

Properties of Spline Spaces Over Structured Hierarchical Box Partitions



Ivar Stangeby and Tor Dokken

Abstract Given a spline space spanned by Truncated Hierarchical B-splines (THB), it is always possible to construct a spline space spanned by Locally Refined B-splines (LRB) that contains the THB-space. Starting from configurations where the two spline spaces are equal, we address what happens to the properties of the LRB-space when it is modified by local one-directional refinement at convex corners of, and along edges between dyadic refinement regions. We show that such local modifications can reduce the number of B-splines over each element to the minimum prescribed by the polynomial bi-degree, and that such local refinements can be used for improving the condition numbers of mass and stiffness matrices.

1 Introduction

The use of Hierarchical B-splines (HB) introduced in [2] has gained much attention in Isogeometric Analysis (IgA) in recent years. Hierarchical B-splines are based on a dyadic sequence of grids determined by scaled lattices. On each hierarchical level a spline space is defined as the tensor product of univariate spline spaces spanned by uniform B-splines.

Hierarchical B-splines do not constitute a partition of unity, a much desired property in both Computer Aided Design (CAD) and IgA. As a remedy to this Truncated Hierarchical B-splines (THB) [5, 13] were introduced, where B-splines on one hierarchical level are suitably *truncated* by B-splines from finer hierarchical levels when the support of a B-spline at a finer level is contained in the support of a B-spline at a coarser level.

An alternative to the THB-approach for forming a partition of unity came with the introduction of Locally Refined B-splines (LRB) [1], where initial tensor product B-splines are split until only B-splines of minimal support remain. LRB permits

I. Stangeby · T. Dokken (✉)
SINTEF, Oslo, Norway
e-mail: ivar.stangeby@sintef.no; tor.dokken@sintef.no

dyadic refinement of hierarchical meshes while ensuring that all B-splines have minimal support. In the occasional case where meshlines at a dyadic level are too short to split an LR B-spline, the meshlines in question are extended. This fact ensures that the spline space spanned by THB-splines is either identical to or constitutes a subset of the LRB spline space.

In IgA *open knot vectors* are used to simplify the interpolation of boundary conditions, as reported for THB in [4] and for LRB in [6]. In *open knot vectors* the multiplicity at boundary knots is set to $m = d + 1$. An alternative approach is to use B-splines with knot multiplicity of $m = 1$ along the boundary. In order to force the partition of unity in this case, a ghost domain is added around the domain of interest, as seen in [7] for both THB and LRB. This distinction is illustrated for univariate cubic splines in Fig. 1.

In Sect. 2 we address the effects the choice of boundary knot multiplicity has on condition numbers. To distinguish between single multiplicity and open knots at the domain boundary we prefix any method using single knot multiplicity on the boundary with a ghosted domain by an “S”. Using this naming convention, the methods addressed in [6] are respectively S-THB and S-LRB. In this paper, we show that for the same tensor product spline space, THB and LRB are superior with respect to condition numbers of mass and stiffness matrices compared to respectively S-LRB and S-THB. We also explain the intriguing near constant behaviour of the condition numbers reported in [7], where S-LRB and S-THB were addressed, and condition numbers seemed to be nearly independent of the refinement level. We show that this is due to single knot multiplicity at domain boundaries for the examples presented in [7]. Further it is shown that for more levels of refinement the condition numbers for the mass matrix for S-THB and S-LRB will meet and then follow the growing curves for respectively THB and LRB.

In HB and THB the refinement procedure (at an element level) consists of marking elements for splitting. Marked elements are subsequently split in both parameter directions. This contrasts with the refinement procedure LRB allows, namely that of splitting an element in a single parameter direction at the time, provided that at least one B-spline is split in the process. This can be used to modify the hierarchical refinement, and possibly improve the approximation properties of the resulting spline space. In the remaining sections we use open knot vectors at domain boundaries and address how such modifications influence the condition numbers for mass and stiffness matrices for bi-cubic spline spaces in particular. The remaining sections are structured as follows:

Section 3 gives a lightweight introduction to box-partitions and spline spaces over such partitions. The starting point for THB and LRB refinement is a tensor product spline space. The key concept of element overloading is defined, the situation where more B-splines cover an element than are needed for spanning the polynomial space over the element. We briefly summarize some key properties. Subsequently, we recall the definitions of both LRB and THB splines. We also relate the refinement strategies for LRB to T-splines [12]. Those readers already familiar with the contents of this section may feel free to skip it.

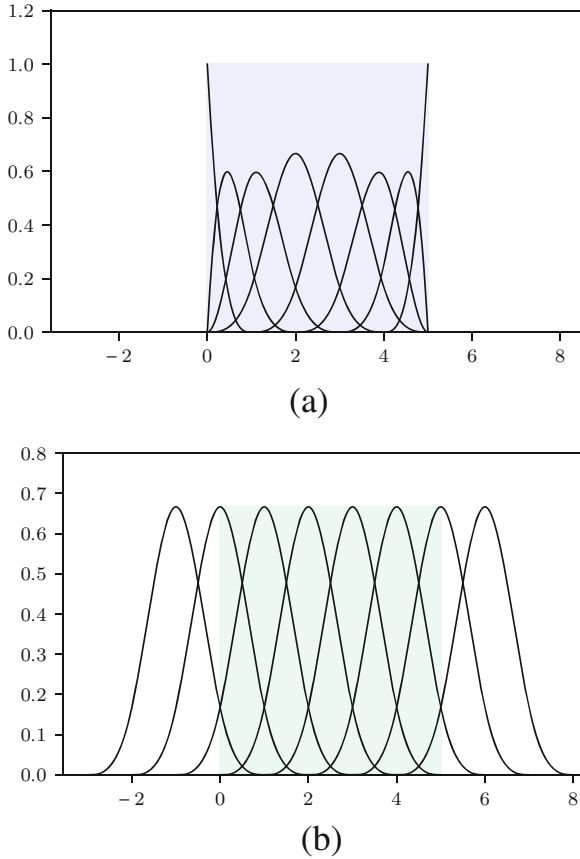


Fig. 1 Spline spaces over the domain $\Omega = [0, 5]$. In **(a)**, the partition of unity is satisfied at the boundary by setting the knot multiplicity to $m = d + 1 = 4$. In **(b)**, the partition of unity is satisfied at the boundary by extending the domain to allow the full polynomial space to be spanned at the boundary elements. The shaded regions indicate the domain Ω , and the spline space spanned by the B-splines over Ω are the same in both cases

Section 4 takes a look at overloading. We look at how to reduce or completely remove overloaded regions in a mesh. We showcase some specific overloading patterns that occur for hierarchical refinement of THB and LRB. Furthermore, we show how local modifications to the LRB-mesh reduce overloading as well as condition numbers.

Section 5 provides a quantitative comparison between the methods. We conduct our numerical experiments using modified central and diagonal refinement examples from [7] with a finer initial tensor-product mesh. This gives enough room on each hierarchical level for the local modifications to take place. The examples show that LRB with no overloading have smaller condition numbers for the mass matrix per degree of freedom than THB and LRB with overloading.

However, the difference seems to be so small that in general all methods have a similar behaviour.

Section 6 summarizes the main results of this paper.

2 Condition Numbers and Knotline Multiplicity at Domain Boundary

In [7], hierarchical refinement was performed for five levels of refinement, using S-THB and S-LRB. The results reported that the number of refinement levels had little to no influence on the evolution of condition numbers for stiffness and mass matrices. There were some minute differences between S-THB and S-LRB, but they followed the same trend. In Fig. 2 we display the condition number of the mass matrix for up to eight refinement levels for S-THB, S-LRB, THB and LRB when run on a hierarchical mesh from [7]. The relevant mesh at the fifth refinement level is displayed in Fig. 4.

The results from [7] is reproduced, and corresponds to the S-THB and S-LRB curves for the first five refinements. However, at the sixth refinement, the curve for the condition number of the mass matrix for S-LRB breaks off and grows exponentially following the curves of LRB that starts three orders of magnitude lower. In Fig. 2 there are also two additional curves (S-LRB1 and LRB1). These are added to show that modifying the mesh by inserting additional knot lines in one parameter direction, with the effect of reducing overloading, significantly reduces the condition numbers of LRB-refinement. This modified mesh is shown in Fig. 4b. We will discuss such modifications more closely in Sect. 4.

Multiplicity of domain boundary knot lines also influences the condition number of the stiffness matrix, as seen in Fig. 3. Here we see that the condition numbers for single boundary knot multiplicity (S-THB, S-LRB and S-LRB1) are two orders of magnitude higher than the condition numbers for open knot vectors (THB, LRB, LRB-1) (Fig. 4).

2.1 Boundary Knotline Multiplicities

We now take a stab at explaining the drastic change in behaviour occurring at $n = 6$ refinements for the S-LRB and S-THB methods as shown in Fig. 2. Since the condition number of a matrix are computed in terms of its largest and smallest eigenvalues, we decided to take a look at the geometric localization of the eigenvectors corresponding to these eigenvalues. By coloring the hierarchical mesh based on the influence of each in terms of the corresponding coefficient in the eigenvector, we obtained a rudimentary geometric visualization of these eigenvectors. In Fig. 5, we see the smallest eigenvector for the mass-matrix corresponding to LRB and S-

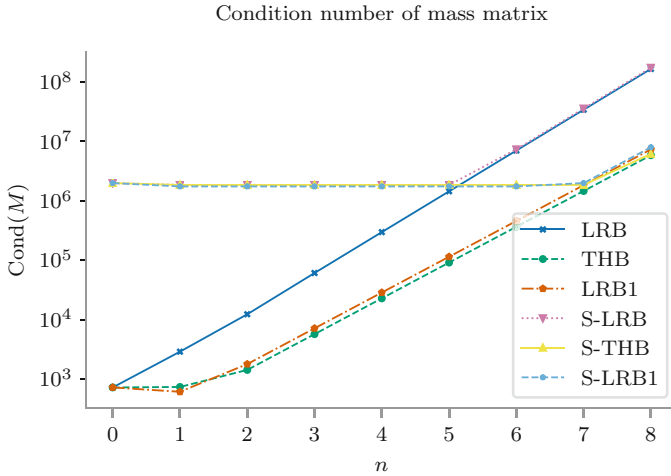


Fig. 2 The condition number of the mass matrix. We see that under repeated refinement, the condition numbers corresponding to spline spaces with open knot vectors (THB, LRB) tends towards the condition numbers corresponding to spline spaces with single knots (S-THB, S-LRB). We also see that a small local modification to reduce overloading in the LRB-space reduces the condition number of the mass matrix (S-LRB1, LRB1)

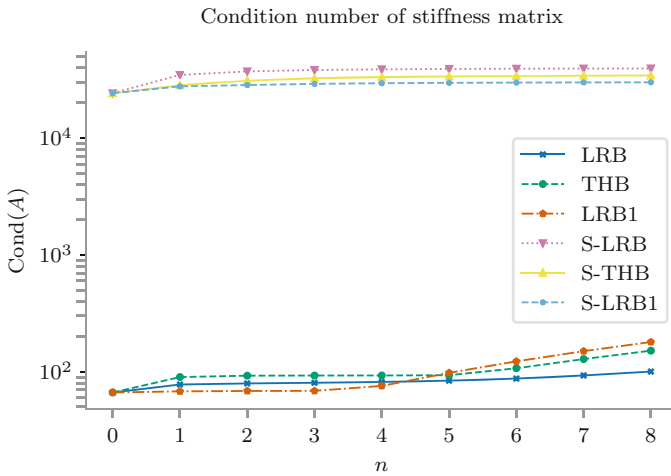
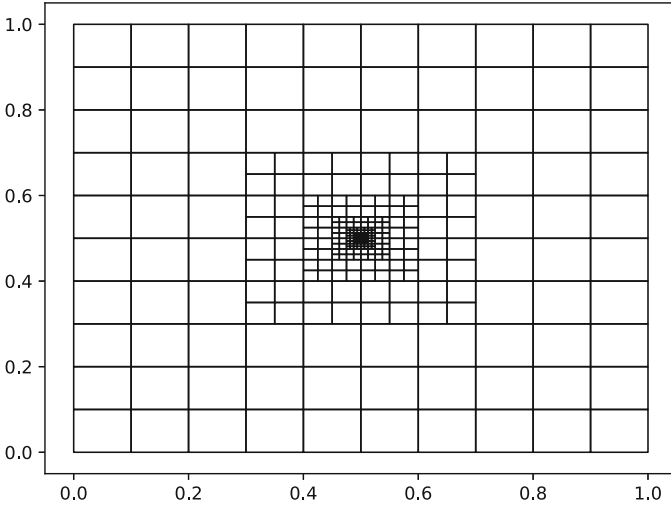
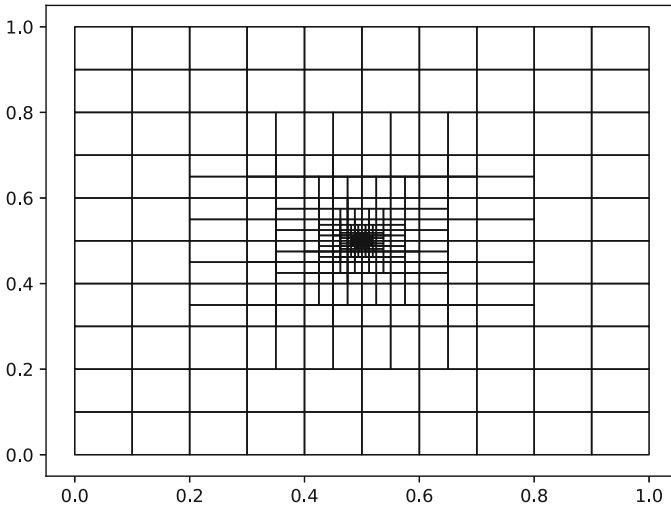


Fig. 3 The condition number of the stiffness matrix. Here the separation between S-LRB, S-THB, LRB and THB are seen in even greater effect



(a)



(b)

Fig. 4 The meshes used for the preliminary comparison. In (a), the unmodified mesh used for S-THB, S-LRB, THB and LRB. In (b) the modified mesh used for S-LRB1 and LRB1. This mesh generates a few extra degrees of freedom

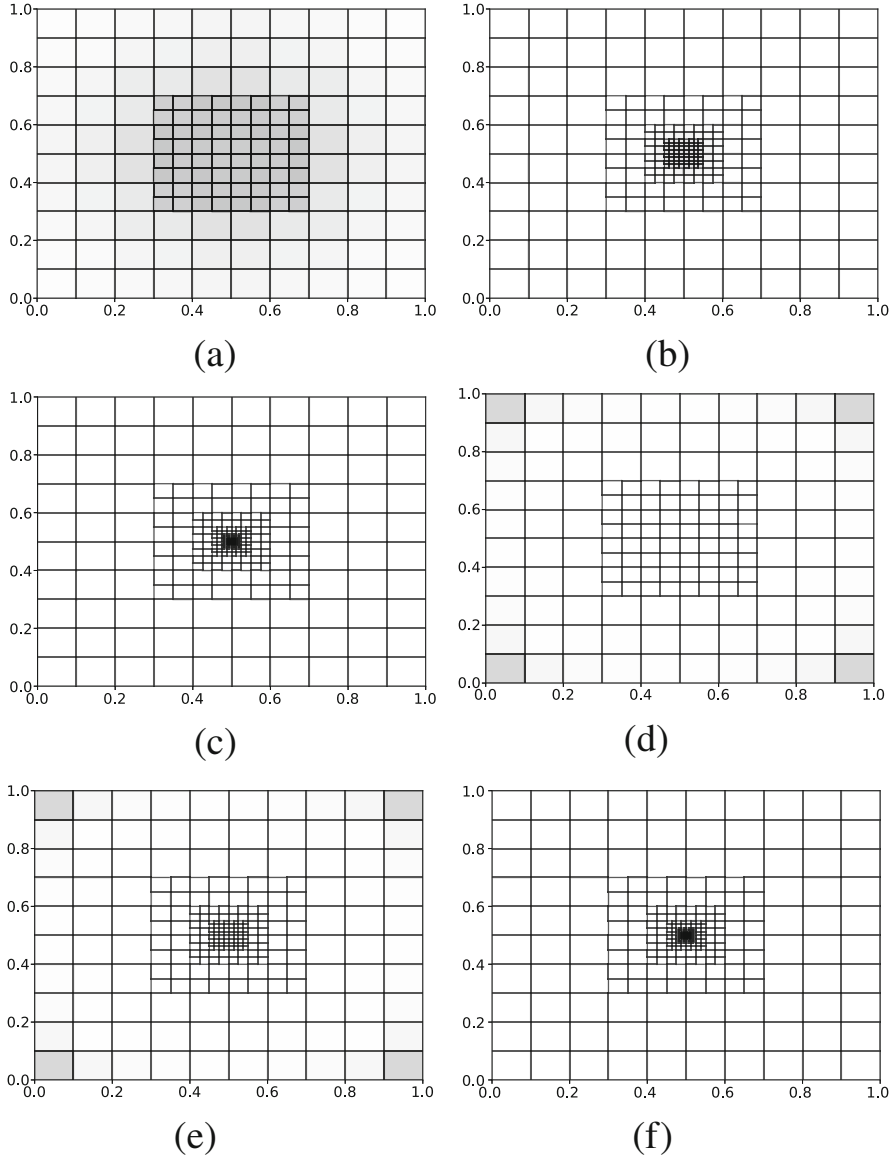


Fig. 5 The eigenvectors corresponding to the smallest eigenvalue of the mass matrix for LRB (a) to (c), and for S-LRB visualized over the hierarchical mesh after one, three and six refinements (d) to (f). Darker color indicates higher influence. As we see, the smallest eigenvalues for LRB is localized in the refined region after only one refinement. On the other hand, S-LRB is localized in the corners of the domain up until but not including six refinements, as shown for $n = 1$ and $n = 3$. The effect of the locally refined region dominates only after $n = 6$ refinements as in (f). (a) $n = 1$ (LRB). (b) $n = 3$ (LRB). (c) $n = 6$ (LRB). (d) $n = 1$ (S-LRB). (e) $n = 3$ (S-LRB). (f) $n = 6$ (S-LRB)

LRB at the first, third and sixth refinement, and in Fig. 6, the corresponding largest eigenvector.

These figures correspond to the behaviour observed in Fig. 2 where the conditioning for LRB grows after only one refinement, whereas S-LRB needs six refinements before the behaviour in the refined region is registered.

The reason for this behaviour is due to the size of the B-splines defined along the boundary in comparison to the size of the B-splines defined in the interior of the domain. In order to illustrate this, we compute analytically the entries in the mass matrix corresponding to B-splines on various tensor product level and compare these values to the mass matrix entry corresponding to a B-spline defined in the corner of the domain.

2.1.1 Observation for the Mass Matrix

Over the domain $\Omega = [0, 1] \times [0, 1]$ we define a tensor product grid with element size ℓ . In the case of bi-cubic spline spaces, the B-spline defined in the lower left corner of the domain can for LRB and S-LRB be written in terms of their knots as

$$\begin{aligned} B &:= B[\mathbf{x}]B[\mathbf{y}], \\ Q &:= B[\mathbf{s}]B[\mathbf{t}], \end{aligned} \tag{1}$$

where $\mathbf{x} = \mathbf{y} = [0, 0, 0, 0, \ell]$ and $\mathbf{s} = \mathbf{t} = [-3\ell, -2\ell, -\ell, 0, \ell]$. In both cases, the two B-splines have only one element of support in the domain Ω , namely $\beta := [0, \ell] \times [0, \ell]$. To get a feel for the differences in influence on the mass matrix these B-splines have, we compute the corresponding diagonal elements in the mass matrix.

The polynomial restrictions of B and Q to the element β is

$$\begin{aligned} B|_{\beta}(x, y) &= \frac{(\ell - x)^3(\ell - y)^3}{\ell^6}, \\ Q|_{\beta}(x, y) &= \frac{(\ell - x)^3(\ell - y)^3}{36\ell^6}. \end{aligned} \tag{2}$$

In other words, $B|_{\beta} = 36Q|_{\beta}$. If we now compute the diagonal mass matrix entries corresponding to these two elements, we obtain the following:

$$\int_{\beta} B^2 = \frac{\ell^2}{49}, \quad \int_{\beta} Q^2 = \frac{\ell^2}{49} \cdot \frac{1}{36^2}. \tag{3}$$

We here see that the matrix element corresponding to the corner B-spline Q arising in S-LRB is three orders of magnitude smaller than the matrix element corresponding to B . Recall the disparity between the curves in Fig. 2, where the differences in the condition numbers also were three orders of magnitude.

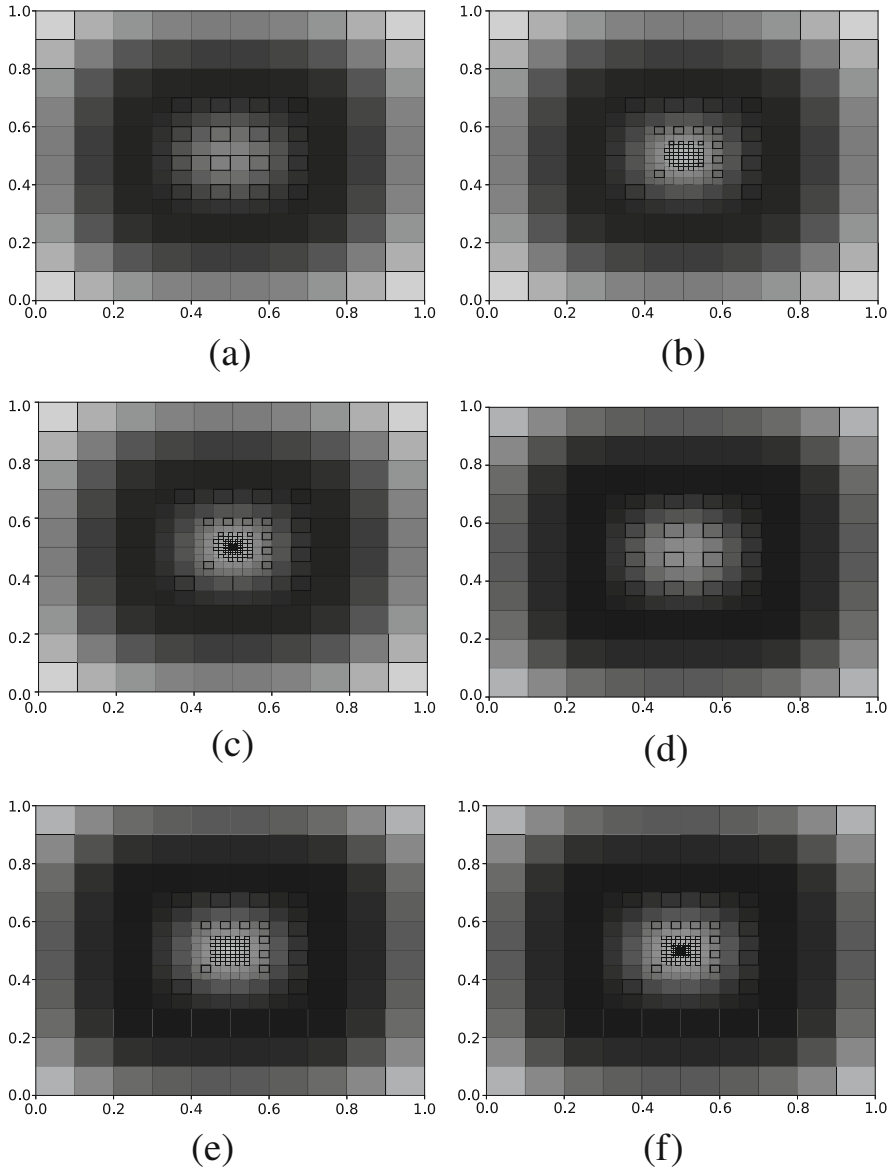


Fig. 6 The eigenvectors corresponding to the largest eigenvalue of the mass matrix for LRB (a) to (c) and for S-LRB visualized over the hierarchical mesh after one, three and six refinements (d) to (f). Darker color indicates higher influence. For the largest eigenvalue, the two methods approximately correspond geometrically, and the largest eigenvalues are constant over refinement levels for each of the methods. (a) $n = 1$ (LRB). (b) $n = 3$ (LRB). (c) $n = 6$ (LRB). (d) $n = 1$ (S-LRB). (e) $n = 3$ (S-LRB). (f) $n = 6$ (S-LRB)

Remark 1 This relation between Q and B is both dimension and degree-dependent. The effect will be magnified for higher spatial dimension and higher polynomial degree. A similar condition number analysis was performed for embedded methods in [10], with scaling effects reported along the lines of the effects reported above.

3 Box Partitions, Meshes, and Spline Spaces

In order for the paper to be self-contained, we review the concept of *box partitions* and spline spaces over such partitions in the following section. Readers already familiar with these notions may feel free to skip this section. While the construction generalizes to any dimension, we will gradually focus our attention to the two-dimensional case, as this is most relevant for our discussion. A fully general treatment can be found in [1, 8]. The fundamental building block of a box partition is the d -dimensional box.

Definition 1 A **box** β in \mathbb{R}^d (or d -**box**) is the Cartesian product of d closed finite intervals J_1, \dots, J_d :

$$\beta = \prod_{i=1}^d J_i. \quad (4)$$

The **dimension** of β is defined to be the number of non-trivial intervals in its definition, and is denoted $\dim(\beta)$. We call a d -box of dimension d an **element**, while a d -box of dimension $d - 1$ is called a **mesh-rectangle**. To any mesh-rectangle, we associate an integer k corresponding to which parametric dimension its trivial component resides, and we call the mesh-rectangle a k -**mesh-rectangle** if this has to be emphasized.

In the two-dimensional setting ($d = 2$), a **meshline** is a one-dimensional mesh-rectangle.

Remark 2 Note that these naming-conventions are independent of the dimension of the ambient space. Hence, a *mesh-rectangle* may very well be something different from a rectangle. As an example, a mesh-rectangle in \mathbb{R}^4 is an axis aligned box. Furthermore, the integer k corresponding to any mesh-rectangle encodes the *direction* of the mesh-rectangle. In the two-dimensional case, where mesh-rectangles are lines, a 1-mesh-rectangle is a vertical line, and a 2-mesh-rectangle is a horizontal line.

As customary in discretizations of computational domains, a large domain is partitioned into a set of non-overlapping smaller geometrical entities. We call such a partition in this specific setting a *box partition*, and this is more precisely defined as follows:

Definition 2 Let $\Omega \subset \mathbb{R}^d$ be an element (d -box of dimension d). A finite collection \mathcal{E} of elements is said to be a **box partition** of Ω if

1. $\beta_i^\circ \cap \beta_j^\circ = \emptyset$ for all $\beta_i, \beta_j \in \mathcal{E}$ where $\beta_i \neq \beta_j$, and
2. $\bigcup_{\beta \in \mathcal{E}} \beta = \Omega$.

In other words, a box partition is an interior-disjoint partition of Ω into a set of smaller elements.

Associated to any element β is its *boundary*, which naturally consists of boxes of dimension one less, i.e., mesh-rectangles. Given a box partition of a larger element Ω , it is therefore sensible to discuss the set of mesh-rectangles associated to this box partition.

Definition 3 (Informal) Given a box partition \mathcal{E} of a domain Ω , we may naturally associate a set of mesh-rectangles \mathcal{M} called a **box mesh** on Ω formed by taking unions of element boundaries.

Remark 3 The link between a box partition \mathcal{E} and the associated box mesh \mathcal{M} is such that by knowing one of them you may recover the other. The box mesh *generated* by a box partition is denoted $\mathcal{M}(\mathcal{E})$, and the box partition *generated* by a box mesh is denoted $\mathcal{E}(\mathcal{M})$.

As our ultimate goal is to define spline spaces based on tensor-product splines over box-partitions, we need to have a concept of knot multiplicity in this more general setting.

Definition 4 A **box mesh with multiplicity** is a pair (\mathcal{M}, μ) where $\mu: \mathcal{M} \rightarrow \mathbb{N}$ associates to each mesh-rectangle γ a positive integer $\mu(\gamma)$, called the **multiplicity** of the mesh-rectangle. Note that this is completely analogous to the notion of knot multiplicity for univariate B-splines.

Definition 5 Let a polynomial multi-degree $p = (p_1, \dots, p_d)$ as well as a box mesh with multiplicity (\mathcal{M}, μ) corresponding to the box-partition \mathcal{E} of a d -dimensional domain Ω be given. The **spline space of degree p over \mathcal{M}** is defined as

$$\begin{aligned}
 S_p^\mu(\mathcal{M}) := & \left\{ f: \Omega \rightarrow \mathbb{R} : f|_\beta \in \Pi_p \text{ for all } \beta \in \mathcal{E} \right. \\
 & \text{and } f \in C^{p_k - \mu(\gamma)} \text{ for all } k\text{-mesh-rectangles } \gamma \in \mathcal{M}, \\
 & \left. \text{with } k = 1, \dots, d \right\}. \quad (5)
 \end{aligned}$$

A dimension formula for general spline spaces over box partitions was presented in [9]. In general, the dimension depends on both the topological properties of the box partition and the parametrization of the box partition. In the two-dimensional case—with some requirements on the length of the constituent meshlines—the

formula reduces to a formula depending only on the topological features of the mesh. We consider this outside the scope of this text, and refer the reader to [9] for details.

In order to compute with spline spaces over box partitions of the form above, we must be able to construct a set of basis functions that span this space. Several constructions has been studied. We will only be dealing with Truncated Hierarchical B-splines, and Locally Refined B-splines.

Before we move on, we define the notion of a *hierarchical mesh*, a type of box partition over which spline bases of the aforementioned type may be defined. This provides a common ground for comparison of the two methods. The construction is simple and relies on marking regions for which a tensor product mesh of various refinement levels is used.

Definition 6 Let Ω be a domain, and let $\mathcal{M}_1 \subset \dots \subset \mathcal{M}_M$ be a sequence of nested tensor product meshes on Ω . Let $\Omega_1 \supset \dots \supset \Omega_M$ be a set of nested subsets of Ω whose boundaries $\partial\Omega_\ell$ align with the meshlines of the corresponding mesh on the coarser level $\mathcal{M}_{\ell-1}$ for $\ell = 2, \dots, M$. The **hierarchical mesh** \mathcal{M} is defined as

$$\mathcal{M} = \{ \gamma \cap \Omega_\ell : \gamma \in \mathcal{M}_\ell \text{ for } \ell = 1, \dots, M \}, \quad (6)$$

i.e., \mathcal{M} consists of meshlines from each level intersected with the corresponding region, see Fig. 7.

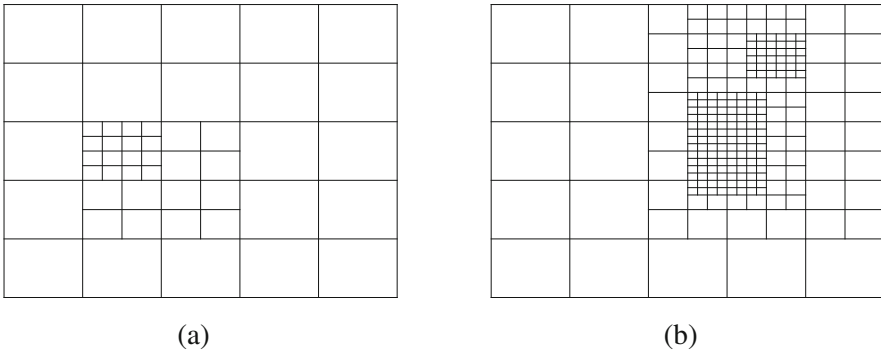


Fig. 7 Two examples of hierarchical meshes. In (a) a mesh consisting of three levels of refinement, and in (b) a mesh with four levels of refinement. Note here that the region residing at level $\ell = 4$ consists of two disjoint components

3.1 Tensor Product Splines

The foundation for all the locally refined spline spaces over box partitions addressed in this paper, is the tensor product B-spline. We start by glossing over some preliminary definitions.

Recall that a univariate B-spline of polynomial degree p relies on exactly $p + 2$ knots. This observation enables us to define B-splines locally without referring to some global knot vector.

Definition 7 Given a polynomial degree p and a non-decreasing knot-vector $\mathbf{t} = (t_1, \dots, t_{p+2})$, we recursively define the **univariate B-spline** $B[\mathbf{t}]: \mathbb{R} \rightarrow \mathbb{R}$ as follows:

If $p = 0$, then

$$B[\mathbf{t}] = \begin{cases} 1, & x \in [t_1, t_2); \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

If $p > 0$, then

$$B[\mathbf{t}](x) = \frac{x - t_1}{t_{p+1} - t_1} B[\mathbf{t}^-](x) + \frac{t_{p+2} - x}{t_{p+2} - t_2} B[\mathbf{t}^+](x), \tag{8}$$

where \mathbf{t}^+ and \mathbf{t}^- are obtained by dropping the first and last elements of \mathbf{t} respectively:

$$\mathbf{t}^+ = (t_2, \dots, t_{p+2}), \quad \mathbf{t}^- = (t_1, \dots, t_{p+1}). \tag{9}$$

In the cases of a vanishing denominator, the whole term is taken to be zero.

Such univariate splines can be easily extended to higher dimensions through a tensor product construction.

Definition 8 Let the polynomial multi-degree $p = (p_1, \dots, p_d)$ and the d local knot vectors $\mathbf{t}^1, \dots, \mathbf{t}^d$ be given. The **d -variate tensor product B-spline** $B[\mathbf{t}^1, \dots, \mathbf{t}^d]: \mathbb{R}^d \rightarrow \mathbb{R}$ is then defined as

$$B[\mathbf{t}^1, \dots, \mathbf{t}^d](\mathbf{x}) = \prod_{i=1}^d B[\mathbf{t}^i](x_i), \tag{10}$$

where $\mathbf{x} = (x_1, \dots, x_d)$.

The **support** of $B[\mathbf{t}^1, \dots, \mathbf{t}^d]$ is the closure of the area where the B-spline takes non-zero values, which we denote by:

$$\text{supp}(B[\mathbf{t}^1, \dots, \mathbf{t}^d]) = \overline{\left\{ \mathbf{x} \in \mathbb{R}^d : B[\mathbf{t}^1, \dots, \mathbf{t}^d](\mathbf{x}) \neq 0 \right\}}. \quad (11)$$

Since our B-spline construction is inherently local, we need to know when a tensor product B-spline has *minimal support* with respect to some box mesh.

Definition 9 (Informal) A B-spline $B = B[\mathbf{t}^1, \dots, \mathbf{t}^d]$ has **support on** (\mathcal{M}, μ) if all the knot lines of B occurs as meshlines in \mathcal{M} . We say that B has **minimal support on** (\mathcal{M}, μ) if in addition, all the knot lines of B occur consecutively in (\mathcal{M}, μ) .

One of the central concepts we will be addressing in this paper is the *overloading* of elements. We make this precise in the following definition.

Definition 10 Let a box partition \mathcal{E} of a domain Ω and a polynomial multi-degree $p = (p_1, \dots, p_d)$ be given. Assume that we construct a set \mathcal{B} of B-splines degree p over the mesh \mathcal{M} corresponding to \mathcal{E} . We say that an element β is **overloaded with respect to** \mathcal{B} if the number of B-splines with support on β is larger than the dimension of the corresponding space of polynomials over this element, namely

$$\dim(\Pi_p(\beta)) = \prod_{i=1}^d (p_i + 1). \quad (12)$$

We now proceed to review the definitions of LR B-splines and THB-splines.

3.2 Locally Refined Spline Spaces

In preparation for the following discussion, we will adopt the notational convention as in [7] in order to differentiate between the distinct types of basis functions. Depending on the underlying box partition, some of these types may coincide.

Type	Basis	Function
Tensor Product B-spline	\mathcal{B}	B
Truncated Hierarchical B-spline	\mathcal{H}	H
LR B-spline	\mathcal{L}	L

Furthermore, in this and the following sections we will be dealing with box partitions and spline spaces in \mathbb{R}^2 , unless otherwise stated.

3.2.1 LR-Splines

Locally Refined B-splines (LRB or LR-splines) was introduced by [1]. The LR-spline framework permits the insertion of local splits in a tensor product mesh, and subsequently enables local refinement of the mesh. Being scaled tensor product B-splines, LR-splines admit a set of nice properties. The set of LR B-splines form a partition of unity. Their scaling weights are positive, meaning that they satisfy the convex hull property, and are therefore inherently stable in computations. Moreover, with some restrictions on the refinement process, linear independence of the resulting set of functions can be guaranteed.

LR-splines are defined over so-called *LR-meshes*, being special box partitions. Starting from an initial tensor product mesh, meshlines are inserted sequentially, yielding a sequence of box-meshes, where no meshline is allowed to terminate in the middle of an element. This is formalized in the following definition, and Fig. 8 gives an example.

Definition 11 An **LR mesh** is a box mesh $\mathcal{M} = \mathcal{M}_N$ resulting from a sequence of meshline insertions in an initial tensor product mesh \mathcal{M}_1 . That is

$$\mathcal{M}_{i+1} = \mathcal{M}_i + \gamma_i \tag{13}$$

for $i = 1, \dots, N - 1$ where each intermediate mesh is a box mesh.

Remark 4 We often think of an LR-mesh as a *sequence* of intermediate meshes

$$\mathcal{M} = \mathcal{M}_N \supseteq \mathcal{M}_{N-1} \supseteq \dots \supseteq \mathcal{M}_2 \subseteq \mathcal{M}_1 \tag{14}$$

as each intermediate step is needed for the LR B-spline construction.

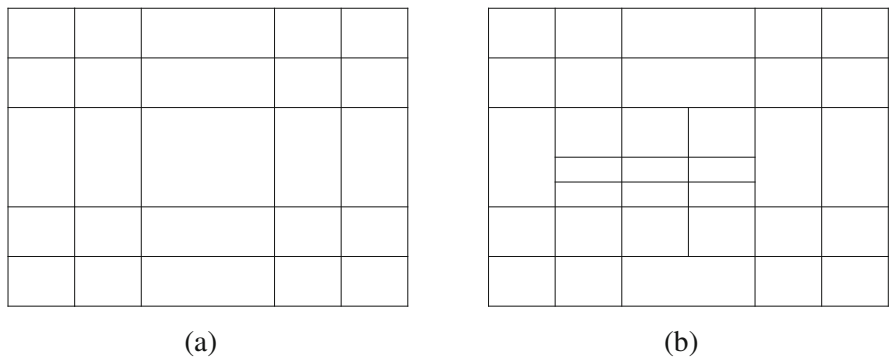


Fig. 8 In (a), an initial tensor product mesh, which is also an LR-mesh. In (b), an LR-mesh obtained from the insertion of three meshlines in the initial tensor product mesh from (a)

Over such an LR-mesh we may define the associated set of LR B-splines algorithmically. Starting from an initial space of tensor product B-splines, meshlines are inserted sequentially. Whenever a meshline completely traverses the support of a B-spline, the B-spline is *split* according to the knot insertion procedure, and two new B-splines are added. The B-spline that was split is removed.

Definition 12 Let \mathcal{M} be an LR-mesh over a domain Ω and $p = (p_1, p_2)$ a polynomial bi-degree. We define the set $\mathcal{L}(\mathcal{M})$ of **LR B-splines of degree p over \mathcal{M}** algorithmically as in Algorithm 3.1.

Algorithm 3.1 The LR B-spline construction

```

Let  $\mathcal{L}_1 := \mathcal{B}(\mathcal{M}_1)$  be the set of tensor product B-splines on  $\mathcal{M}_1$ .
for each intermediate mesh  $\mathcal{M}_{i+1} = \mathcal{M}_i + \gamma_i$ , with  $i = 1, \dots, N - 1$  do
   $\mathcal{L}_{i+1} := \mathcal{L}_i$ 
  while there exists  $B \in \mathcal{L}_{i+1}$  without minimal support on  $\mathcal{M}_{i+1}$  do
     $B^+, B^- = \text{SPLIT}(B, \gamma_i)$  ▷ knotline insertion
     $\mathcal{L}_{i+1} = (\mathcal{L}_{i+1} \setminus \{B\}) \cup \{B^+, B^-\}$  ▷ update the set of B-splines
  end while
end for
 $\mathcal{L}(\mathcal{M}) := \mathcal{L}_N$ 

```

Remark 5 Note that all LR B-splines have *minimal support* on the resulting mesh. This is by construction. However, there is an important distinction to be made, namely that the set of LR B-splines *differ* from the set of minimal support B-splines on the resulting mesh. This is due to the LR refinement procedure putting some restrictions on the resulting mesh. A survey on the properties of LR-splines and minimal support B-splines are given in [8].

3.2.2 Truncated Hierarchical B-Splines

Hierarchical B-splines, first introduced in [2], is a method for specifying locally refined spline spaces on hierarchical meshes. Recall that a hierarchical mesh consists of regions corresponding to various levels of tensor product grids. The hierarchical B-spline construction involves replacing any B-spline with support completely contained in a region of a finer level by B-splines at this finer level. This procedure will, however, lead to coarse B-splines partially overlapping the finer regions, and does not constitute a partition of unity.

A remedy to this problem came with the introduction of Truncated Hierarchical B-splines [5], where B-splines on a coarse level are *truncated* by B-splines on a finer level. This leads to the resulting set of B-splines forming a partition of unity. The construction relies on the *truncation operator*. Recall that a spline $f \in \text{span}(\mathcal{B}_\ell)$

can be represented in terms of the finer basis $\mathcal{B}_{\ell+1}$:

$$f = \sum_{B_i \in \mathcal{B}_{\ell+1}} c_i^{\ell+1}(f) B_i, \tag{15}$$

where $c_i^{\ell+1}(f)$ is the coefficient multiplying B_i in the representation of f in terms of $\mathcal{B}_{\ell+1}$. For uniform B-splines, this relation is often called the *two-scale relation*. The truncation operator is defined as follows:

Definition 13 Let $B \in \mathcal{B}_\ell$ be a coarse B-spline. The **truncation** with respect to the set of fine B-splines $\mathcal{B}_{\ell+1}$ and the corresponding region $\Omega_{\ell+1}$ is

$$\text{trunc}^{\ell+1} B := \sum_{\substack{B_i \in \mathcal{B}_{\ell+1} \\ \text{supp} B_i \not\subseteq \Omega_{\ell+1}}} c_i^{\ell+1}(B) B_i. \tag{16}$$

Remark 6 The definition above represents the truncation operator in an *additive* sense, where the contributions from the finer level are summed up. It is also possible to represent the truncation operator *subtractively*, by instead removing the bits of the representation that have been replaced by finer B-splines:

$$\text{trunc}^{\ell+1} B = B - \sum_{\substack{B_i \in \mathcal{B}_{\ell+1} \\ \text{supp} B_i \subseteq \Omega_{\ell+1}}} c_i^{\ell+1}(B) B_i \tag{17}$$

Definition 14 Let \mathcal{M} be a hierarchical mesh over a domain Ω (see Definition 6) and $p = (p_1, p_2)$ a polynomial bi-degree. On each level $\ell = 1, \dots, N$ we have a tensor product spline space V_ℓ spanned by a collection of B-splines $\mathcal{B}_\ell = \mathcal{B}(\mathcal{M}_\ell)$. We define the set of **THB-splines of degree p over \mathcal{M}** algorithmically as in Algorithm 3.2.

Algorithm 3.2 The THB-spline construction

Let $\mathcal{H}_1 = \mathcal{B}(\mathcal{M}_1)$ be the set of tensor product B-splines on \mathcal{M}_1 .

for each level $\ell = 1, \dots, N - 1$ **do**

$$\mathcal{H}_{\ell+1}^{\text{trunc}} := \left\{ \text{trunc}^{\ell+1} H : H \in \mathcal{H}_\ell \text{ and } \text{supp}(H) \not\subseteq \Omega_{\ell+1} \right\}$$

$$\mathcal{H}_{\ell+1}^{\text{new}} := \left\{ B \in \mathcal{B}_{\ell+1} : \text{supp}(B) \subseteq \Omega_{\ell+1} \right\}$$

$$\mathcal{H}_{\ell+1} := \mathcal{H}_{\ell+1}^{\text{trunc}} \cup \mathcal{H}_{\ell+1}^{\text{new}}$$

end for

$\mathcal{H} = \mathcal{H}_N$.

Remark 7 Note that in the cases where a B-spline $B \in \mathcal{B}_\ell$ is truncated with respect to $\mathcal{B}_{\ell+1}$ and $\Omega_{\ell+1}$ and the support of B happen to be entirely contained in $\Omega_{\ell+1}$, the truncation operator completely removes the coarse B-spline. In the THB-spline construction, this has the effect of replacing the coarse B-splines with fine B-splines defined in its support.

Remark 8 A simple framework for the implementation of truncated hierarchical B-splines is given in [3], and this serves as a good introduction to the many ways such splines have been implemented in the literature. Efficient algorithms for the assembly of finite element matrices are also presented.

3.2.3 T-Splines

While not directly addressed in this paper, we briefly mention T-splines as LRB with local modifications to the LR-meshes used in this paper will reproduce the spline space generated by semi-standard T-splines [12] and Analysis Suitable T-splines [11]. An example of an Analysis Suitable T-mesh in the index domain is displayed in Fig. 9 to the left, with the corresponding LR-mesh to the right. This is a close up of the structure of a mesh similar to the one used in Fig. 11b.

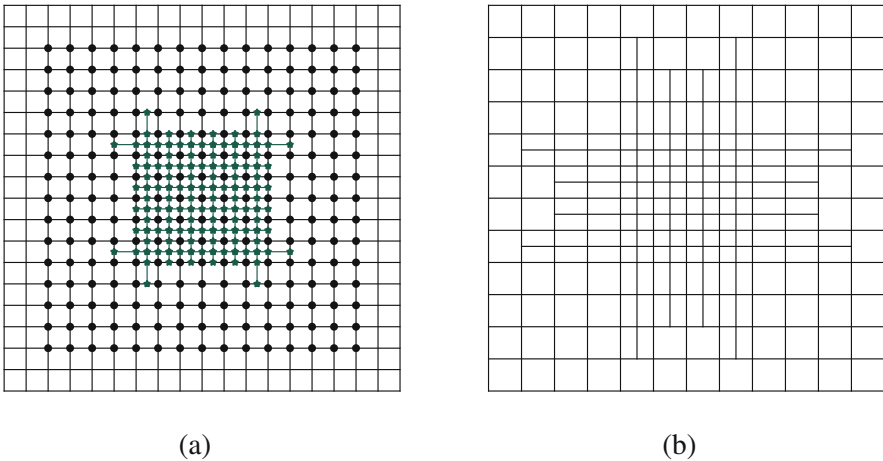


Fig. 9 In (a) a T-spline mesh in the index domain. The dots denote Greville points or “anchors” for each individual B-spline. A black dot is a B-spline at level $\ell = 0$ and a green star a B-spline at level $\ell = 1$. The resulting spline space can be replicated by an LR-mesh without overloaded elements (c.f. Fig. 11b), as displayed in (b). Here we have used multiplicity $m = 4$ along the boundary

4 Local Modification of Meshes and the Reduction of Overloading

In this section we take a deeper look at the overloading of elements, and how local modifications to the mesh may be used to remedy this. Recall from the previous definition that an element β in a box-partition \mathcal{E} is said to be *overloaded* if the number of supported B-splines on the element exceeds the number needed to span the full polynomial space over the element.

We are interested in such overloaded regions, because by reducing or removing completely the overloading on elements we may

1. reduce the bandwidth of the resulting finite element matrices; and
2. improve conditioning of finite element matrices.

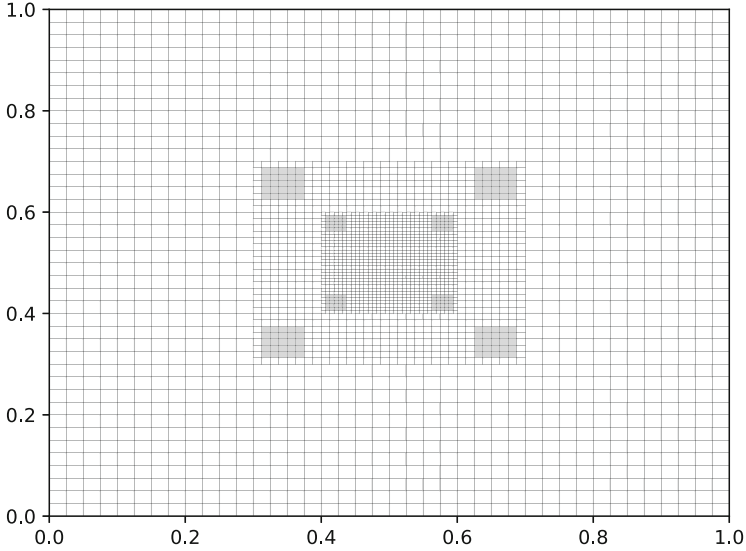
Such overloaded regions occur for LRB in convex corners of a fine hierarchical level, where a large B-spline from one hierarchical level overlaps several elements of a finer hierarchical level. For THB, overloading occurs along any border between two hierarchical levels. By coloring in elements with too many supported B-splines we obtain a visualization of this phenomenon, as seen in Fig. 10 on a hierarchical mesh with three levels of refinement.

In order to reduce, or completely remove such overloaded regions, we may for LRB extend meshlines from the fine hierarchical level to the coarse level, in order to split the culprit B-splines. The length needed for this extended meshline depends on the polynomial degree of the B-spline to be split. In Fig. 11 we see the effects of two types of meshline extension to the LRB-mesh from Fig. 10 for a space of bi-cubic splines. The corresponding splines are named LRBNO and T-LRBNO, signifying the fact that these local modifications completely remove overloading.

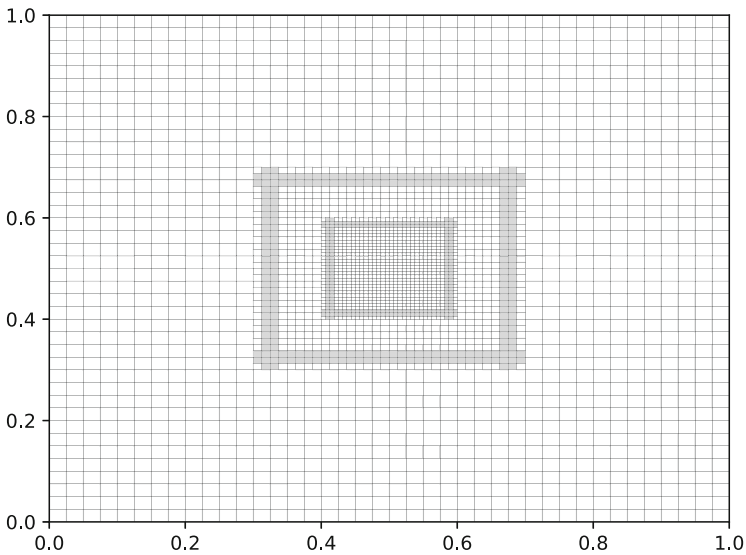
In order to capture what is happening, we take a closer look at overloading in a convex corner in Fig. 12 where we show how B-splines from the coarse level of a hierarchical mesh may overlap with B-splines from the fine level in such a way that too many B-splines are active over a given element.

5 Numerical Experiments

In order to compare the methods addressed in this paper, we assemble the mass and stiffness matrices associated to discretizations of partial differential equations using IgA or FEM. By computing the condition number of these matrices, we obtain a metric useful for comparison. These matrices arise amongst others in the discretizations of the Poisson equation, and in the computation of the L^2 -projection of a function.

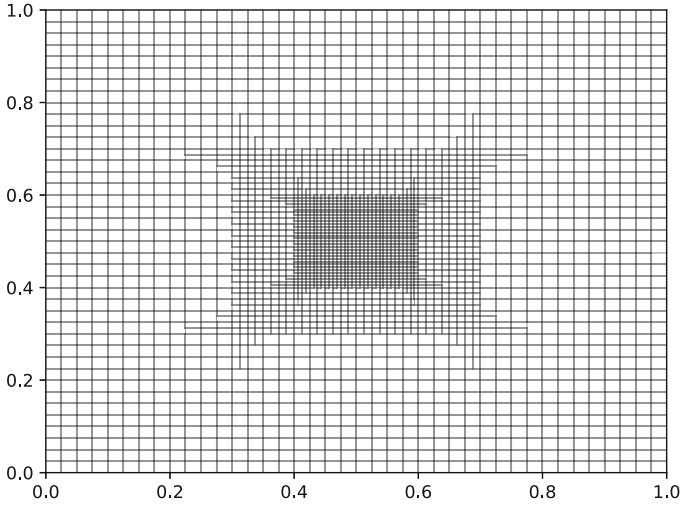


(a)

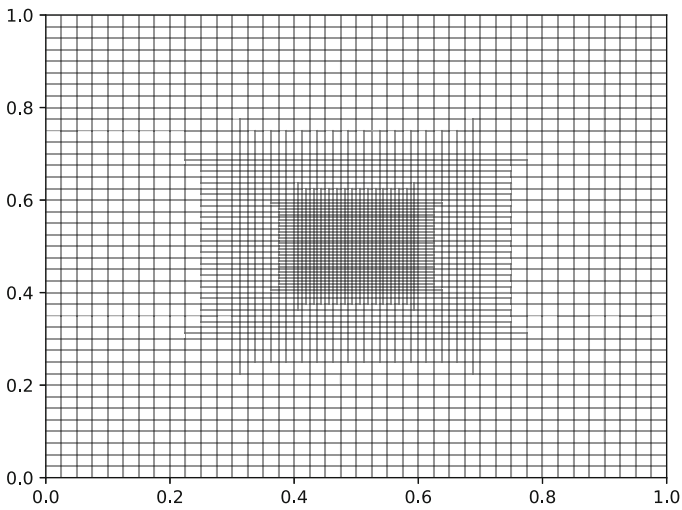


(b)

Fig. 10 The overloading patterns on a hierarchical mesh with two levels of refinement. In (a), we see that regions in the corners of the refined regions are overloaded, due to the influence of four LR B-splines from the coarser layer, whose support has not been split by any newly introduced meshlines. In (b) we observe “bands” of overloaded elements along the boundary between two consecutive refinement levels for THB, arising due to the fact that fine B-splines must be completely contained in the support of a coarse B-spline before truncation occurs. (a) LRB. (b) THB



(a)



(b)

Fig. 11 Two different local modifications with the effect of completely removing the overloaded elements. In (a) we extend the meshlines closest to the convex corner by three elements, and the meshline next to them by one element. This has the effect of completely removing the overloading on the corner elements. In (b), we make a mesh that can be defined using T-splines that has no overloading. As in (a) meshlines closest to the corners are extended by three, while meshlines at the borders between refinement levels are extended by two as in Fig. 9. (a) LRBNO. (b) T-LRBNO

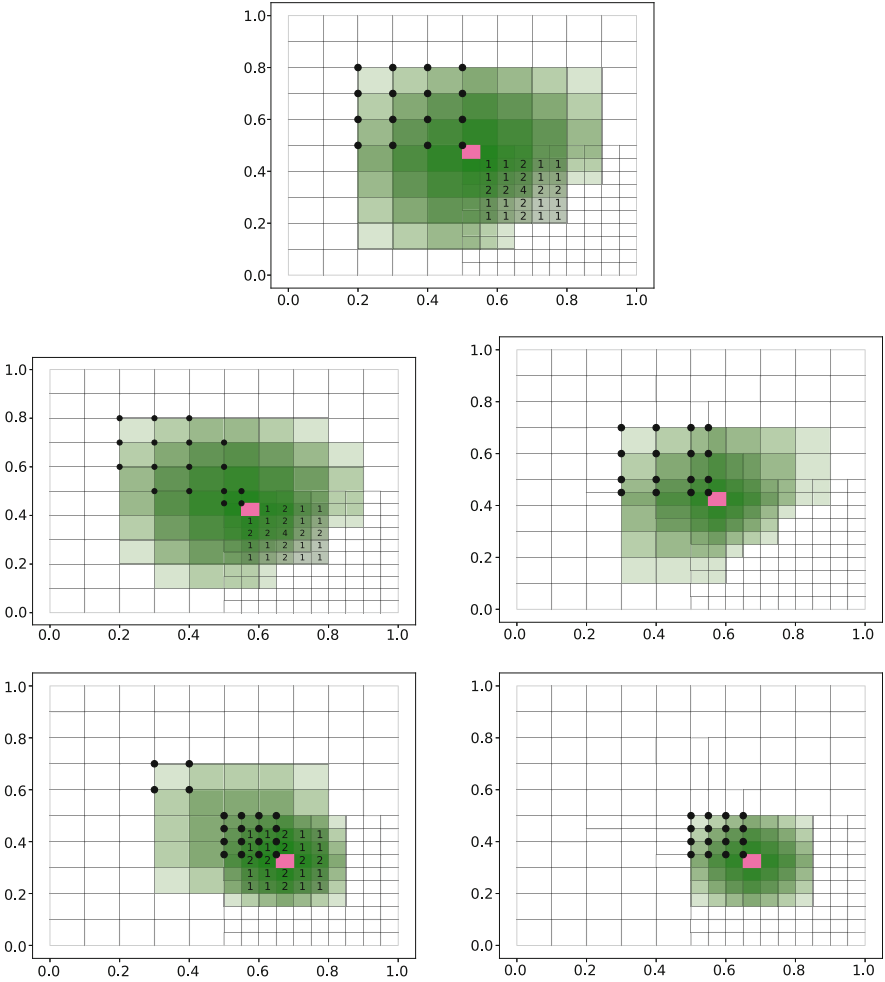


Fig. 12 The effects of extending meshlines on the bi-cubic B-splines covering the element in pink. The upper left corner of each B-spline is marked with a black dot. The knotlines of each B-spline can be identified by starting from the dot and going four knotlines to the right/down. We chose to not use Greville points as some overloaded configurations produce overlapping Greville points. In the *upper mesh* we look at the element just inside the corner of the region refined, and no overloading occurs. In the *middle meshes* we move one element diagonally into the refined region. Before refinement the overloading is one, and after additional lines are inserted the overloading is removed. In the *bottom meshes* we move two additional element diagonally into the refined region. Before refinement the overloading is four, after additional lines are inserted the overloading is removed

5.1 L^2 -Projection

Given a domain Ω , a function $f: \Omega \rightarrow \mathbb{R}$ in some space of functions V , and a finite-dimensional subspace V_h of V , we are interested in finding the function $u \in V_h$ that minimizes the $L^2(\Omega)$ -error

$$\|e\|_{L^2} = \|u - f\|_{L^2}. \quad (18)$$

This can be reformulated as a variational equation by requiring u to satisfy

$$\int_{\Omega} u v d\Omega = \int_{\Omega} f v d\Omega, \quad (19)$$

for all $v \in V_h$. By introducing a basis $\{\varphi_1, \dots, \varphi_N\}$ for V_h , which in our case will be one of the THB or LRB-bases, we may write this as a linear equation

$$\mathbf{M}\mathbf{c} = \mathbf{b}, \quad (20)$$

where \mathbf{M} is the *mass matrix* and \mathbf{c} is the vector of coefficients representing u in our chosen basis. The entries for \mathbf{M} and the right-hand side \mathbf{b} are given as

$$\mathbf{M}_{ij} = \int_{\Omega} \varphi_i \varphi_j d\Omega, \quad \mathbf{b}_j = \int_{\Omega} f \varphi_j d\Omega. \quad (21)$$

5.2 The Poisson Equation

A commonly encountered differential equation is the *Poisson equation*. Given a function $f: \Omega \rightarrow \mathbb{R}$, we wish to find a function u in a space of admissible functions V such that

$$-\Delta u = f \text{ in } \Omega, \quad (22)$$

subject to the boundary conditions

$$u = 0 \text{ on } \Gamma_D, \quad \frac{\partial u}{\partial \mathbf{n}} = g \text{ on } \Gamma_N. \quad (23)$$

Here Γ_D denotes the *Dirichlet*-boundary and Γ_N the *Neumann*-boundary. We assume $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Furthermore, \mathbf{n} is the outward facing boundary normal to Ω and g is the prescribed flux along the boundary. This is called the strong form of the Poisson equation.

By multiplying the strong form with a suitable test function, and integrating over the domain, we obtain the variational form of the Poisson equation. The

requirements on the smoothness of the sought solution u can be relaxed, by moving some derivatives onto the test-functions. Again, we seek the solution u in a subspace V_h of V spanned by a set of basis functions $\{\varphi_1, \dots, \varphi_N\}$. The variational form of the Poisson-equation then reads

$$\int_{\Omega} \nabla u \cdot \nabla v d\Omega = \int_{\Gamma_N} g v dS - \int_{\Omega} f v d\Omega, \quad (24)$$

for all $v \in V_h$. Rewriting this in terms of the basis functions, we obtain the system of linear equations

$$\mathbf{A} \mathbf{c} = \mathbf{b}, \quad (25)$$

where \mathbf{A} is the *stiffness matrix* of the problem. The entries of \mathbf{A} and \mathbf{b} are given as

$$\mathbf{A}_{ij} = \int_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j d\Omega, \quad \mathbf{b}_j = \int_{\Gamma_N} \varphi_j g dS - \int_{\Omega} f \varphi_j d\Omega. \quad (26)$$

5.3 Condition Numbers

The *condition number* of a matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ quantifies how sensitive the solution \mathbf{x} to the linear system $\mathbf{B} \mathbf{x} = \mathbf{y}$ is to small perturbations both in \mathbf{B} and the right-hand side \mathbf{y} and is formally defined as

$$\text{Cond}(\mathbf{B}) := \|\mathbf{B}\| \|\mathbf{B}^{-1}\|, \quad (27)$$

where $\|\cdot\|$ is some matrix norm. Note that the condition number is norm-dependent, but all matrix norms are equivalent on $\mathbb{R}^{n \times n}$. We will be computing the condition numbers in the 2-norm, and in this specific setting for *normal* matrices the condition number can be computed as the ratio between the largest and smallest eigenvalue

$$\text{Cond}(\mathbf{B}) = \frac{|\lambda_1(\mathbf{B})|}{|\lambda_n(\mathbf{B})|}. \quad (28)$$

Here $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, i.e., ordered in a decreasing fashion.

As in [7], we chose to estimate the condition numbers of the matrices *before* imposing any boundary conditions, as imposing boundary conditions can have a large impact on the conditioning of the matrix. The mass matrix \mathbf{M} is non-singular, even with no imposed boundary condition. The stiffness matrix \mathbf{A} however will be singular, and have a zero-eigenvalue of multiplicity one.

In addition to this, the computation of the smallest eigenvalue of a matrix is a numerically unstable procedure. We will therefore *estimate* the condition numbers as follows:

$$\text{Cond}(\mathbf{M}) \approx \frac{|\lambda_1(\mathbf{M})|}{|\lambda_n(\mathbf{M})|}, \text{ and } \text{Cond}(\mathbf{A}) \approx \frac{|\lambda_1(\mathbf{A})|}{|\lambda_{n-1}(\mathbf{A})|}, \quad (29)$$

using the second-smallest eigenvalue for the stiffness matrix.

5.4 Numerical Results

Below we present results of the numerical simulations. As LRBNO generates higher dimensional spline spaces than THB and LRB we plot the condition numbers as a function of the degrees of freedom. Just plotting the condition numbers as a function of the levels provides less information. By including the dimension of the spline space we obtain a clearer distinction between the methods.

5.4.1 Central Refinement

We assemble the stiffness and mass matrices on a sequence of meshes corresponding to central refinement, shown at the third refinement for bi-cubic splines in Figs. 10 and 11. In addition to the bi-cubic case, we also assemble the matrices on similar meshes for bi-quadratic and bi-quartic spline spaces, where the spacing between each refined region is kept the same for all degrees. The results are shown in Figs. 13 and 14. Unfortunately, due to time constraints, we were not able to get results for bi-quartic THB-splines.

Start by noting that for the mass matrix, THB performs better than LRB with no modifications, while for the stiffness matrices, the two methods are comparable with LRB having a slight advantage. The number of degrees of freedom are the same. By locally modifying the mesh, as is the case for LRBNO and T-LRBNO, we see that the number of degrees of freedom goes up, as expected. The condition number per degree of freedom is smallest for T-LRBNO.

5.4.2 Diagonal Refinements

We now consider the case of diagonal refinement for bi-cubic spline spaces. Again, we use the same hierarchical mesh for LRB and THB. We will only consider one sequence of meshes with local modifications. In the diagonal refinement setting, the corners of the refined region are sufficiently close to each other so that we need to make a decision on which direction to refine in. The diagonally refined mesh is not compatible with a T-spline type mesh, and will therefore not be taken into consideration here.

We assemble stiffness and mass matrices on the meshes displayed in Fig. 15. The results are shown in Figs. 16 and 17. Note that for the diagonal refinement, the number of degrees of freedom generated when removing overloading, shown in

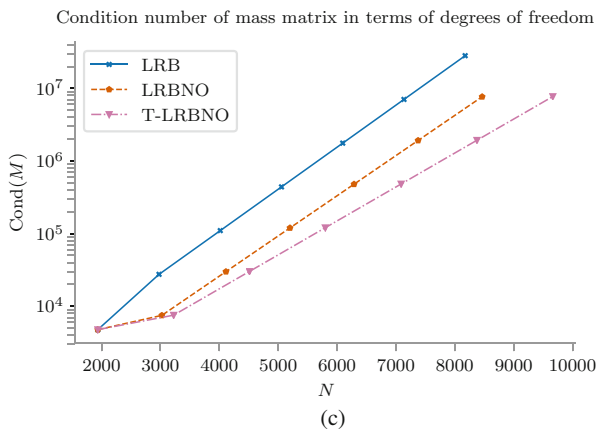
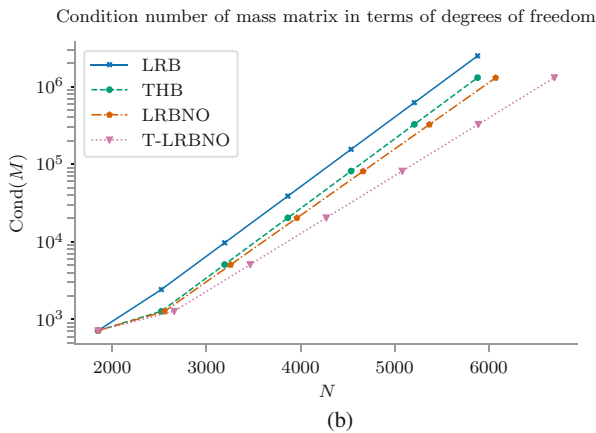
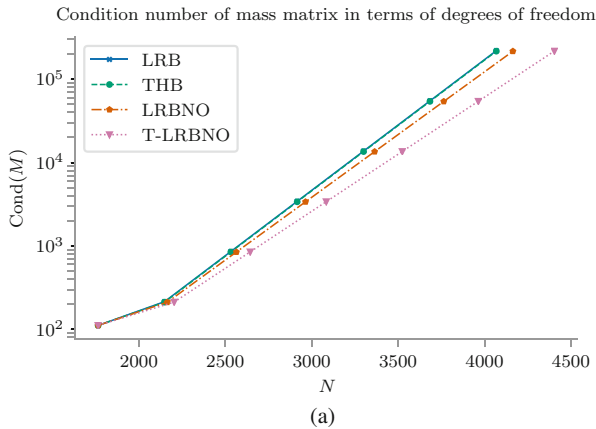


Fig. 13 The condition numbers for mass matrices over a centrally refined hierarchical mesh for six levels of refinement. In the figures N denotes the number of degrees of freedom in the corresponding space. (a) Quadratic. (b) Cubic. (c) Quartic

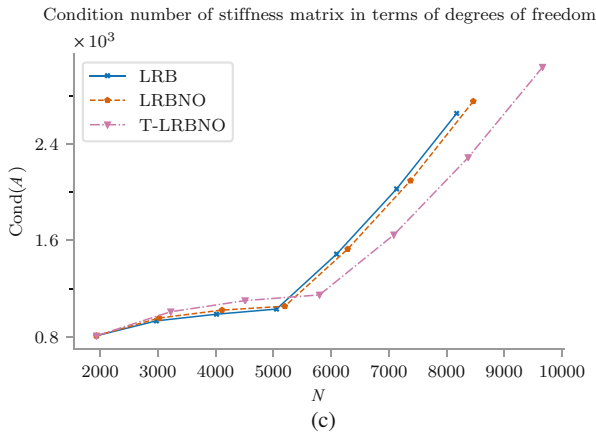
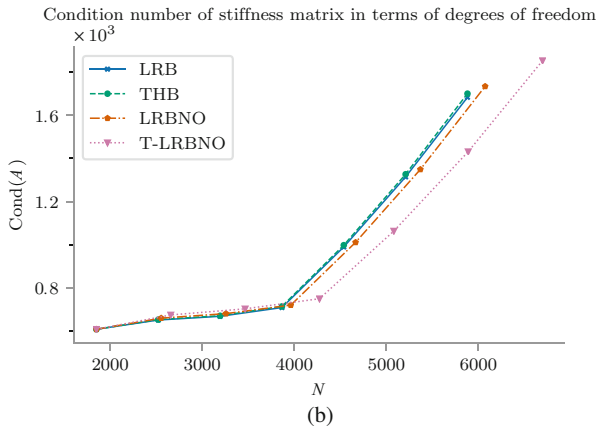
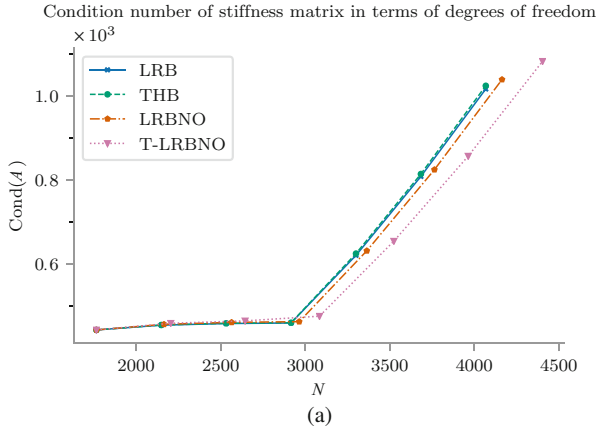
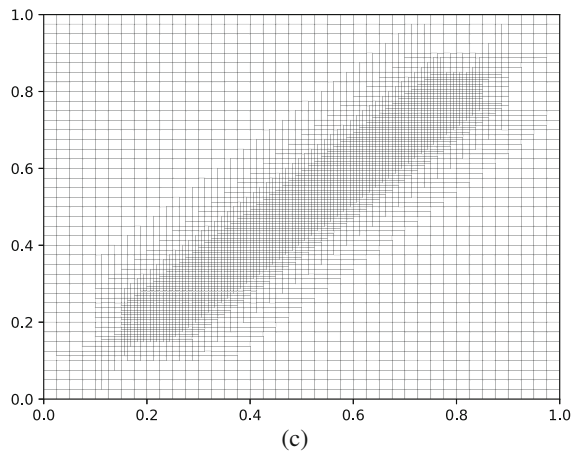
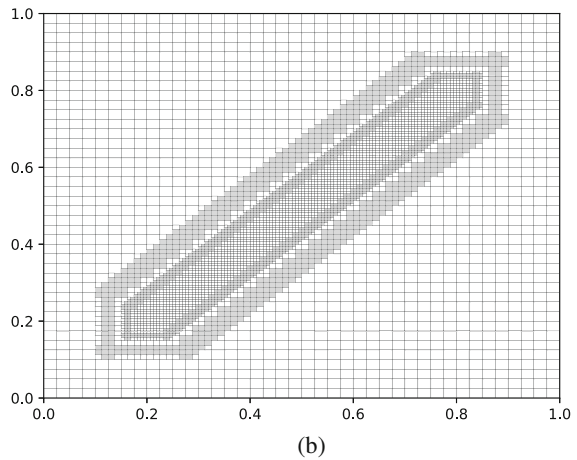
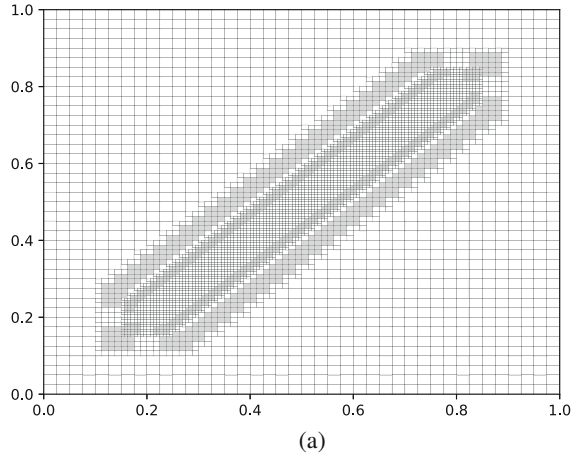


Fig. 14 The condition numbers for stiffness matrices over a centrally refined hierarchical mesh for six levels of refinement. In the figures N denotes the number of degrees of freedom in the corresponding space. (a) Quadratic. (b) Cubic. (c) Quartic

Fig. 15 The overloading patterns on a hierarchical mesh with three levels of diagonal refinement. In this case, we see in greater effect the behaviour of LRB over convex corners. Here the difference in overloading between THB and LRB are smaller, as opposed to the central refinement case, due to the high number of corners relative to the length of the sides of the refined levels. By using a one-directional meshline extension along the diagonal, and extensions similar to the central-refinement case, we may completely remove overloading. **(a)** LRB. **(b)** THB. **(c)** LRBNO



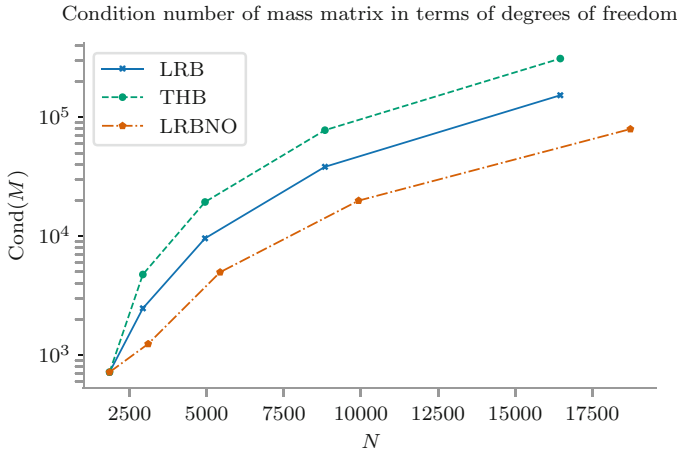


Fig. 16 Condition numbers for mass matrices over a diagonally refined hierarchical mesh for four levels of refinement. There is one data point for each method at each refinement level. The first point is the same for all methods as the all methods start from the same tensor product spline space. N denotes the number of degrees of freedom in the corresponding spline space. The none overloaded LRBNO mesh has clearly the smallest condition numbers

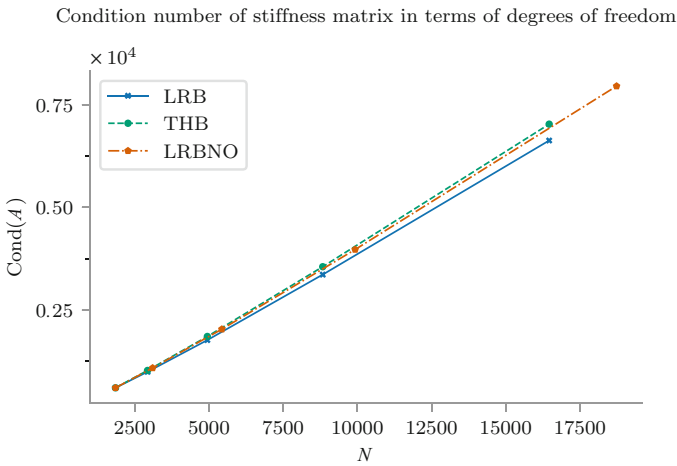


Fig. 17 The condition numbers for stiffness matrices over a diagonally refined hierarchical mesh for four levels of refinement. In the figures N denotes the number of degrees of freedom in the corresponding space. All methods are similar in behaviour with respect to condition numbers as a function of degrees of freedom

the mesh in Fig. 15c, is a fair bit larger than the unmodified counterparts. Despite this, LRBNO outperforms THB and LRB by a significant amount when it comes to the mass matrix. The conditioning of the stiffness matrix on the other hand grows

approximately linearly with the number of degrees of freedom, and no significant effect of the overload-reduction can be seen.

6 Conclusion

We have addressed differences and similarities of Truncated Hierarchical B-splines (THB) and Locally Refined B-splines (LRB) on similar hierarchical meshes. The overall conclusion is that there are no big differences between the methods with respect to condition numbers of mass and stiffness matrices for the example meshes addressed.

- When THB and LRB are run on identical meshes THB has better condition numbers for the mass matrix except for the most complex example run, the diagonal example in Figs. 15 and 16. The behaviour of the stiffness matrix is very similar for both methods.
- When making a mesh for LRB that has no overloading the condition numbers for the mass matrix of LRB are smaller than those of THB, with condition numbers of stiffness being similar. It should be noted that using meshes for LRB that has no overloading guarantees that the B-splines generated are linearly independent, and that the number of B-splines covering an element is the minimal needed for spanning the polynomial space over the element. For hierarchical meshes of bi-degree less than (4, 4) there is always linear independence in the set of LR B-splines generated. For bi-degree (4, 4) and higher linear dependence can occur in very special configurations when the elements outside two opposing concave corners of a refinement region is covered by the same B-spline from a cruder level. This happens for bi-degree (4,4) when a refinement region is split if just one element from the cruder level is not refined, e.g., the refinement region is locally very narrow.

When trying to represent hierarchical refinements using T-splines as in Fig. 9 there is a region of one directional refinement of length two just outside the boundary of the refinement region. This gives a smoother transition between refinement levels that can also be replicated by LRB. The results in Figs. 13 and 14 show a better behaviour than going directly from one refinement level to the next. Having such an intermediate level of refinement if possible is advantageous. However, in situations such as the diagonal refinement in Fig. 15 this is not possible.

Most often THB is described as based on dyadic sequences of grids determined by scaled lattices over which uniform B-spline spaces are defined. This implies that there is single knot multiplicity along domain boundaries. However, variants of THB are published [4] where open knots are used along the domain boundary. In Sect. 2 we have shown that open knot vectors are preferable, not only with respect to simplified interpolation of boundary conditions, but also to avoid that the condition number of the mass matrix is biased by the boundary B-splines. As we see the same effect for LR B-splines we have a strong recommendation that open knot vectors

are used for locally refined splines, rather than single knot multiplicity at domain boundaries.

Acknowledgments This project has received funding from the The Research Council of Norway under grant agreement No 270922.

References

1. Tor Dokken, Tom Lyche, and Kjell Fredrik Pettersen. “Polynomial Splines over Locally Refined Box-Partitions”. In: *Computer Aided Geometric Design* 30.3 (Mar. 2013), pp. 331–356.
2. David R. Forsey and Richard H. Bartels. “Hierarchical B-Spline Refinement”. In: *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’88. New York, NY, USA: ACM, pp. 205–212.
3. Eduardo M. Garau and Rafael Vázquez. “Algorithms for the Implementation of Adaptive Isogeometric Methods Using Hierarchical B-Splines”. In: *Applied Numerical Mathematics* 123 (Jan. 1, 2018), pp. 58–87.
4. Carlotta Giannelli, Bert Jüttler, Stefan K. Kleiss, Angelos Mantzaflaris, Bernd Simeon, and Jaka Špeh. “Thb-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 299 (2016), pp. 337–365.
5. Carlotta Giannelli, Bert Jüttler, and Hendrik Speleers. “THB-Splines: The Truncated Basis for Hierarchical Splines”. In: *Computer Aided Geometric Design* 29.7 (Oct. 1, 2012), pp. 485–498.
6. Kjetil André Johannessen, Trond Kvamsdal, and Tor Dokken. “Isogeometric Analysis Using LR B-Splines”. In: *Computer Methods in Applied Mechanics and Engineering* 269 (Feb. 2014), pp. 471–514.
7. Kjetil André Johannessen, Filippo Remonato, and Trond Kvamsdal. “On the Similarities and Differences between Classical Hierarchical, Truncated Hierarchical and LR B-Splines”. In: *Computer Methods in Applied Mechanics and Engineering* 291 (July 1, 2015), pp. 64–101.
8. Francesco Patrizi and Tor Dokken. “Linear dependence of bivariate Minimal Support and LR B-Splines over LR-Meshes”. In: (Nov. 12, 2018). arXiv: 1811.04919 [math].
9. Kjell Fredrik Pettersen. *On the Dimension of Multivariate Spline Spaces*. Tech. rep. <http://hdl.handle.net/11250/2432305>. 2013.
10. F. de Prenter, C.V. Verhoosel, G.J. van Zwieten, and E.H. van Brummelen. “Condition number analysis and preconditioning of the finite cell method”. In: *Computer Methods in Applied Mechanics and Engineering* 316(2017). Special Issue on Isogeometric Analysis: Progress and Challenges, pp. 297–327.
11. M.A. Scott, X. Li, T.W. Sederberg, and T.J.R. Hughes. “Local refinement of analysis-suitable T-splines”. In: *Computer Methods in Applied Mechanics and Engineering* 213–216 (2012), pp. 206–222.
12. Thomas W. Sederberg, David L. Cardon, G. Thomas Finnigan, Nicholas S. North, Jianmin Zheng, and Tom Lyche. “T-spline Simplification and Local Refinement”. In: *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 276–283.
13. A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. “A hierarchical approach to adaptive local refinement in isogeometric analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 200.49 (2011), pp. 3554–3567.