcal{S}}_{((1,2,4,5),(1,2,4,5)),(2,2)}$<br>$\varsigma_2 = 1$ | $N^{\mathcal{S}}_{\Xi_3,\mathbf{2}} = N^{\mathcal{S}}_{((2,4,5,6),(1,2,4,5)),(2,2)}$<br>$\varsigma_3 = 1$ | $\varsigma_k$ |
| $N^{\mathcal{S}}_{((0,1,2,3),(1,2,4,5)),(2,2)}$ | $\varsigma_1^{(1)} = 1 \cdot \varsigma_1$ | | | 1 |
| $N^{\mathcal{S}}_{((1,2,3,4),(1,2,4,5)),(2,2)}$ | $\varsigma_1^{(2)} = \left(\frac{e^\alpha}{2e^\alpha+1}\right) \cdot \varsigma_1$ | $\varsigma_2^{(1)} = \varsigma_1^{(2)} + \left(\frac{e^\alpha+1}{2e^\alpha+1}\right) \cdot \varsigma_2$ | | 1 |
| $N^{\mathcal{S}}_{((2,3,4,5),(1,2,4,5)),(2,2)}$ | | $\varsigma_2^{(2)} = \left(\frac{e^\alpha+1}{e^\alpha+2}\right) \cdot \varsigma_2$ | $\varsigma_3^{(1)} = \varsigma_2^{(2)} + \left(\frac{1}{e^\alpha+2}\right) \cdot \varsigma_3$ | 1 |
| $N^{\mathcal{S}}_{((3,4,5,6),(1,2,4,5)),(2,2)}$ | | | $\varsigma_3^{(2)} = 1 \cdot \varsigma_3$ | 1 |

Table 1: Split insertion (Iteration 1): The evolution of the weights for the three TB-splines that are traversed by the split $\gamma_1$, converting $\mathcal{M}_0$ into $\mathcal{M}_1$ and leading to the creation of four new TB-splines, as illustrated in Figure 8. The scaling relations of the TB-splines that underwent splitting are given in (15). This iteration only involves the primary split step described in Remark 14 since no TB-splines undergo a secondary split.

Since the created TB-splines as a result of the primary split step are all new and different, their weights are the same as their scaling coefficients in (16). We list these eight TB-splines with their corresponding weights in Column 1 of Table 2. We now proceed with the secondary split step, where the new TB-splines are tested against traversal by any existing split in $\mathcal{M}_1$. It turns out that two of them are traversed by a split. These two TB-splines are depicted in Figure 10 and lead to

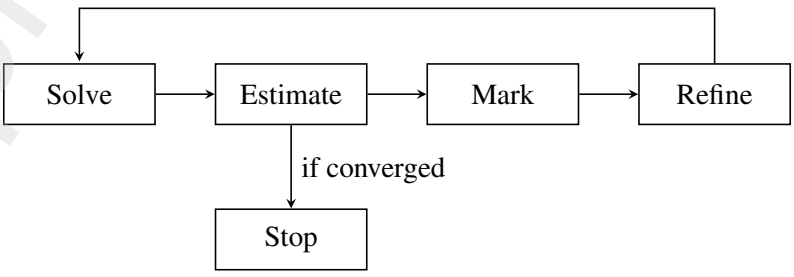
$$\begin{aligned} N_{((1,2,4,5),(1,2,3,4)),(2,2)} &= \left(\frac{e^\alpha+1}{2e^\alpha+1}\right) N_{((1,2,3,4),(1,2,3,4)),(2,2)} + \left(\frac{e^\alpha+1}{e^\alpha+2}\right) N_{((2,3,4,5),(1,2,3,4)),(2,2)}, \\ N_{((1,2,4,5),(2,3,4,5)),(2,2)} &= \left(\frac{e^\alpha+1}{2e^\alpha+1}\right) N_{((1,2,3,4),(2,3,4,5)),(2,2)} + \left(\frac{e^\alpha+1}{e^\alpha+2}\right) N_{((2,3,4,5),(2,3,4,5)),(2,2)}. \end{aligned} \tag{17}$$

After the secondary split step, we arrive at the final set $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_2)$. The resulting weights of the TB-splines in this set are tabulated in Table 2.

In conclusion, we want to emphasize that (15), (16), (17) present the two-scale relations for split insertion in an individual TB-spline, however, they do not take into account a possible existing weight of the TB-spline. The algorithmic evolution of the weights corresponding to each TB-spline after every split insertion is summarized in Tables 1 and 2.

## 4. Isogeometric analysis with LR TB-splines

In this section, we review the standard adaptive scheme for solving differential equations, which comprises the four (cyclic) key steps outlined below.



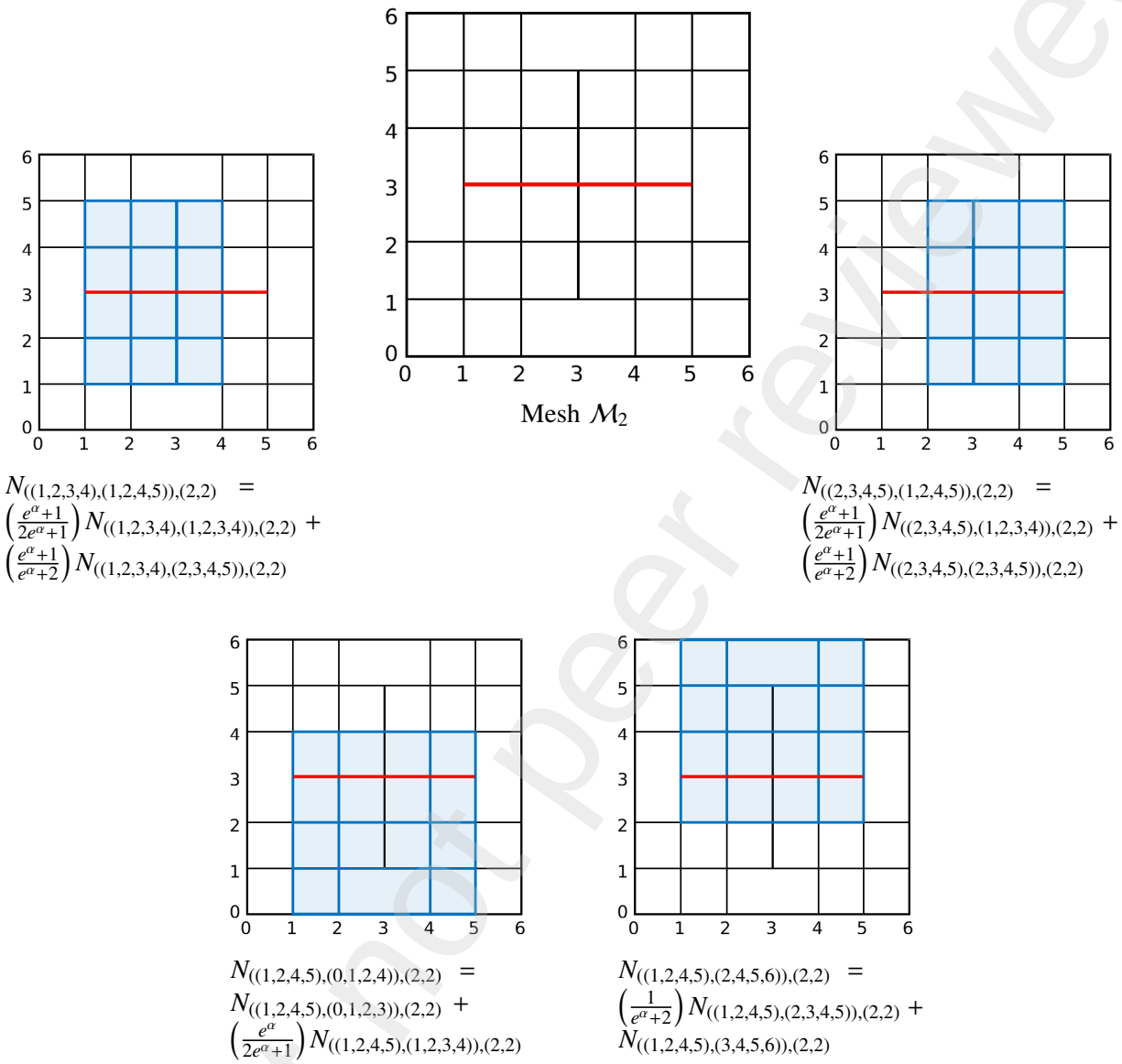
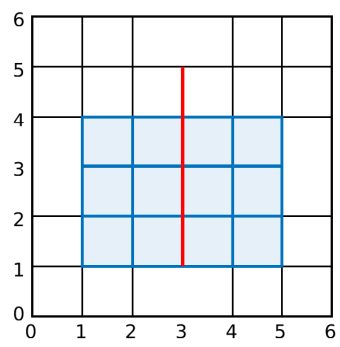
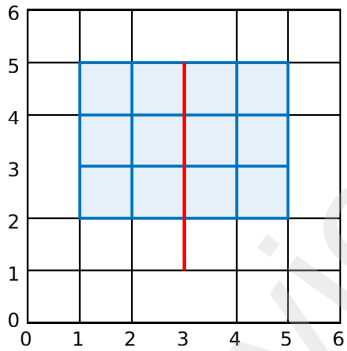



Figure 9: Split insertion (Iteration 2 - Primary split): The mesh $\mathcal{M}_2$ obtained by inserting a horizontal split $\gamma_2 = [1,5]\times\{3\}$ into the mesh $\mathcal{M}_1$ illustrated in Figure 8, together with all four TB-splines traversed by $\gamma_2$. The supports of the TB-splines are highlighted and annotated with the corresponding scaling relation between the original TB-spline and the two new TB-splines into which it has been refined, with their scaling factors for the local space $\mathbb{P}_2^{(10)} \otimes \mathbb{P}_2^{(10)}$ of bi-degree $\boldsymbol{p} = (2,2)$.

For the first step in the adaptive cycle, we consider the isogeometric Galerkin method based on LR TB-splines. For the sake of simplicity, we focus on second-order elliptic differential problems with homogeneous Dirichlet boundary conditions. Then, we introduce different marking strategies based on either an ad hoc approach or a residual-based error estimator, depending on the type of problem we are handling. Finally, we present the refinement strategy we have incorporated to refine the functions marked as the result of previous steps in the adaptive cycle.



$$N_{((1,2,4,5),(1,2,3,4)),(2,2)} = \left(\frac{e^\alpha+1}{2e^\alpha+1}\right) N_{((1,2,3,4),(1,2,3,4)),(2,2)} + \left(\frac{e^\alpha+1}{e^\alpha+2}\right) N_{((2,3,4,5),(1,2,3,4)),(2,2)}$$



$$N_{((1,2,4,5),(2,3,4,5)),(2,2)} = \left(\frac{e^\alpha+1}{2e^\alpha+1}\right) N_{((1,2,3,4),(2,3,4,5)),(2,2)} + \left(\frac{e^\alpha+1}{e^\alpha+2}\right) N_{((2,3,4,5),(2,3,4,5)),(2,2)}$$

Figure 10: Split insertion (Iteration 2 - Secondary split): After the primary split step on the mesh $\mathcal{M}_2$ as depicted in Figure 9, eight new TB-splines have been created of which two are further traversed by $\gamma_1 = \{3\} \times [1, 5]$. The supports of these two TB-splines are highlighted and annotated with the corresponding scaling relation between the original TB-spline and the two new TB-splines into which it has been refined, with their scaling factors for the local space $\mathbb{P}_2^{(10)} \otimes \mathbb{P}_2^{(10)}$ of bi-degree $\boldsymbol{p} = (2, 2)$.

| Primary split | | Secondary split | | |
|---|---|---|---|---|
| $N^{\mathcal{S}}_{((1,2,3,4),(1,2,4,5)),(2,2)}, N^{\mathcal{S}}_{((1,2,4,5),(0,1,2,4)),(2,2)}$ | | $N^{\mathcal{S}}_{\Xi_4,\mathbf{2}} = N^{\mathcal{S}}_{((1,2,4,5),(1,2,3,4)),(2,2)}$ | $N^{\mathcal{S}}_{\Xi_5,\mathbf{2}} = N^{\mathcal{S}}_{((1,2,4,5),(2,3,4,5)),(2,2)}$ | Final |
| $N^{\mathcal{S}}_{((1,2,4,5),(2,4,5,6)),(2,2)}, N^{\mathcal{S}}_{((2,3,4,5),(1,2,4,5)),(2,2)}$ | | $\varsigma_4 = \left(\frac{e^\alpha}{2e^\alpha+1}\right)$ | $\varsigma_5 = \left(\frac{1}{e^\alpha+2}\right)$ | $\varsigma_k$ |
| $N^{\mathcal{S}}_{\Xi_1,\mathbf{2}} = N^{\mathcal{S}}_{((1,2,3,4),(1,2,3,4)),(2,2)}$ | $\varsigma_1 = \frac{e^\alpha+1}{2e^\alpha+1}$ | $\varsigma_4^{(1)} = \varsigma_1 + \left(\frac{e^\alpha+1}{2e^\alpha+1}\right) \cdot \varsigma_4$ | | $\frac{3e^{2\alpha}+4e^\alpha+1}{(2e^\alpha+1)^2}$ |
| $N^{\mathcal{S}}_{\Xi_2,\mathbf{2}} = N^{\mathcal{S}}_{((1,2,3,4),(2,3,4,5)),(2,2)}$ | $\varsigma_2 = \frac{e^\alpha+1}{e^\alpha+2}$ | | $\varsigma_5^{(1)} = \varsigma_2 + \left(\frac{e^\alpha+1}{2e^\alpha+1}\right) \cdot \varsigma_5$ | $\frac{2(e^\alpha+1)^2}{2e^{2\alpha}+5e^\alpha+2}$ |
| $N^{\mathcal{S}}_{\Xi_3,\mathbf{2}} = N^{\mathcal{S}}_{((1,2,4,5),(0,1,2,3)),(2,2)}$ | $\varsigma_3 = 1$ | | | $1$ |
| $N^{\mathcal{S}}_{\Xi_4,\mathbf{2}} = N^{\mathcal{S}}_{((1,2,4,5),(1,2,3,4)),(2,2)}$ | $\varsigma_4 = \frac{e^\alpha}{2e^\alpha+1}$ | Remove $N^{\mathcal{S}}_{\Xi_4,\mathbf{2}}$ | | $0$ |
| $N^{\mathcal{S}}_{\Xi_5,\mathbf{2}} = N^{\mathcal{S}}_{((1,2,4,5),(2,3,4,5)),(2,2)}$ | $\varsigma_5 = \frac{1}{e^\alpha+2}$ | | Remove $N^{\mathcal{S}}_{\Xi_5,\mathbf{2}}$ | $0$ |
| $N^{\mathcal{S}}_{\Xi_6,\mathbf{2}} = N^{\mathcal{S}}_{((1,2,4,5),(3,4,5,6)),(2,2)}$ | $\varsigma_6 = 1$ | | | $1$ |
| $N^{\mathcal{S}}_{\Xi_7,\mathbf{2}} = N^{\mathcal{S}}_{((2,3,4,5),(1,2,3,4)),(2,2)}$ | $\varsigma_7 = \frac{e^\alpha+1}{2e^\alpha+1}$ | $\varsigma_4^{(2)} = \varsigma_7 + \left(\frac{e^\alpha+1}{e^\alpha+2}\right) \cdot \varsigma_4$ | | $\frac{2(e^\alpha+1)^2}{2e^{2\alpha}+5e^\alpha+2}$ |
| $N^{\mathcal{S}}_{\Xi_8,\mathbf{2}} = N^{\mathcal{S}}_{((2,3,4,5),(2,3,4,5)),(2,2)}$ | $\varsigma_8 = \frac{e^\alpha+1}{e^\alpha+2}$ | | $\varsigma_5^{(2)} = \varsigma_8 + \left(\frac{e^\alpha+1}{e^\alpha+2}\right) \cdot \varsigma_5$ | $\frac{e^{2\alpha}+4e^\alpha+3}{(e^\alpha+2)^2}$ |

Table 2: Split insertion (Iteration 2): The evolution of the weights for the four TB-splines that are traversed by the split $\gamma_2$, converting $\mathcal{M}_1$ into $\mathcal{M}_2$. The primary split step creates eight new TB-splines, as illustrated in Figure 9. The scaling relations of the TB-splines that underwent splitting are given in (16), and the resulting weights are tabulated in the first column. Subsequently, when the newly created TB-splines undergo a secondary split, two TB-splines are further divided as described by the scaling relations given in (17) and illustrated in Figure 10. This table serves to illustrate the evolving weights of TB-splines across consecutive refinement steps.

### 4.1. Isogeometric Tchebycheffian Galerkin method

Let $\mathfrak{L}$ be a linear second-order elliptic differential operator on the domain $\Omega \subset \mathbb{R}^d$ with Lipschitz boundary $\partial\Omega$. We consider the differential problem

$$\begin{cases} \mathfrak{L}u = \mathrm{f}, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \tag{18}$$

whose variational formulation reads as:

$$\text{find } u \in \mathbb{V} \quad \text{such that} \quad a(u, v) = F(v), \quad \forall v \in \mathbb{V}, \tag{19}$$

where $\mathbb{V}$ is a suitable function space, $a : \mathbb{V} \times \mathbb{V} \to \mathbb{R}$ is a bilinear form induced by the considered differential operator $\mathfrak{L}$, and $F : \mathbb{V} \to \mathbb{R}$ is a linear form depending on f.

Using the Galerkin method we approximate the solution of (18) based on the weak form (19). We select a finite-dimensional approximation space on $\Omega$,

$$\mathbb{W} \coloneqq \langle \varphi_1, \varphi_2, \dots, \varphi_{n_\mathbb{W}} \rangle \subset \mathbb{V}, \quad \dim(\mathbb{W}) = n_\mathbb{W}, \tag{20}$$

and we look for

$$u_\mathbb{W} \in \mathbb{W} \quad \text{such that} \quad a(u_\mathbb{W}, w) = F(w), \quad \forall w \in \mathbb{W}.$$

Taking

$$u_\mathbb{W} = \sum_{i=1}^{n_\mathbb{W}} c_i \varphi_i$$

leads to a linear system $A\mathbf{c} = \mathbf{F}$, where the matrix $A$ and the vector $\mathbf{F}$ are defined as

$$A_{i,j} \coloneqq a(\varphi_j, \varphi_i), \quad i, j = 1, \dots, n_\mathbb{W}, \qquad \mathrm{F}_i \coloneqq F(\varphi_i), \quad i = 1, \dots, n_\mathbb{W}. \tag{21}$$

Selecting different subspaces $\mathbb{W}$ results in different Galerkin methods.

Complying with the isogeometric approach, the fields of interest are described by means of spline basis functions. The basis functions in (20) are traditionally selected as tensor-product B-splines or NURBS. The characteristics outlined in Proposition 4 indicate that TB-splines are plug-to-plug compatible with classical (polynomial) B-splines and turn out to be a good choice as well [31]. Furthermore, the set of LR TB-splines, as presented in Definition 13, is a suitable choice for incorporating adaptive refinement.

Let us select a space of bivariate LR Tchebycheffian splines as the approximation space in (20),

$$\mathbb{W} = \mathbb{S}_{\boldsymbol{p}}(\mathcal{M}) = \left\langle N_{\boldsymbol{\Xi}_k, \boldsymbol{p}} \in \mathcal{S}_{\boldsymbol{p}}(\mathcal{M}) \right\rangle. \tag{22}$$

This space is spanned by the set of TB-splines $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M})$ related to the local space $\mathbb{P}_{p_1}^{\mathcal{W}_1} \otimes \mathbb{P}_{p_2}^{\mathcal{W}_2}$ of bi-degree $\boldsymbol{p} = (p_1, p_2)$ with maximal smoothness and roots $\mathcal{W}_1$ and $\mathcal{W}_2$ corresponding to the ECT-spaces in each parametric direction; see Definition 13. These TB-splines are constructed on an LR-mesh $\mathcal{M}$ defined on the domain $\Omega = [0, 1]^2$.

The univariate building blocks for our basis functions are univariate TB-splines identified by ECT-spaces of the form (5). They offer a diverse array of algebraic polynomial, exponential, and trigonometric function combinations, equipped with a wide spectrum of shape parameters. Properly selecting the space's structure, incorporating various function types, along with appropriate shape parameters, is of vital importance to effectively harness the capabilities of the ECT-spaces. An in-depth study for the selection of ECT-spaces has been presented in [31, Section 3.2] based on an (automatic) problem-driven strategy.

### 4.2. Adaptive strategy

Ever since the advent of polynomial LR B-splines, a prominent issue under discussion has been the quest for an optimal refinement strategy that aligns with objectives such as achieving linear independence, ensuring mesh quality, and accommodating grading and shape regularities. The same issue arises in the LR TB-spline setting. Thanks to the structural similarities between polynomial B-splines and TB-splines, the refinement strategies developed for polynomial LR B-splines are also applicable for LR TB-splines.

#### 4.2.1. Local refinement strategy

We adopt the local refinement strategy introduced for polynomial LR B-splines in [18, Section 4.2] and known as structured mesh refinement. As opposed to the classical finite element method, in structured mesh refinement we select the LR TB-splines contributing most to the approximation error rather than the box-partition elements contributing most to the error and refine those LR TB-splines. This approach proves to be a more sensible choice since in LR-mesh refinement any newly inserted split must traverse the support of at least one LR TB-spline.

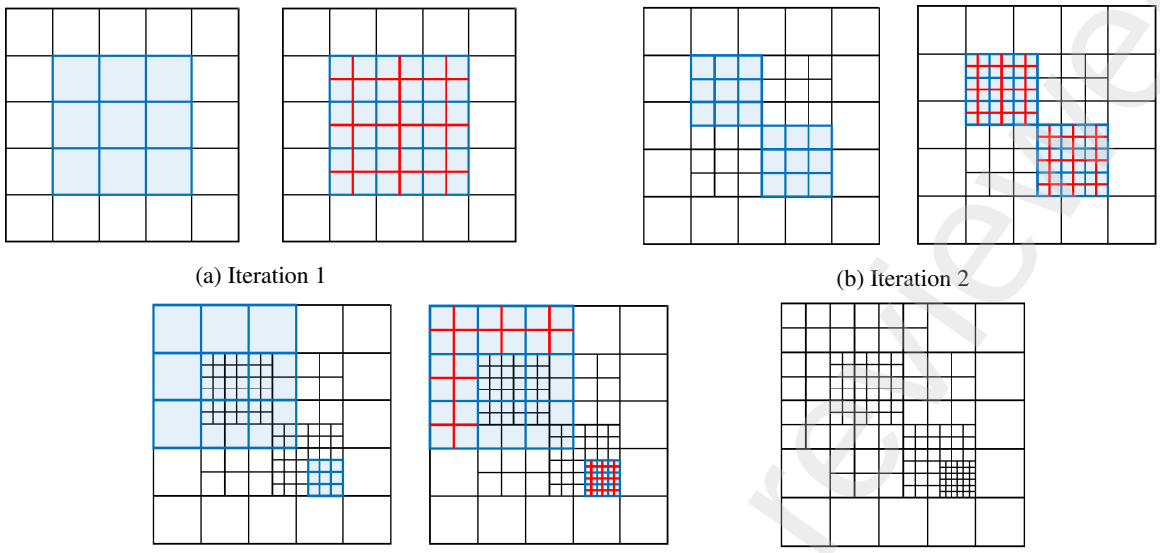
The structured mesh refinement is a dyadic refinement where all selected LR TB-splines are refined by halving all of the intervals in their local knot vectors, resulting in the insertion of a net of splits (located in the TB-spline supports) in accordance with Definition 13 (see also Algorithm 1 in Appendix A). An LR-mesh obtained with the structured mesh refinement strategy is called a structured LR-mesh.

Figure 11 illustrates an example of (three iterations of) structured mesh refinement for LR TB-splines of bi-degree $\boldsymbol{p} = (2, 2)$. Specifically, Figure 11 (c) exhibits how the maximal smoothness is maintained in the LR-meshes across the meshlines. If the split already partially exists in the mesh then instead of increasing the multiplicity of already existing meshlines, we only elongate it to traverse the complete support of the marked TB-spline. And if the split already exists completely then instead of increasing the multiplicity of the corresponding meshlines, we neglect that split insertion, staying true to the configuration of LR-meshes with fixed multiplicity of 1 for all internal meshlines (as assumed in Section 3.1).

**Remark 17.** *One of the shortcomings of the structured mesh strategy is that it might produce a linearly dependent set of LR (T)B-splines. Many efforts have been made to achieve linear independence of LR B-splines. One of the suggested solutions was achieving local linear independence on each element in the mesh by enforcing the non-nested support ($N_2S$) property; see [7, 30]. These LR B-splines possess nice properties, such as*

- *the number of non-zero LR B-splines over an element $\sigma$ is $(p_1 + 1)(p_2 + 1)$,*
- *they span the full local spline space,*
- *they form a partition of unity, without the use of scaling weights.*

*Besides the function-based refinement strategies, there are also some box-based strategies, similar to the classical finite element approach, where the elements contributing the most to the error are selected for refinement. These strategies are hierarchical locally refined (HLR) meshing [7] and effective grading (EG) refinement [28], which produce LR-meshes with good grading while ensuring local linear independence. All of the aforementioned refinement strategies are developed for polynomial B-splines on LR-meshes, but are also applicable for LR TB-splines. It should be noted, however, that prioritizing the local linear independence of the LR (T)B-splines imposes many constraints on the refinement strategy. On the other hand, with the structured mesh refinement the generated meshes, at least in the region away from the boundary, are locally tensor meshes. Hence, the LR (T)B-splines defined in these zones of the mesh behave like the standard (T)B-splines and are locally linearly independent.*



(a) Iteration 1 (b) Iteration 2

(c) Iteration 3 (d) Final mesh

Figure 11: Example of structured mesh refinement for LR TB-splines of bi-degree $\boldsymbol{p} = (2, 2)$. Figure (a), left: Initial tensor mesh $\mathcal{M}_0$, with highlighted the support of the LR TB-spline to be refined. Figure (a), right: Set of splits to be inserted to accomplish structured mesh refinement, halving each interval in the support of the marked TB-spline. Figure (b): Refined LR-mesh $\mathcal{M}_1$ after the first iteration, with highlighted the supports of the two LR TB-splines to be refined in Iteration 2 and the set of splits to be inserted in this iteration. Figure (c), left: Refined LR-mesh $\mathcal{M}_2$ after the second iteration, with highlighted the supports of the two LR TB-splines to be refined in Iteration 3. Here the marked TB-splines have a different knot resolution. Figure (c), right: Illustration of how to maintain the multiplicity of the meshlines to one in the mesh. If the split to be inserted already partially exists in the mesh then it is only extended for the remaining part of the support of the TB-spline to be refined. Figure (d): Final LR-mesh $\mathcal{M}_3$ obtained.

#### *4.2.2. Marking strategy*

After discussing the Solve and Refine steps in the adaptive cycle, we now focus on the remaining Estimate and Mark steps. As explained in Section 4.2.1, the structured mesh refinement strategy is based on refining LR TB-splines, hence we need to mark the TB-splines contributing maximally to the approximation error. This is estimated by two different strategies listed below.

- Problem-oriented ad hoc refinement: This method involves leveraging a priori knowledge about the problem's behavior. By understanding the problem's geometric feature and identifying specific regions in the domain that require refinement, we can selectively mark functions with support in those areas.

- Error-based automatic refinement: This method uses the residual-based error indicator in the $L^2$-norm. Assuming we have computed the discrete solution $u_{\mathbb{W}}$ of the model problem (18), the error indicator $\eta_\sigma$ on an element $\sigma$ within an LR-mesh $\mathcal{M}$ is given by

$$\eta_\sigma := \|\mathrm{f} - \mathfrak{Q} u_{\mathbb{W}}\|_{L^2(\sigma)}. \tag{23}$$

This quantity is defined element-wise, while we need to mark basis functions for the structured mesh refinement strategy. Therefore, we first mark elements with high error and then mark all the TB-splines that have support over those elements. Our element selection is based on a proper threshold

criterion,

$$\eta_\sigma \geq \psi \cdot \max_{\hat{\sigma}} \eta_{\hat{\sigma}}, \quad \psi \geq 0. \tag{24}$$

The marking parameter $\psi$ facilitates refinement in the spectrum from global refinement to no refinement at all by selecting $\psi = 0$ and $\psi > 1$, respectively. Note that the proportion of elements marked can vary with each step since the computation of the threshold depends only on the magnitude of the maximum of the error, without considering the distribution of estimated errors.

**Remark 18.** *As an alternative to marking elements with errors exceeding a certain threshold as in* (24)*, we can also label a specific percentile of elements based on their error distribution. In this paper, we just use the criterion defined in* (24) *and maintain the same value for the marking parameter $\psi$ across all iterations. This choice is made because the nature of all the case studies discussed in Section 5 involves small regions with substantial error magnitudes. Consequently, this strategy effectively restricts the refinement region to a minimum extent.*

## 5. Numerical results

In this section, we present some case studies using LR TB-splines in isogeometric Galerkin discretizations of second-order problems. In all case studies, the approximation spaces are taken as bivariate Tchebycheffian spline spaces of the form (22) over LR-meshes, related to the local space $\mathbb{P}_{p_1}^{\mathcal{W}_1} \otimes \mathbb{P}_{p_2}^{\mathcal{W}_2}$ of bi-degree $\boldsymbol{p} = (p_1, p_2)$ with roots $\mathcal{W}_1$ and $\mathcal{W}_2$ corresponding to each parametric direction. The LR-mesh refinement is carried out with the structured mesh refinement strategy described in Section 4.2.1. The initial mesh $\mathcal{M}_0$ is a tensor mesh that partitions the domain with $m + 1$ equidistant breakpoints, i.e., the distance between two consecutive breakpoints is $h_0 = 1/m$. Given that the structured mesh refinement follows a dyadic pattern, the minimum distance between two consecutive meshlines along one parametric direction for each iteration of refinement advances as

$$h_\ell = h_{\ell-1}/2, \quad \ell = 1, 2, \ldots.$$

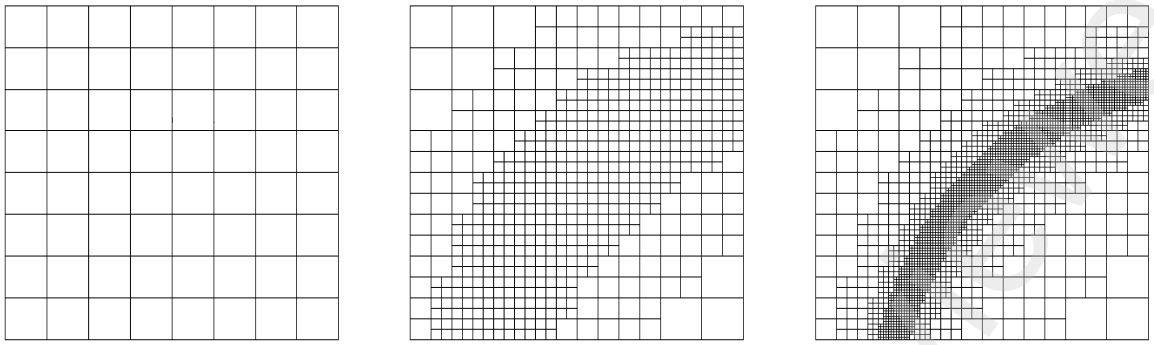
We focus on the general second-order problem of the form

$$\begin{cases} -\nabla \cdot (\kappa \nabla u) + \mathbf{a} \cdot \nabla u + c\, u = \mathrm{f}, & \text{in } \Omega = (0,1)^2, \\ u = \mathrm{g}, & \text{on } \partial\Omega, \end{cases} \tag{25}$$

where $\mathbf{a}$ is the advection flow velocity, $\kappa$ the diffusivity, $c$ the reaction coefficient, f the prescribed source function, and g the Dirichlet boundary function.

As usual, homogeneous boundary conditions are satisfied pointwise exactly. In the non-homogeneous case, the boundary function is approximated in the underlying LR Tchebycheffian spline space by a suitable approximation strategy (e.g., least-squares approximation or quasi-interpolation) and subsequently the reduction to the homogeneous case is considered, so dealing again with a special instance of the problem (18). It is important to remark that, when discontinuous Dirichlet boundary conditions are involved, a proper boundary treatment is imperative. In such a case, in order to avoid oscillations along the boundary, we approximate the boundary function by means of the shape-preserving Schoenberg operator (see [21, 33]), i.e, we consider the linear combination of the boundary TB-splines whose coefficients are obtained by evaluating the boundary function at the corresponding Greville abscissae (assuming that linear polynomials belong to the ECT-space of interest).

For the (numerical) computation of the integrals required in the construction of the linear system in (21), we employ element-wise Gaussian quadrature rules. The choice of quadrature points for discretizations



(a) $\mathcal{M}_0$ (dof = 100) (b) $\mathcal{M}_2$ (dof = 604) (c) $\mathcal{M}_4$ (dof = 2538)

Figure 12: Case study 1. Some LR-meshes for the solution of the Poisson problem in (26), obtained with an ad hoc refinement strategy. The corresponding number of degrees of freedom (dof) for bi-degree $\boldsymbol{p} = (2, 2)$ are mentioned.

involving TB-splines is a non-trivial task, often necessitating quadrature rules of higher order compared to those used for classical polynomial splines. In all case studies presented we use $5\boldsymbol{p}$ quadrature points in each element for LR TB-splines consisting of exponential functions with large shape parameters.

### *5.1. Case study 1: Poisson problem*

In this case study we address a Poisson problem commonly used in the literature as a benchmark for adaptive refinement schemes; see [18, 30]. We consider (25) with $\kappa = 1$, $c = 0$, $\mathbf{a} = \mathbf{0}$, and f obtained from the exact solution

$$u(x_1, x_2) = \arctan\Big(100\Big(\sqrt{(x_1 - 1.25)^2 + (x_2 + 0.25)^2} - \frac{\pi}{3}\Big)\Big). \tag{26}$$

While the problem may seem mathematically smooth since we are just solving for the Laplacian of the analytical solution, the highly varying right-hand side introduces complexity in the solution. The solution has rapid changes across the arc of the circumference

$$(x_1 - 1.25)^2 + (x_2 + 0.25)^2 = (\pi/3)^2, \tag{27}$$

traversing the domain $\Omega = [0, 1]^2$ and resulting into an internal sharp layer.

In the literature, bi-quadratic polynomial splines are prominently used for the computation of approximate solutions of this problem. Here, we select Tchebycheffian splines of the same degree to prove the efficacy of Tchebycheffian splines. Since the problem exhibits an internal sharp layer, a Tchebycheffian spline space with exponential functions (with suitable parameters) is a good choice. Following (5), we take the local space as

$$\mathbb{P}_2^{(\alpha,-\alpha)} \otimes \mathbb{P}_2^{(\alpha,-\alpha)} = \left\langle 1, e^{\alpha x_1}, e^{-\alpha x_1} \right\rangle \otimes \left\langle 1, e^{\alpha x_2}, e^{-\alpha x_2} \right\rangle, \quad \text{with } \alpha = 10.$$

We use an ad hoc refinement strategy as outlined in Section 4.2.2, where we utilize a priori knowledge about the problem to identify the specific regions in the domain that require refinement. More precisely, we mark the TB-splines that have support touching/overlapping the arc (27) and refine them according to the structured mesh refinement strategy. The plots of some LR-meshes are illustrated in Figure 12 and an approximate solution is visualized in Figure 13.

Figure 14 illustrates the convergence of the error in $L^\infty$-norm, computed by sampling the approximate and exact solutions on a uniform grid consisting of 1001 points along each direction in the domain. We

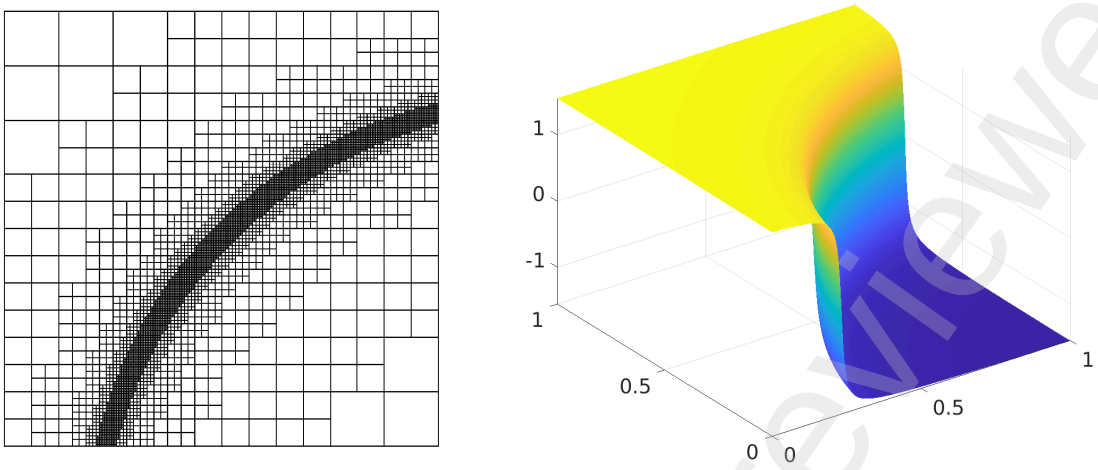



Figure 13: Case study 1. Plot of the approximate solution (right) for the Poisson problem with exact solution (26), obtained by LR TB-splines related to the local space $\mathbb{P}_2^{(10,-10)} \otimes \mathbb{P}_2^{(10,-10)}$ on the mesh $\mathcal{M}_5$ with dof = 5101 (left).

| $h_\ell$ | dof | $\mathbb{P}_2^{(10,-10)} \otimes \mathbb{P}_2^{(10,-10)}$ | $\mathbb{P}_2 \otimes \mathbb{P}_2$ |
|---|---|---|---|
| 1/8 | 100 | 1.5873929998 | 7.9295697929 |
| 1/16 | 277 | 0.6163964505 | 1.7534324399 |
| 1/32 | 604 | 0.2982445346 | 0.3859733442 |
| 1/64 | 1249 | 0.0937572015 | 0.0961967520 |
| 1/128 | 2538 | 0.0139797634 | 0.0139829152 |
| 1/256 | 5101 | 0.0012123306 | 0.0012128007 |

Table 3: Case study 1. Comparison of the $L^\infty$ error for the Poisson problem with exact solution in (26) when considering TB-splines and B-splines of bi-degree $\boldsymbol{p} = (2, 2)$ on a sequence of LR-meshes, obtained with an ad hoc refinement strategy, some of them are illustrated in Figures 12 and 13.

assess the convergence rates of Tchebycheffian splines in comparison to classical polynomial splines on LR-meshes, starting from a coarse tensor mesh with $h_0 = 1/8$ to $h_5 = 1/256$ after five iterations of local refinement. We also compare the results of their tensor mesh counterpart with the same resolution. The plot clearly shows that the approximate solution obtained by TB-splines on LR-meshes converges faster than their tensor mesh equivalent of the same resolution. Table 3 shows the $L^\infty$-error for all LR-mesh iterations and it highlights smaller error for TB-splines compared to B-splines for spaces of equivalent dimension, especially on the coarser meshes.

### *5.2. Case study 2: Reaction-diffusion problem*

Here we solve a reaction-diffusion problem (25) on a square, with

$$\kappa = 10^{-3}, \quad c = 1, \quad \mathbf{a} = \mathbf{0}, \quad \mathrm{f} = 0; \tag{28}$$

see [1, 25]. The boundary condition for this problem is set to zero except near the corners as illustrated in Figure 15 (left). The treatment of this discontinuous Dirichlet boundary condition is carried out with a Schoenberg quasi-interpolant.

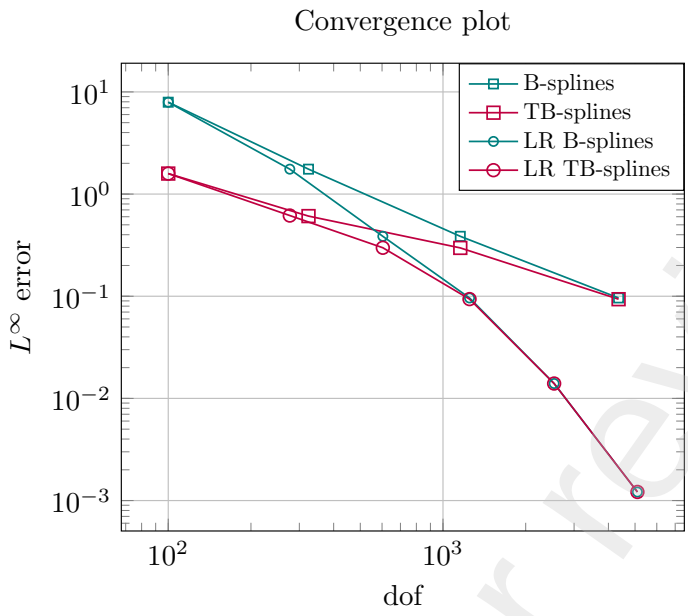



Figure 14: Case study 1. Convergence plot of the $L^\infty$ error for the Poisson problem with exact solution in (26) when considering TB-splines and B-splines of bi-degree $\boldsymbol{p} = (2, 2)$ on tensor meshes and LR-meshes.

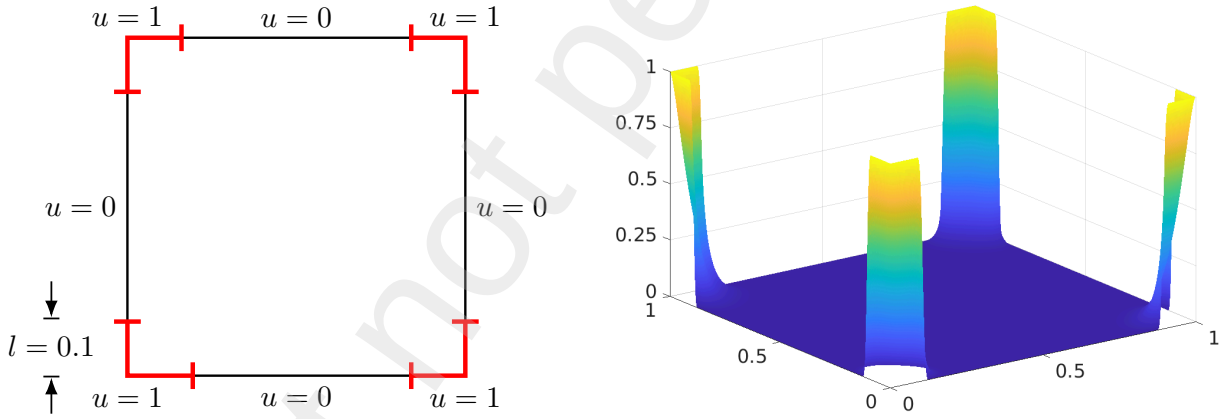



Figure 15: Case study 2. Left: The domain $\Omega = [0, 1]^2$ with the Dirichlet boundary conditions, where all corners are set to $u = 1$ up to the length $l = 0.1$ (red) and the rest of the boundary is set to $u = 0$. Right: Plot of an approximate solution for the reaction-diffusion problem specified in (28), which is computed on a very fine tensor mesh with B-splines of bi-degree $\boldsymbol{p} = (3, 3)$ with $h = 1/1000$ and dof = 1006009. We use this solution as a reference to compare the other results in this case study.

The dominance of reaction over diffusion is defined by the Damköhler number, defined as the ratio of the reaction coefficient ($c$) to the diffusion coefficient ($\kappa$). In this case study, the system is strongly reaction dominated, as the Damköhler number is $10^3$. Therefore, we expect the solution to be zero throughout the domain, except near the corners, where enforcing the boundary condition rapidly spikes the solution to one. The approximate solution belonging to a polynomial spline space tends to exhibit spurious oscillations

around the jump, unless the discretization is fine enough. Using locally refined structures can help with suppressing the oscillations and resolve the spikes near the corners. In addition, the Tchebycheffian spline's ability in capturing sharp layers can further reduce such spurious oscillations, without refining too much. Therefore, we select again a Tchebycheffian spline space with exponential functions, where the local space is given by

$$\mathbb{P}_3^{(\alpha,-\alpha)} \otimes \mathbb{P}_3^{(\alpha,-\alpha)} = \left\langle 1, x_1, e^{\alpha x_1}, e^{-\alpha x_1} \right\rangle \otimes \left\langle 1, x_2, e^{\alpha x_2}, e^{-\alpha x_2} \right\rangle, \quad \text{with } \alpha = 35.$$

We use a residual-based error indicator to guide the local refinement. Given that $\mathrm{f} = 0$, the element-wise error estimator in (23) becomes

$$\eta_\sigma = \|\nabla \cdot (\kappa \nabla u_{\mathbb{W}}) - c\, u_{\mathbb{W}}\|_{L^2(\sigma)}, \tag{29}$$

and we set the marking parameter to $\psi = 0.95$ in the marking criterion (24).

In the absence of an explicit expression for the exact solution of this problem, we check the accuracy of the approximate solutions by comparing them with a high-fidelity B-spline solution. This B-spline solution, depicted in Figure 15 (right), is computed on a tensor mesh with an element size of $h = 1/1000$, with B-splines of bi-degree $\boldsymbol{p} = (3, 3)$ and dof = 1006009. The contour plots in Figure 16 illustrate the error in solutions obtained by LR TB-splines and LR B-splines, both with respect to the reference B-spline solution, along with the LR-meshes on which the system is solved. The approximate solutions are evaluated on a uniform grid of $2001 \times 2001$ points. Analyzing the contour plots, it shows that LR TB-spline solutions exhibit smaller oscillation areas, and the magnitude of these oscillations is also reduced, especially on the coarser meshes. This comparison confirms the capability of TB-splines in capturing sharp layers while eliminating unwanted spurious oscillations.

### *5.3. Case study 3: Advection-diffusion problem*

In this case study we consider a standard benchmark problem with advection flow skew to any parametric direction [15, 24, 39, 31]. We solve an advection-diffusion problem (25) on a square, with

$$\kappa = 1, \quad c = 0, \quad \mathbf{a} = \mathfrak{a}\,(\cos(\theta), \sin(\theta))^T, \quad \theta = \frac{\pi}{4}, \quad \mathfrak{a} = 10^4, \quad \mathrm{f} = 0, \tag{30}$$

and the discontinuous Dirichlet boundary conditions as shown in Figure 17. The jump in the boundary conditions at the point $(0, 0.2)$ creates an inner sharp layer aligned with the advection flow direction identified by $(\cos(\theta), \sin(\theta))$. Additionally, the solution also exhibits a sharp boundary layer.

In advection-diffusion problems, the dominance of advection over diffusion is defined by the global Péclet number

$$\mathbf{Pe}_g := \frac{\|\mathbf{a}\|}{\kappa}.$$

This problem is advection-dominated since the Péclet number is $\mathbf{Pe}_g = \mathfrak{a} = 10^4$. Such problems tend to exhibit spurious oscillations in their approximate solutions when polynomial splines are used, until the discretization is fine enough to resolve the sharp layers featured by the exact solution. A common practice to overcome this issue is to use stabilization methods such as the SUPG and GLS method; see [9, 16]. While these stabilization methods effectively eliminate spurious oscillations, it is also important to note that they tend to "smooth out" the layers featured by the exact solution. Moreover, their effectiveness is hugely dependent on the choice of some parameters appearing in the various stabilization methods. It has been showcased in [31] that Tchebycheffian splines can offer a flexible alternative for such advection-dominated problems without the need for stabilization and better localization of boundary and internal layers.

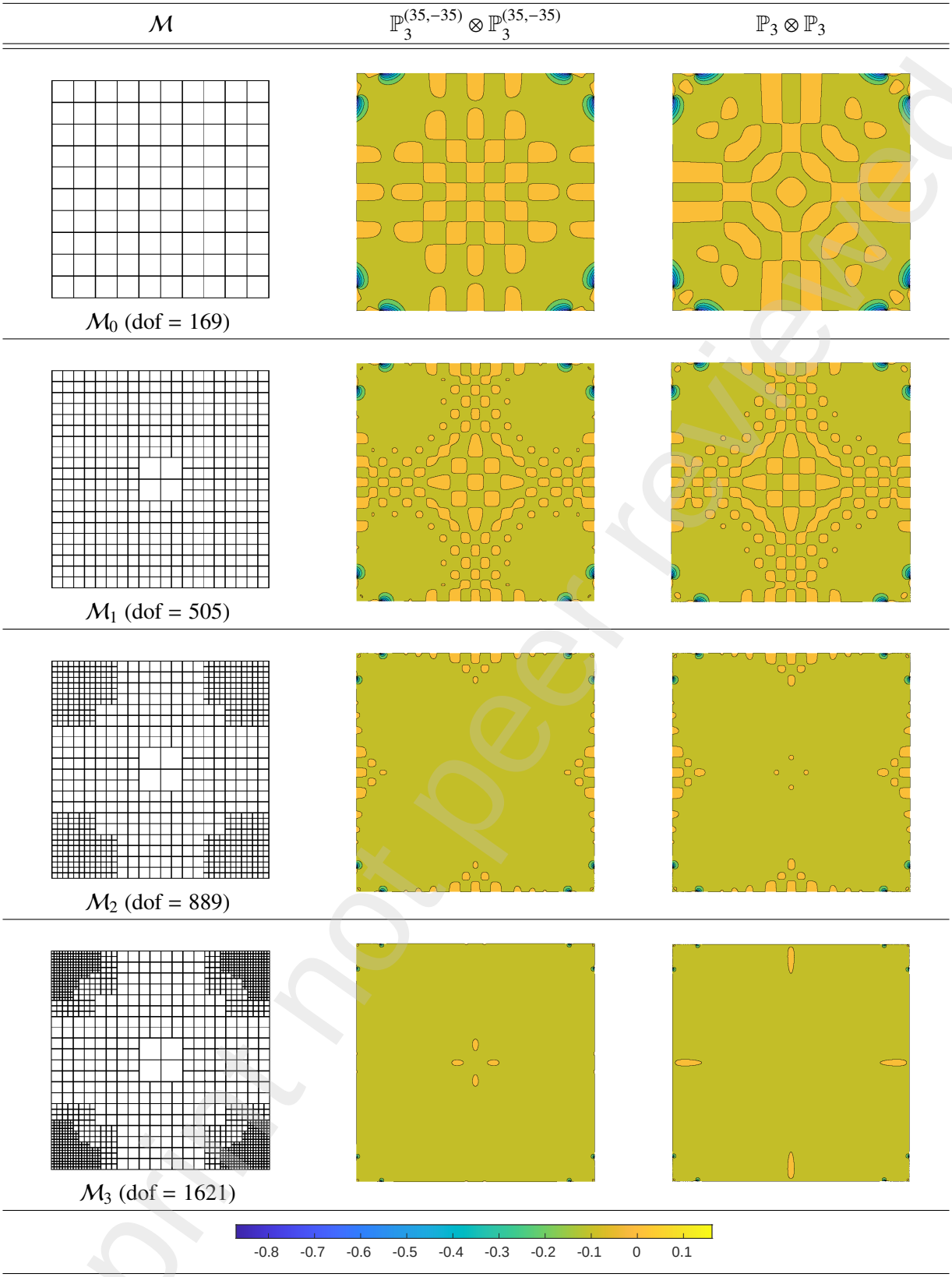



Figure 16: Case study 2. Some LR-meshes for the solution of the reaction-diffusion problem specified in (28), where the refinement is guided by the residual-based error indicator in (29) with marking parameter $\psi = 0.95$, along with the contour plots of the error in solutions obtained by TB-splines and B-splines of bi-degree $\boldsymbol{p} = (3, 3)$, both with respect to the reference solution shown in Figure 15.

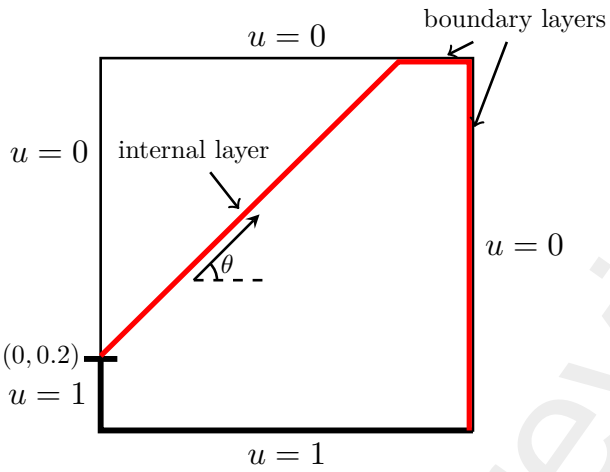



Figure 17: Case study 3. The domain $\Omega = [0, 1]^2$ with the Dirichlet boundary conditions, the sharp internal layer (red) generated along the advection flow at angle $\theta$ and the boundary layers (red).

To capture the sharp layers of the problem specified in (30), an adequate choice of Tchebycheffian spline space contains exponential functions with suitable parameters [31]. These parameters are dependent on the components of the advection flow velocity in the two parametric directions, given as

$$\mathbb{P}_4^{(\mathfrak{a}\cos(\theta))} \otimes \mathbb{P}_4^{(\mathfrak{a}\sin(\theta))} = \left\langle 1, x_1, (x_1)^2, (x_1)^3, e^{\mathfrak{a}\cos(\theta)x_1} \right\rangle \otimes \left\langle 1, x_2, (x_2)^2, (x_2)^3, e^{\mathfrak{a}\sin(\theta)x_2} \right\rangle. \tag{31}$$

For the adaptive refinement we use the ad hoc refinement strategy described in Section 4.2.2. Since we expect the internal and boundary layers to be sharp, we can refine along these layers. Figure 18 illustrates the LR-mesh $\mathcal{M}_2$ for $\boldsymbol{p} = (4, 4)$ obtained with this ad hoc strategy after two iterations, along with the approximate solutions using the LR TB-spline space related to the local space (31) and the LR B-spline space on the mesh $\mathcal{M}_2$. Since this mesh is not fine enough, the solution with plain LR B-splines would exhibit huge spurious oscillations. Hence, it is necessary to use stabilization in the LR B-spline solution to have a comparable solution against the unstabilized LR TB-spline solution. The discontinuous Dirichlet boundary condition in all the cases is imposed by the Schoenberg quasi-interpolant.

For comparison, we also solve the same problem with B-splines on an LR-mesh after many refinement steps to obtain a fine enough mesh such that the stabilization of the solution is not required anymore. The resulting mesh $\mathcal{M}_6$ is presented in Figure 19, where the mesh is obtained using the same ad hoc refinement strategy as described before for $\boldsymbol{p} = (4, 4)$.

Table 4 summarizes the values of maximal over- and undershoot in the neighborhood of the layers evaluated on a uniform grid of $3001 \times 3001$ points along each direction for the different setups of the meshes and spline spaces, together with the resolution of the mesh $h_\ell$ and the number of degrees of freedom (dof).

The approximate solution with LR TB-splines on an LR-mesh gives a similar result compared to the TB-spline solution on a tensor mesh with the same $h_2 = 1/128$, showing that the use of an LR-mesh for TB-splines results in the same level of accuracy with fewer degrees of freedom (about 2.5 times fewer in this case). The stabilized LR B-spline solution on mesh $\mathcal{M}_2$ does eliminate the spurious oscillations but the corresponding solution is too smooth, resulting in a poor localization of the internal and boundary layers. On the other hand, the unstabilized LR B-spline solution on a very fine mesh $\mathcal{M}_6$ localizes the layers accurately but at a very high cost. In conclusion, TB-splines on an LR-mesh are able to eliminate spurious

| TB-spline space | Local refinement | SUPG stabilization | $h_\ell$ | dof | max | min |
|---|---|---|---|---|---|---|
| $\mathbb{P}_4^{(\mathfrak{a}\cos(\theta))}\otimes\mathbb{P}_4^{(\mathfrak{a}\sin(\theta))}$ | Yes | No | 1/128 | 6957 | 1.0058 | $-6.5089\times10^{-3}$ |
| $\mathbb{P}_4^{(\mathfrak{a}\cos(\theta))}\otimes\mathbb{P}_4^{(\mathfrak{a}\sin(\theta))}$ | No | No | 1/128 | 17424 | 1.0059 | $-6.5036\times10^{-3}$ |
| $\mathbb{P}_4\otimes\mathbb{P}_4$ | Yes | Yes | 1/128 | 6957 | 1.0019 | $-1.6772\times10^{-3}$ |
| $\mathbb{P}_4\otimes\mathbb{P}_4$ | Yes | No | 1/2048 | 121283 | 1.0071 | $-1.3056\times10^{-3}$ |

Table 4: Case study 3. Comparison of maximum and minimum of the TB-spline solution (without stabilization) against the B-spline solution (with SUPG stabilization) on the same LR-mesh $\mathcal{M}_2$ as in Figure 12, the B-spline solution (without stabilization) on the very fine LR-mesh $\mathcal{M}_6$ as in Figure 19, and the TB-spline solution on a tensor mesh of the same resolution as in $\mathcal{M}_2$ for $\boldsymbol{p} = (4, 4)$ and $\mathfrak{a} = 10^4$.

oscillations without the need for stabilization in the approximate solution, while still capturing the sharp layers precisely.

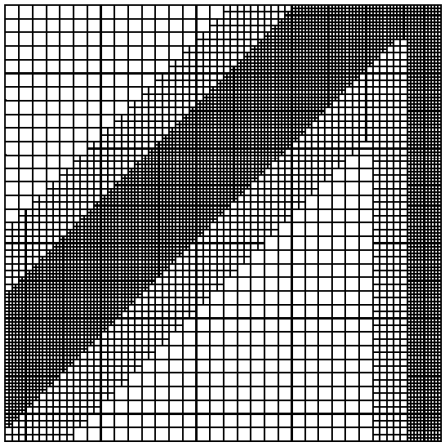
## 6. Conclusion

Polynomial splines are smooth piecewise functions with pieces belonging to algebraic polynomial spaces. Replacing the algebraic polynomial spaces with E(C)T-spaces gives rise to the rich universe of Tchebycheffian splines. Most of the results known for polynomial splines extend in a natural way to the Tchebycheffian setting. In particular, under suitable assumptions, Tchebycheffian splines can be represented in terms of TB-splines, basis functions with similar properties to the classical polynomial B-splines. Multivariate versions of TB-splines can be easily obtained by taking tensor products and popular local refinement technologies, based on local tensor products, can also be applied in the Tchebycheffian setting. For example, the knot insertion procedure allows for defining LR TB-splines as a natural generalization of LR B-splines.

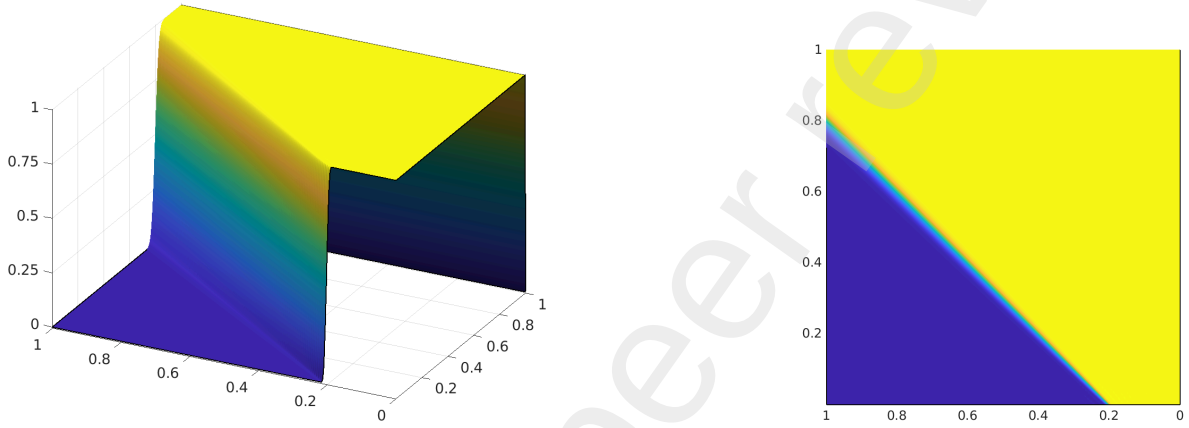
In this paper we have been focusing on Tchebycheffian splines whose pieces belong to null-spaces of constant-coefficient linear differential operators. Although this is a more restricted class, it offers the freedom of combining algebraic polynomial, exponential, and trigonometric functions, thus providing a large family of approximation spaces for isogeometric methods that are flexible and effective both from the analytical and geometrical point of view. They can be identified according to problem-oriented selection strategies [31]. Moreover, the corresponding (LR) TB-splines can be easily incorporated in any software library supporting polynomial (LR) B-splines to enrich its capability, thanks to efficient evaluation and manipulation routines recently developed for this class [38].

LR TB-splines are a viable spline technology on unstructured meshes and offer a valid alternative to classical LR B-splines in adaptive isogeometric analysis. We have shown that the adaptive strategy combined with problem-oriented approximation spaces may create a synergistic effect and may produce results of similar quality with less levels of refinement, and so fewer degrees of freedom, without the need for possible stabilization, compared to the polynomial setting.

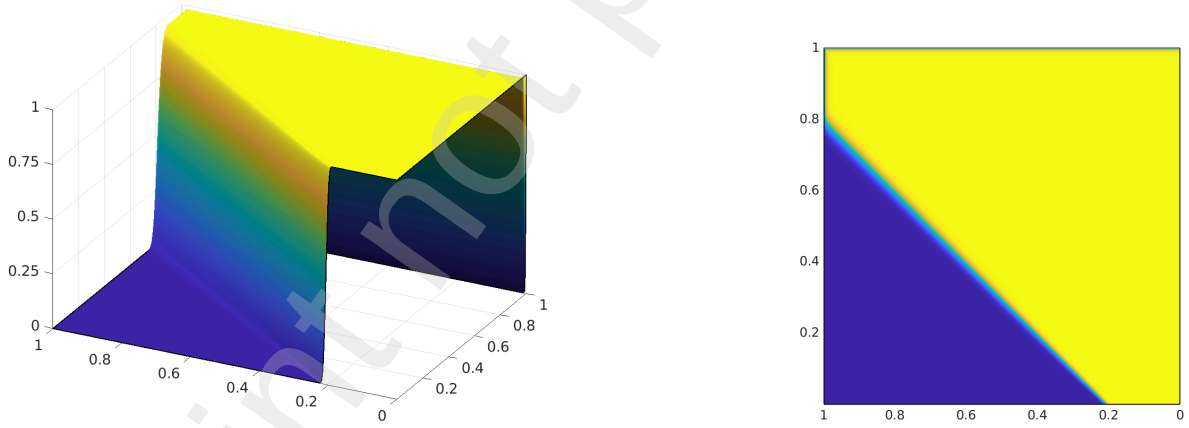
The complete structural similarity with the polynomial case also implies that LR TB-splines suffer from the same weaknesses as their polynomial counterpart. For example, LR TB-splines might be (locally) linearly dependent; see also Remark 17. Common refinement strategies proposed in the literature, such as the minimum span, full span, and structured refinement [18, 37], do not ensure linear independence of the



(a) LR-mesh $\mathcal{M}_2$ for $\boldsymbol{p} = (4, 4)$ with $h_2 = 1/128$ and dof = 6957 obtained with ad-hoc refinement strategy.




(b) TB-spline space: $\mathbb{P}_4^{(a\cos(\theta))} \otimes \mathbb{P}_4^{(a\sin(\theta))}$.




(c) B-spline space: $\mathbb{P}_4 \otimes \mathbb{P}_4$ with SUPG stabilization.

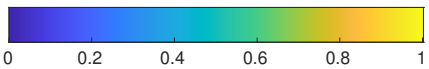



Figure 18: Case study 3. Plot of the LR-mesh $\mathcal{M}_2$ and the corresponding approximate solutions using different TB-spline spaces of $\boldsymbol{p} = (4, 4)$ for the advection-diffusion problem specified in (30) with advection skew at $\theta = 45°$ and global Péclet number $\mathbf{Pe}_g = 10^4$.

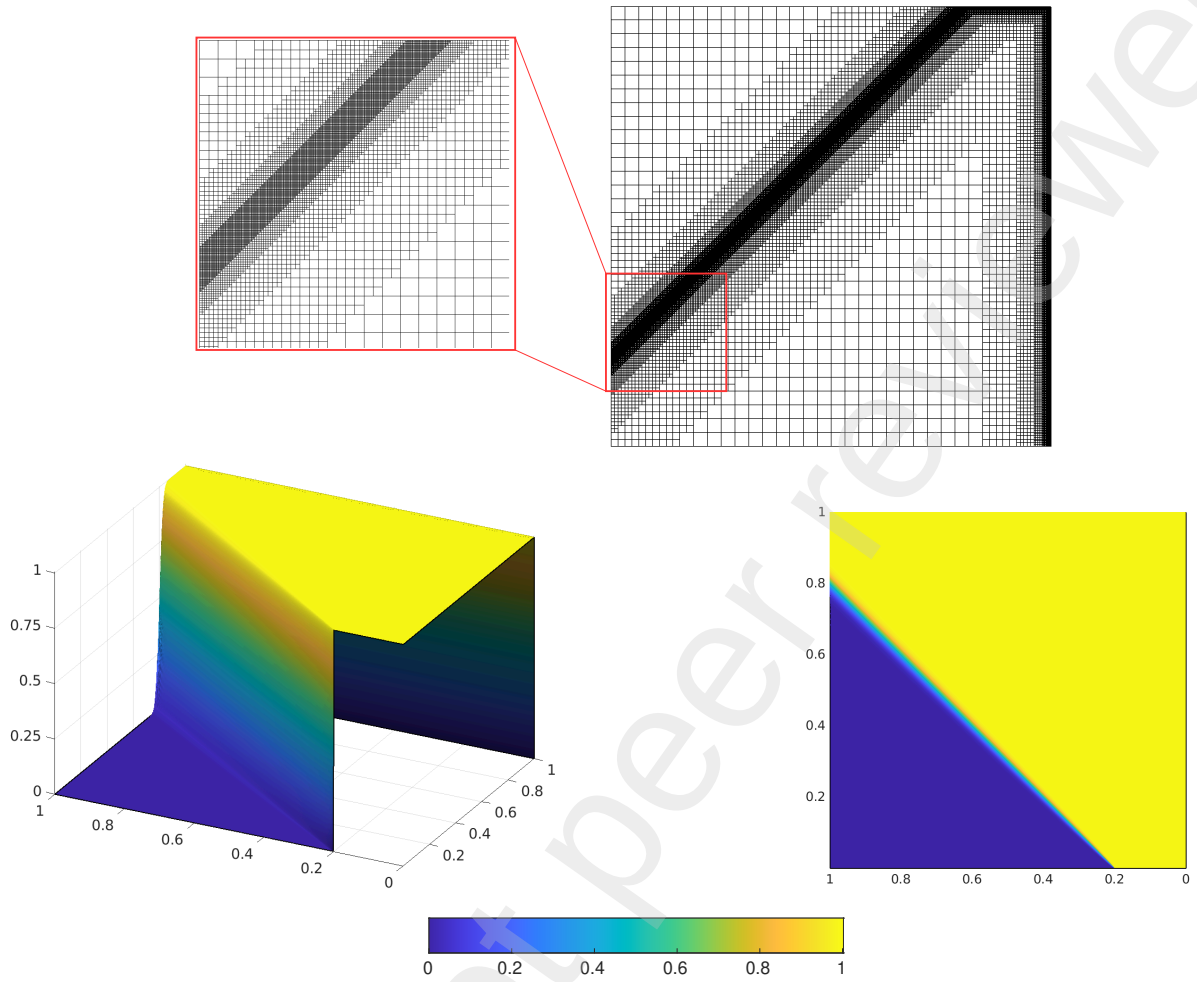



Figure 19: Case study 3. Plot of the LR-mesh $\mathcal{M}_6$ of $\boldsymbol{p} = (4, 4)$ with $h_6 = 1/2048$ and dof = 121283, obtained with ad hoc refinement, and the corresponding approximate solution using the B-spline space (i.e., the local space is $\mathbb{P}_4 \otimes \mathbb{P}_4$) without SUPG stabilization for the advection-diffusion problem specified in (30) with advection skew at $\theta = 45°$ and global Péclet number $\mathbf{Pe}_g = 10^4$.

LR B-splines. The so-called peeling algorithm can be adopted to remove redundant basis functions [12, 29] but refinement strategies ensuring linear independence are strongly preferred, both from the theoretical and algorithmic point of view. A complete characterization of linear independence for LR B-splines is still not known, but there exist features of the underlying LR mesh, like the so-called non-nested support ($N_2S$) property [7], that guarantee the stronger property of local linear independence. Refinement strategies ensuring the $N_2S$ property for LR B-splines have been proposed in [8, 28, 30]. It is likely the case that the structural similarity paves again the path for an extension of the above properties from the polynomial to the Tchebycheffian setting. This can be an interesting topic for future investigation. In particular, the identification of refinement strategies ensuring (local) linear independence for LR (T)B-splines, not relying on peculiar properties of algebraic polynomials but only on structural properties of the involved spaces, may provide further insights and understandings of the LR paradigm, also in the polynomial case.

**Acknowledgements**

This work was supported by the European Union's Horizon 2020 research and innovation program through the MSCA-ITN-2019 project GRAPES (contract n. 860843 – CUP E54I19002020006). We also acknowledge the MUR Excellence Department Project MatMod@TOV (CUP E83C23000330006) awarded to the Department of Mathematics, University of Rome Tor Vergata and the Italian Research Center on High Performance Computing, Big Data and Quantum Computing (CUP E83C22003230001). The last two authors are members of Gruppo Nazionale per il Calcolo Scientifico, Istituto Nazionale di Alta Matematica.

## Appendix A. LR TB-spline refinement algorithm

The splitting procedure described in Step 2 of Definition 13 can be subdivided into primary and secondary splits according to Remark 14. The corresponding implementation of LR TB-splines is summarized in Algorithm 1.

**Algorithm 1:** LR TB-spline refinement

**Input :** $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_{j-1})$ TB-spline set; $\gamma_j$ Split

**Output:** $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)$ Refined TB-spline set

$\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j) \leftarrow \mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_{j-1})$ and $\hat{\mathcal{S}}_{\boldsymbol{p}}(\mathcal{M}_j) \leftarrow \emptyset$;
`/* Primary split: */`
Extend $\gamma_j$ by any existing split in $\mathcal{M}_{j-1}$ that intersects and is collinear with $\gamma_j$;
**for** *every TB-spline* $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}} \in \mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_{j-1})$ **do**
  **if** $\gamma_j$ *traverses support of* $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}$ **then**
    Refine $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}$ and compute splitting coefficients $\nu_k^{(1)}, \nu_k^{(2)}$ from (11);
    Update the weights according to (14);
    Remove $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}$ from $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)$ and add new TB-splines to $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)$ and $\hat{\mathcal{S}}_{\boldsymbol{p}}(\mathcal{M}_j)$;
  **end**
**end**
`/* Secondary split: */`
**while** $\hat{\mathcal{S}}_{\boldsymbol{p}}(\mathcal{M}_j)$ *is non-empty* **do**
  **for** *every TB-spline* $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}} \in \hat{\mathcal{S}}_{\boldsymbol{p}}(\mathcal{M}_j)$ **do**
    **for** *every existing split* $\gamma_\ell$ *in* $\mathcal{M}_{j-1}$ **do**
      **if** $\gamma_\ell$ *traverses support of* $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}$ **then**
        Refine $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}$ and compute splitting coefficients $\nu_k^{(1)}, \nu_k^{(2)}$ from (11);
        Update the weights according to (14);
        Remove $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}$ from $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)$ and add new TB-splines to $\mathcal{S}_{\boldsymbol{p}}(\mathcal{M}_j)$ and $\hat{\mathcal{S}}_{\boldsymbol{p}}(\mathcal{M}_j)$;
      **end**
    **end**
    Remove $N_{\boldsymbol{\Xi}_k,\boldsymbol{p}}$ from $\hat{\mathcal{S}}_{\boldsymbol{p}}(\mathcal{M}_j)$;
  **end**
**end**

## References


[1] 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*, Computer Methods in Applied Mechanics and Engineering **199** (2010), 229–263.

[2] C. Bracco, D. Berdinsky, D. Cho, M. Oh, and T. Kim, *Trigonometric generalized T-splines*, Computer Methods in Applied Mechanics and Engineering **268** (2014), 540–556.

[3] C. Bracco and D. Cho, *Generalized T-splines and VMCR T-meshes*, Computer Methods in Applied Mechanics and Engineering **280** (2014), 176–196.

[4] C. Bracco, T. Lyche, C. Manni, F. Roman, and H. Speleers, *Generalized spline spaces over T-meshes: Dimension formula and locally refined generalized B-splines*, Applied Mathematics and Computation **272** (2016), 187–198.

[5] ———, *On the dimension of Tchebycheffian spline spaces over planar T-meshes*, Computer Aided Geometric Design **45** (2016), 151–173.

[6] C. Bracco, T. Lyche, C. Manni, and H. Speleers, *Tchebycheffian spline spaces over planar T-meshes: Dimension bounds and dimension instabilities*, Journal of Computational and Applied Mathematics **349** (2019), 265–278.

[7] A. Bressan, *Some properties of LR-splines*, Computer Aided Geometric Design **30** (2013), 778–794.

[8] A. Bressan and B. Jüttler, *A hierarchical construction of LR meshes in 2D*, Computer Aided Geometric Design **37** (2015), 9–24.

[9] A. N. Brooks and T. J. R. Hughes, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering **32** (1982), 199–259.

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

[11] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng, *Polynomial splines over hierarchical T-meshes*, Graphical models **70** (2008), 76–86.

[12] T. Dokken, T. Lyche, and K. F. Pettersen, *Polynomial splines over locally refined box-partitions*, Computer Aided Geometric Design **30** (2013), 331–356.

[13] C. Giannelli, B. Jüttler, and H. Speleers, *THB-splines: The truncated basis for hierarchical splines*, Computer Aided Geometric Design **29** (2012), 485–498.

[14] R. R. Hiemstra, T. J. R. Hughes, C. Manni, H. Speleers, and D. Toshniwal, *A Tchebycheffian extension of multidegree B-splines: Algorithmic computation and properties*, SIAM Journal on Numerical Analysis **58** (2020), 1138–1163.

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

[16] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert, *A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations*, Computer Methods in Applied Mechanics and Engineering **73** (1989), 173–189.

[17] K. A. Johannessen, M. Kumar, and T. Kvamsdal, *Divergence-conforming discretization for Stokes problem on locally refined meshes using LR B-splines*, Computer Methods in Applied Mechanics and Engineering **293** (2015), 38–70.

[18] K. A. Johannessen, T. Kvamsdal, and T. Dokken, *Isogeometric analysis using LR B-splines*, Computer Methods in Applied Mechanics and Engineering **269** (2014), 471–514.

[19] K. A. Johannessen, F. Remonato, and T. Kvamsdal, *On the similarities and differences between classical hierarchical, truncated hierarchical and LR B-splines*, Computer Methods in Applied Mechanics and Engineering **291** (2015), 64–101.

[20] G. Kermarrec, V. Skytt, and T. Dokken, *Optimal surface fitting of point clouds using local refinement: Application to GIS data*, Springer Nature, 2023.

[21] T. Lyche, C. Manni, and H. Speleers, *Foundations of spline theory: B-splines, spline approximation, and hierarchical refinement*, Splines and PDEs: From Approximation Theory to Numerical Linear Algebra (T. Lyche, C. Manni, and H. Speleers, eds.), Lecture Notes in Mathematics, vol. 2219, Springer, Cham, 2018, pp. 1–76.

[22] ———, *Tchebycheffian B-splines revisited: An introductory exposition*, Advanced Methods for Geometric Modeling and Numerical Simulation (C. Giannelli and H. Speleers, eds.), Springer INdAM Series, vol. 35, Springer, Cham, 2019, pp. 179–216.

[23] C. Manni, F. Pelosi, and M. L. Sampoli, *Generalized B-splines as a tool in isogeometric analysis*, Computer Methods in Applied Mechanics and Engineering **200** (2011), 867–881.

[24] ———, *Isogeometric analysis in advection–diffusion problems: Tension splines approximation*, Journal of Computational and Applied Mathematics **236** (2011), 511–528.

[25] C. Manni, F. Pelosi, and H. Speleers, *Local hierarchical h-refinements in IgA based on generalized B-splines*, Mathematical Methods for Curves and Surfaces 2012 (M. Floater et al., eds.), Lecture Notes in Computer Science, vol. 8177, Springer–Verlag, Heidelberg, 2014, pp. 341–363.

[26] M. L. Mazure, *How to build all Chebyshevian spline spaces good for geometric design?*, Numerische Mathematik **119** (2011), 517–556.
[27] B. Mourrain, *On the dimension of spline spaces on planar T-meshes*, Mathematics of Computation **83** (2014), 847–871.
[28] F. Patrizi, *Effective grading refinement for locally linearly independent LR B-splines*, BIT Numerical Mathematics **62** (2022), no. 4, 1745–1764.
[29] F. Patrizi and T. Dokken, *Linear dependence of bivariate minimal support and locally refined B-splines over LR-meshes*, Computer Aided Geometric Design **77** (2020), 101803.
[30] F. Patrizi, C. Manni, F. Pelosi, and H. Speleers, *Adaptive refinement with locally linearly independent LR B-splines: Theory and applications*, Computer Methods in Applied Mechanics and Engineering **369** (2020), 113230.
[31] K. Raval, C. Manni, and H. Speleers, *Tchebycheffian B-splines in isogeometric Galerkin methods*, Computer Methods in Applied Mechanics and Engineering **403** (2023), 115648.
[32] D. F. Rogers, *An introduction to NURBS: With historical perspective*, Morgan Kaufmann Publisher, San Francisco, 2001.
[33] I. J. Schoenberg, *On spline functions*, Inequalities (O. Shisha, ed.), Academic Press, New York, 1967, pp. 255–286.
[34] L. L. Schumaker, *Spline functions: Basic theory*, third ed., Cambridge University Press, Cambridge, 2007.
[35] L. L. Schumaker and L. Wang, *Approximation power of polynomial splines on T-meshes*, Computer Aided Geometric Design **29** (2012), 599–612.
[36] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri, *T-splines and T-NURCCs*, ACM Transactions on Graphics **22** (2003), 477–484.
[37] V. Skytt and T. Dokken, *Scattered data approximation by LR B-spline surfaces: A study on refinement strategies for efficient approximation*, Geometric Challenges in Isogeometric Analysis (C. Manni and H. Speleers, eds.), Springer INdAM Series, vol. 49, Springer, Cham, 2022, pp. 217–258.
[38] H. Speleers, *Algorithm 1020: Computation of multi-degree Tchebycheffian B-splines*, ACM Transactions on Mathematical Software **48** (2022), 12.
[39] H. Speleers, C. Manni, F. Pelosi, and M. L. Sampoli, *Isogeometric analysis with Powell-Sabin splines for advection-diffusion-reaction problems*, Computer Methods in Applied Mechanics and Engineering **221–222** (2012), 132–148.
[40] I. Stangeby and T. Dokken, *Properties of spline spaces over structured hierarchical box partitions*, Isogeometric Analysis and Applications 2018 (H. van Brummelen et al., eds.), Lecture Notes in Computational Science and Engineering, vol. 133, Springer, Cham, 2021, pp. 177–207.
[41] C. Zimmermann and R. A. Sauer, *Adaptive local surface refinement based on LR NURBS and its application to contact*, Computational Mechanics **60** (2017), 1011–1031.