U-splines: splines over unstructured meshes

Derek C. Thomas[1], Luke Engvall[1], Steven K. Schmidt[2], Kevin Tew[1], and Michael A. Scott[2]

[1]Coreform LLC, Orem, Utah, USA
[2]Department of Civil and Environmental Engineering, Brigham Young University, Provo, UT, USA

November 7, 2022

U-splines are a novel approach to the construction of a spline basis for representing smooth objects in Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE). A spline is a piecewise-defined function that satisfies continuity constraints between adjacent cells in a mesh. U-splines differ from existing spline constructions, such as Non-Uniform Rational B-splines (NURBS), subdivision surfaces, T-splines, and hierarchical B-splines, in that they can accommodate local variation in cell size, polynomial degree, and smoothness simultaneously over more varied mesh configurations. Mixed cell types (e.g., triangle and quadrilateral cells in the same mesh) and T-junctions are also supported, although the continuity of interfaces with triangle and tetrahedral cells is limited in the present work. The U-spline algorithm introduces a new technique for using local null space solutions to construct basis functions for the global spline null space problem. The U-spline construction is presented for curves, surfaces, and volumes with higher dimensional generalizations possible. A set of requirements are given to ensure that the U-spline basis is positive, forms a partition of unity, is complete, and is locally linearly independent.

Contents

1 Introduction

Computer aided engineering (CAE) can provide feedback regarding the expected behavior of a given part before costly fabrication is undertaken. The predominant simulation technique in current use in CAE is finite element analysis (FEA). In FEA, the simulation of a computer aided design (CAD) is typically preceded by a process known as meshing, in which a faceted approximation of the original CAD model is constructed to satisfy the requirements of the simulation pipeline. Inconsistencies in common industrial CAD representations, such as small gaps between adjacent CAD faces, must be resolved during the meshing process resulting in a final mesh approximation that is referred to as "watertight" or "analysis-suitable".

A faceted mesh is a piecewise-defined function that satisfies continuity constraints between adjacent cells. The function, restricted to the domain of each cell, is a linear polynomial or facet. Across interfaces, or the boundary of adjacent cells, the function is $\mathcal{C}^0$-continuous. Continuity or smoothness refers to the level at which a function shares the same values on either side of an interface. If the function is discontinuous at the interface then the function can have different values at adjacent points in the cells on either side of the interface. A function is said to be value continuous if the function produces the same value at points in each cell that are adjacent across the interface. Other levels of continuity are also possible; the slope and curvature of the function can also be continuous across the interface. Higher levels of continuity require mathematical definitions in terms of derivatives. We say a function has continuity of order ϑ where $\vartheta \geq -1$; we also say a function is $\mathcal{C}^\vartheta$-continuous. $\mathcal{C}^{-1}$ is discontinuous, $\mathcal{C}^0$ is value continuous, etc.

While it is technically correct to call a faceted mesh a spline, the term spline has become synonymous with meshes of higher degree and continuity. We will often use the term smooth spline to disambiguate the term. The most recognized example of a smooth spline is the B-splines or basis splines. Originating with Cox, de Boor, and Mansfield, the basis spline functions are the minimally supported spline functions on a partitioning of an interval. The minimal, compact support of B-splines is significant for both design and simulation.

The simplicity and efficiency of prevailing smooth spline constructions, their superior approximation properties, and numerical robustness have led to the widespread industrial adoption of smooth splines as the foundation of modern CAD tools. Indeed, nonuniform rational B-splines (NURBS), a generalization of B-splines, are foundational to virtually all CAD modeling environments, where they are used primarily for non-analytic curve and surface representation.

Interestingly, the superior behavior of smooth splines, when applied to FEA has long been recognized by analysts [62]. However, most splines, other than simple $\mathcal{C}^0$ splines, were viewed as too expensive or the construction too complex for general-purpose FEA. Efforts were made to improve the geometric definition in FEA through the use of subdivision surfaces [10] and NURBS in FEA [24], among others, but it was not until the introduction of the concept of isogeometric analysis (IGA) [26] that a large-scale effort to exploit the properties of splines in FEA commenced. IGA can be understood simply to be FEA with smooth splines. The potential benefits of this approach have become clear [14, 21, 41], but the watertight meshing of complex CAD geometries with smooth splines remains a significant barrier to broader adoption.

Our opinion is that a smooth spline meshing technology for industrial-scale FEA problems should be capable of smoothly and accurately representing complex CAD models, be compatible with prevailing industrial spline representations such as NURBS and T-splines, support the local modification of cell size (h-refinement), degree (p-refinement), and intercell continuity (ϑ-refinement), naturally generalize to higher dimensions, and provide a locally-supported, complete, positive basis for the underlying spline space that forms a partition of unity and is (locally) linearly independent. The U-spline technology satisfies all these technical objectives.

1.1 Previous work

The need for advanced smooth spline surface meshes in CAD and, in particular, animation and graphics applications, led to the development of both subdivision surfaces [46] and later T-splines [55]. A significant benefit of the T-spline construction is its compatibility with NURBS representations. Additional developments to follow on the advances of subdivision surfaces and T-splines include PHT-splines [16] and polynomial splines over T-meshes [34]. In these works, the continuity of the splines is restricted to be less than half of the polynomial degrees on adjacent cells. The importance of handling singular or extraordinary vertices smoothly has long been recognized and many approaches have been proposed [49, 44, 63].

The potential benefits of smooth unstructured spline meshes was recognized soon after the advent of

FEA [4]. Despite these early efforts, the majority of finite element research was carried out on $\mathcal{C}^0$ constructions and so FEA came to be associated primarily with $\mathcal{C}^0$ basis functions. More recently, subdivision surfaces were applied to shells [10]. Shortly after the original introduction of IGA [26], work commenced on IGA based on T-splines [6]. This was motivated by both the unstructured nature of T-splines and the need for adaptive local refinement. The need for guarantees on analysis properties of the basis led to the introduction of analysis-suitable T-splines (ASTS) [32]. Other efforts to produce refineable splines suitable for use in FEA followed such as locally refined (LR) B-splines [17] and hierarchical B-splines and truncated hierarchical B-splines [22, 52]. Constructions based on geometric rather than parametric continuity have also been explored [23, 30].

There has also been significant work on spline constructions over unstructured meshes within the wider numerical analysis community, although many of these approaches have not been widely adopted in FEA. Classic approaches commonly employed to produce continuity greater than $\mathcal{C}^0$ include Arqyris elements [4], Clough-Tocher elements [3], and Powell-Sabin splines [59, 60] among others. Significant work has been carried out on the dimension of spline spaces for both triangle [2, 31] and T-meshes [33, 34]. Meshes consisting of both squares and triangles with potentially hanging vertices, also called T-junctions, have been considered [51] although only splines of continuity $\mathcal{C}^0$ were explored. The approximation power of splines over T-meshes for splines of reduced continuity greater than $\mathcal{C}^0$ has been established [51]. Several types of simplex splines have been introduced to facilitate the construction of splines in unstructured settings [43]. Several adaptations of simplex splines to Powell-Sabin and other splits have been proposed to allow their use on unstructured meshes [11, 38, 61]. Additional methods combine the solution of continuity constraints together with the solution of the governing PDEs [5, 25, 50]. Splines based on both triangles [28, 45, 69] and tetrahedra [68] have been employed in IGA.

Practical constructions of mixed degree or multi-degree smooth splines are currently limited to univariate constructions and their tensor products and consequently have not seen extensive use in FEA although basic constructions are used in $\mathcal{C}^0$ *hp*-adaptive methods [15]. In CAGD, univariate mixed degree or multi-degree splines have been proposed [56, 57, 65]. We should mention that multivariate multi-degree splines with higher continuity have been constructed on triangulations without a basis for the purpose of numerical analysis [25].

While B-splines, due to their tensor product structure, naturally generalize to arbitrary dimensions, it remains a challenge to generalize less structured splines to higher dimensions. Initial efforts have been made to parameterize an irregular volume with a collection of one or more tensor product or swept volumetric B-splines or NURBS (sometimes called a multi-patch construction) [39, 1, 71, 70, 67]. T-splines have been primarily used in two-dimensional surface applications, but modest efforts to expand T-splines to the volumetric regime have been made, building on top of the multi-patch approach used with B-splines and NURBS [18, 19, 72, 66, 37, 36, 42]. The volumetric constructions have not yet seen widespread industrial adoption.

Each of these prior technologies have provided important advances and served to demonstrate the power and utility of splines in a wide array of applications, including FEA. However, each method also has known technical limitations in the level of smoothness permitted, maximum dimension, the placement of local refinement features, the polynomial degree supported, or the mathematical quality of the resulting basis functions. The U-spline algorithm presented here represents a fundamentally different method for understanding and constructing splines from any previous work and overcomes many of these limitations.

1.2 Current work and key contributions

A spline space can be constructed directly from a mesh and assigned smoothness constraints, usually through the construction and analysis of an associated *global* smoothness constraint matrix. Various approaches to accomplish this have been proposed but we mention, in particular, the approach based on minimal determining sets [2] as it most closely relates to the U-spline algorithm described in this work. Finding a minimal determining set corresponds to finding a basis of the appropriate size for the spline space. Unfortunately, this approach does not provide any insight into the quality or utility of a basis other than existence and quickly becomes intractable for meshes of even moderate size; accurate determination of even the rank of a large constraint matrix of floating point numbers is a difficult problem.

Of more practical use is an algorithm for the direct construction of a spline basis from a mesh that satisfies the desirable properties of the B-splines: (local) linear independence, minimal (compact) support, and positivity. An important corollary of the minimal support property of the B-splines that is not often

appreciated is the fact that the minimal support property requires that when a single B-spline basis function is expressed in Bernstein form (i.e., written in terms of the Bernstein polynomials), the function is minimally supported in the number of positive Bernstein coefficients. In algebraic terms, this means that the vectors of Bernstein coefficients of the B-spline basis functions form the sparsest positive basis of the nullspace of the smoothness constraint matrix. However, the problem of finding the sparsest basis for the nullspace of a global smoothness constraint matrix, derived from an underlying mesh, is notoriously difficult. In fact, for *general* matrices this type of problem is known as the Nullspace Problem and has been shown to be NP-hard [12, 13].

Using a brute force approach to solve the *global* nullspace problem to determine the members of a spline basis is an intractable problem. To avoid these issues, the U-spline approach leverages a prescribed *admissible* mesh topology and properties of a Bernstein-like basis [40] in order to incrementally, through a series of local operations, construct member functions of the sparsest possible spline basis without directly solving the global nullspace problem. Note that although this approach is generally applicable to bases that satisfy the properties of a Bernstein-like basis, for simplicity, only examples using polynomial Bernstein bases are considered.

Additionally, the U-spline approach seeks to alleviate, if not eliminate, many of those longstanding limitations in prevailing spline representations discussed previously, providing a smooth spline meshing technology for industrial-scale FEA applications. In particular, this work can be seen as a generalization of the key innovations underlying both ASTS and Bézier extraction [8, 54]. Bézier extraction is a method for providing a local Bernstein representation of a non-local smooth spline function, and is used widely in IGA. In the U-spline approach, to guarantee the mathematical properties of U-spline spaces, we also impose restrictions on allowable mesh topologies, as is done in ASTS, but do away with (semi) global data structures, such as knot vectors and T-meshes, and fully adopt a Bernstein representation of spline functions, as is done in Bézier extraction. Leveraging this local Bernstein point-of-view, we achieve greater *locality* in the U-spline algorithm, making it possible to construct well-behaved bases for a much wider range of spline spaces than is encompassed by ASTS, including, for the first time, those that permit local variation in degree.

The key contributions of U-splines can be described as follows:

- An algorithm for (1) solving a series of small and highly localized nullspace problems and (2) finding appropriate combinations of the basis vectors of these localized nullspaces to determine the U-spline basis functions. Importantly, the size of each localized nullspace problem is bounded by the local characteristics of the local basis chosen for each cell, the local mesh topology, and the associated smoothness constraints.

- The algorithm is expressed entirely in terms of integers and requires no floating point operations until after the indices of the nonzero Bernstein coefficients of a U-spline basis function have been determined.

- The need for artificial constructs like global or local knot vectors, control meshes, or T-meshes is eliminated. The only input is a properly specified Bézier mesh that characterizes an associated spline space and the only outputs are linear combinations of Bernstein coefficients that describe U-spline basis functions that span that spline space.

- The algorithm naturally generalizes to higher dimensions.

- Since continuity constraints, restricted to cell interfaces, are the primal building blocks of the U-spline algorithm, local variations in h, p, and ϑ can be processed by the same algorithm as well as T-junctions, extraordinary vertices and triangles. The introduction of local variation in degree in a smooth spline setting is a particularly important and unique U-spline innovation.

- The only requirement on the local basis assigned to each cell in the Bézier mesh is that it must be Bernstein-like [40]. The key property is that it must have well-ordered derivatives at cell interfaces. This means that a mixture of standard polynomial Bernstein bases over quadrilateral and triangular cells can be used in addition to more exotic Bernstein-like bases based on exponential, trigonometric, and other special functions. In this work, we restrict our focus to polynomial Bernstein bases.

- A simple definition of admissibility is given that characterizes a Bézier mesh topology, degree, and smoothness and ensures that the U-spline algorithm, applied to these meshes, produces U-spline basis functions that are *locally* linearly independent (thus forming a basis for the spline space), positive,

form a partition of unity, and are complete up through a specified polynomial degree. To measure completeness in the presence of local variation in degree we require that U-splines satisfy both a local and global completeness measure, both of which are fully characterized by the Bézier mesh.

- The satisfaction of local linear independence is the *pace-setting* property of U-splines, in that it controls to the greatest extent, the allowable Bézier mesh topologies. We should note that by relaxing this requirement to only *global* linear independence, the class of allowable Bézier mesh topologies that can be processed by the U-spline algorithm is greatly expanded. We have successfully constructed U-spline bases in this more general setting but postpone a thorough investigation of those more general U-spline spaces to a future work, as we have found local linear independence to be particularly beneficial to applications of the technology we are interested in.

- As far as the authors are aware, the generality of U-splines spaces, in particular local variation in degree, is beyond the application of the mathematical analysis tools commonly used to rigorously characterize spline spaces. To compensate for this, we instead validate our mathematical claims numerically through a rigorous regime of randomly generated examples and postpone a theoretical exploration of these claims to a future work. We anticipate that this work will spur additional research in the theoretical characterization of the spline spaces generated by the U-spline algorithm.

- When the input Bézier mesh coincides with single- or multi-patch NURBS or analysis-suitable T-splines the U-spline algorithm produces those spline spaces and associated bases with pointwise exactness.

In one dimension, we consider U-splines of any degree and any continuity up to the degree. In higher dimensions, we will only consider U-splines up to degree three with a maximal continuity of $\mathcal{C}^2$, and supersmooth[1] interfaces up to $\mathcal{C}^3$.

We recognize that the name U-spline was originally used to refer to the definition of splines over unordered knot sequences [47]. Because the need for unstructured splines is significant and the application of splines over unordered knot sequences has not yet achieved widespread use, we instead use the U-spline designation for our splines over unstructured meshes.

1.3 Organization of paper

We begin by introducing the Bernstein polynomials in section 2, their use for defining polynomial spaces over each element of a Bézier mesh in section 3, and the representation of a piecewise function in Bernstein form over a Bézier mesh in section 4. Next, we describe the set of continuity constraints associated with a Bézier mesh in section 5, and in section 6 we discuss the associated nullspace problem which must be solved to represent a spline over the Bézier mesh that conforms to the constraints. In section 7 we describe a few tools and metrics for grouping and measuring distances between Bernstein functions on a Bézier mesh, in preparation for describing the construction of local basis vectors on a Bézier mesh in section 8. An *admissible* Bézier mesh, also called a U-spline mesh, is described in section 9, including a set of simple admissibility conditions, which place constraints on the layout of cells, the degree, and the smoothness of interfaces in the Bézier mesh which guarantee certain mathematical properties of the corresponding U-spline space. Section 10 lays out the algorithm for constructing a U-spline basis, and section 11 describes the mathematical properties of this basis and the corresponding U-spline space. A few notable examples of U-splines are highlighted in section 12.

For a more basic introduction to the essential concepts behind U-splines, we recommend the reader work through the exercieses in appendix A, which introduce some of the more technical ideas through simple examples and practice problems, and their accompanying solutions. The appendix also provides additional detail on the topics of continuity constraints in appendix B, k-cell basis vector construction in appendix C, and a concept called a *ribbon* used in defining the admissibility conditions on a U-spline mesh in appendix D. Finally, appendix E lists Bézier extraction coefficient values for a select collection of U-spline basis function examples.

[1] The term *supersmooth* is defined in section 3.

2 The Bernstein polynomials

A fundamental U-spline building block is the Bernstein polynomial basis [20]. A univariate Bernstein polynomial $B_i^p : \overline{\Omega} \to \mathbb{R}$, $i = 0, \ldots, p$, is defined over a parent domain $\overline{\Omega} = [0, 1]$ as

$$B_i^p(\xi) = \binom{p}{i} \xi^i (1-\xi)^{p-i} \tag{1}$$

where p is the polynomial degree and $\xi \in \overline{\Omega}$ is the parent coordinate. We denote the space spanned by the Bernstein polynomial basis by $\mathcal{B}$ and call it the Bernstein space. The Bernstein space is complete through polynomial degree $|p|$. In other words, all polynomials up through degree p are contained in $\mathcal{B}$. The Bernstein polynomials possess many additional desirable properties such as pointwise nonnegativity and partition of unity [20].

Note that we have borrowed the idea of a "parent" domain from FEA, where it is used to standardize and simplify the evaluation of basis functions [27]. The connection of the parent domain to the parametric domain of U-spline basis functions is described in section 3.2.

2.1 Ordering of derivatives

Of particular importance to the U-spline construction algorithms described later is the natural ordering exhibited by derivatives of the Bernstein polynomials. Specifically, we say that a function $f : \mathbb{R} \to \mathbb{R}$ vanishes n times at a real value a if $f^{(i)}(a) = 0$ for all $i \in [0, n)$. Consider then that the nth derivative of the Bernstein polynomial $B_i^p(\xi)$ is given by

$$\left.\frac{d^n B_i^p(\xi)}{d\xi^n}\right|_{\xi=0} = \frac{p!}{(p-n)!}\binom{n}{i}(-1)^{n-i} \tag{2}$$

$$\left.\frac{d^n B_i^p(\xi)}{d\xi^n}\right|_{\xi=1} = \frac{p!}{(p-n)!}\binom{n}{p-i}(-1)^{i-p}. \tag{3}$$

From these equations, we see that $\frac{d^n B_i^p(\xi)}{d\xi^n}$ vanishes i times at $\xi = 0$ and $p - i$ times at $\xi = 1$. For example, the value and derivatives of $B_2^3(\xi)$ vanish at $\xi = 0$ for $n = 0, 1$ since $i = 2$. This property can be observed in fig. 1 where we plot the Bernstein polynomials and their derivatives. Note that we have elected to plot normalized functions, found via $\hat{B}_i^{p^{(n)}}(\xi) = B_i^{p^{(n)}}(\xi) / \max_{\xi \in \overline{\Omega}} \left| B_i^{p^{(n)}}(\xi) \right|$ and $f^{(n)} = \frac{d^n f}{d\xi^n}$, so that all functions may fit comfortably on the same axis. In each plot, the Bernstein polynomials are shown by the thick solid line. Thin solid lines correspond to first derivatives, dashed lines to second derivatives, and dotted lines to third derivatives.

2.2 Degree elevation

The order of the first nonzero derivative of a Bernstein basis function B_i^p evaluated on the left boundary of the parent domain is given by i and by $p - i$ on the right boundary. Consequently, if a single Bernstein basis function B_i^p is to be represented in terms of a degree-elevated Bernstein basis B_j^q, $q > p$, the nonzero derivatives of both bases must match on the boundaries of the domain.

This requirement places bounds on the indices of the degree elevated Bernstein basis functions that are required to represent B_i^p. If the two sets of basis functions are ordered from left to right then $i \leq j$ and $p - i \leq q - j$. These requirements can be combined to obtain a range of valid values for j:

$$i \leq j \leq q - p + i. \tag{4}$$

2.3 Multivariate Bernstein polynomials

In a d-dimensional multivariate setting, Bernstein polynomials are commonly defined over boxes (e.g., quadrilaterals and hexahedra) and simplicial parent domains (e.g., triangles and tetrahedra). The multivariate Bernstein basis functions presented here possess similar derivative ordering properties to the univariate basis described previously. To accommodate this d-dimensional extension, we introduce a dimensional index $k \in \{0, \ldots, d\}$.

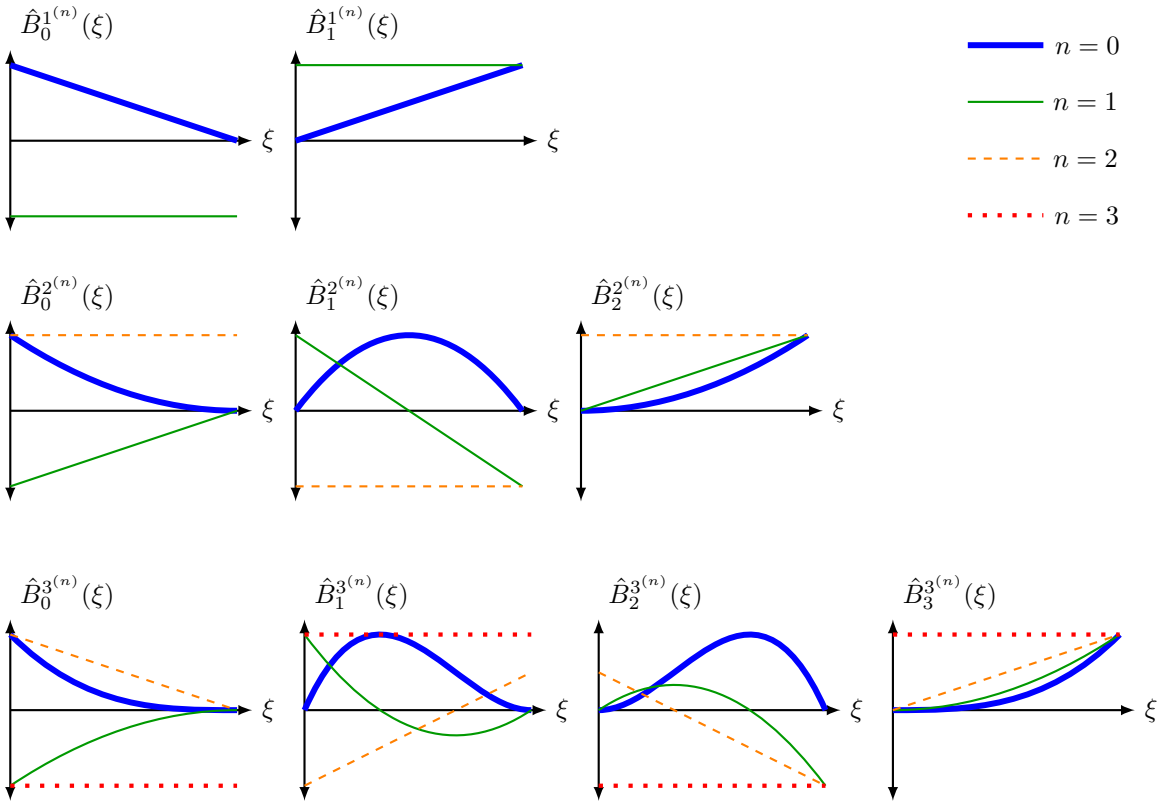


Figure 1: The Bernstein polynomials of degree 1, 2, and 3 and their normalized derivatives, evaluated on $\xi \in \overline{\Omega}$.

2.3.1 Box

A multivariate Bernstein polynomial $B_{\mathsf{i}}^{\mathsf{p}} : \overline{\Omega} \to \mathbb{R}$ is defined over the d-dimensional hypercube or box $\overline{\Omega} = \bigotimes_{k=0}^{d-1}[0,1]$ as the tensor product of univariate Bernstein polynomials

$$B_{\mathsf{i}}^{\mathsf{p}}(\boldsymbol{\xi}) = \prod_{k=0}^{d-1} B_{i_k}^{p_k}(\xi_k) \tag{5}$$

where $\mathsf{i} = (i_0, \ldots, i_{d-1})$ and $\mathsf{p} = (p_0, \ldots, p_{d-1})$ are tuples with i_k and p_k representing the univariate Bernstein basis function index and degree in dimensional direction k, respectively, and the parent coordinate $\boldsymbol{\xi} = [\xi_0, \ldots, \xi_{d-1}] \in \overline{\Omega}$.

2.3.2 Simplicial

A multivariate Bernstein polynomial $B_{\mathsf{i}}^{p} : \overline{\Omega} \to \mathbb{R}$ is defined over the convex hull of the d-dimensional unit simplex $\overline{\Omega} = \left\{\boldsymbol{\xi} = [\xi_0, \ldots, \xi_d] \in \mathbb{R}^{d+1} : \sum_{k=0}^{d} \xi_k = 1 \text{ and } \xi_k \geq 0 \text{ for } k = 0, ..., d\right\}$ as

$$B_{\mathsf{i}}^{p}(\boldsymbol{\xi}) = p! \prod_{k=0}^{d} \frac{\xi_k^{i_k}}{i_k!} \tag{6}$$

where $\mathsf{i} = \left\{i_k : 0 \leq k \leq d, \sum_{k=0}^{d} i_k = p\right\}$ is an index tuple, p is the polynomial degree, and the parent coordinate $\boldsymbol{\xi} \in \overline{\Omega}$ is commonly called a barycentric coordinate. For each boundary of the simplex, the nonzero entries in the basis are precisely the basis for the simplex of dimension $d-1$. Observe that the standard univariate Bernstein basis is merely a special case of the multivariate simplicial form.

2.3.3 Tensor-product hybrid

A multivariate Bernstein polynomial $B^{\mathsf{p}}_{\mathsf{i}} : \overline{\Omega} \to \mathbb{R}$ is defined over the domain $\overline{\Omega}$ constructed by taking the tensor product of an n-dimensional simplex with Bernstein basis $B^{p'}_{\mathsf{j}}(\boldsymbol{\xi})$ and a $(d-n)$-dimensional simplex with Bernstein basis $B^{p''}_{\mathsf{k}}(\boldsymbol{\xi})$, $0 < n < d$,

$$\overline{\Omega} = \left\{ \boldsymbol{\xi} = [\xi_0, \ldots, \xi_n, \xi_{n+1}, \ldots, \xi_{d+1}] \in \mathbb{R}^{d+2} : \sum_{k=0}^{n} \xi_k = 1, \sum_{k=n+1}^{d+1} \xi_k = 1, \text{ and } \xi_k \geq 0 \text{ for } k = 0, ..., d+1 \right\}$$

as

$$B^{\mathsf{p}}_{\mathsf{i}}(\boldsymbol{\xi}) = B^{p'}_{\mathsf{j}}(\boldsymbol{\xi}) \otimes B^{p''}_{\mathsf{k}}(\boldsymbol{\xi}) \tag{7}$$

where $\mathsf{i} = \left\{ i_k : 0 \leq k \leq d+1, \sum_{k=0}^{n} i_k = p', \sum_{k=n+1}^{d+1} i_k = p'', \right\}$ is an index tuple, $\mathsf{p} = (p', p'')$ is the polynomial degree, and the parent coordinate $\boldsymbol{\xi} \in \overline{\Omega}$ is the Cartesian product of the two lower-dimensional barycentric coordinates. Similar constructions exist for boxes, and tensor products of simplices and boxes.

2.4 Bernstein-like bases

Although the focus is on the polynomial Bernstein basis in this work, this is not a necessary requirement. It has been shown [40] that quasi-extended Chebyshev (QEC) spaces possess a Bernstein-like basis with the following property: Let $\mathcal{E}$ be an $(n+1)$-dimensional QEC-space on the bounded closed interval $[a, b]$. Then, $\mathcal{E}$ possesses a quasi-Bernstein-like basis relative to (a, b), that is, a basis $B_i : \overline{\Omega} \to \mathbb{R}$, $i = 0, 1, ..., n$ such that:

- $B_0(a) \neq 0$, and B_0 vanishes n times at b; $B_n(b) \neq 0$, and B_n vanishes n times at a;
- for $1 \leq i \leq n-1$, B_i vanishes exactly i times at a and exactly $(n-i)$ times at b.
- for $0 \leq i \leq n$, B_i is positive on (a, b).

This property is the key requirement for the U-spline definition and construction and so U-splines can be constructed from meshes with a QEC space assigned to each cell.

3 The Bézier mesh

As mentioned in the previous section, a key property of U-splines is the ability to construct a spline basis on an unstructured Bézier mesh. A Bézier mesh $\mathbf{B}$ is defined by

1. A polyhedral mesh topology,
2. A local parameterization on each cell in the mesh,
3. A Bernstein space $\mathcal{B}$ assigned to each cell in the mesh,
4. A minimum level of continuity specified on each interface between cells.

In this section, the notation used throughout the remainder of this paper to describe a Bézier mesh is introduced.

3.1 Topology

Formally, the Bézier mesh is a tiling of a d-dimensional manifold with box and simplex k-cells, $0 \leq k \leq d$, where k is the dimension of the cell. More precisely, the Bézier mesh $\mathbf{B}$ is a cell complex where:

- Each k-dimensional cell c is a closed subspace of $\mathbb{R}^d$,
- Any lower-dimensional cell $\mathsf{a} \subset \mathsf{c}$ is also in $\mathbf{B}$,
- The non-empty intersection of any two cells a and b in $\mathbf{B}$ is a lower-dimensional cell contained in both.

We have the following correspondence to common mesh entities:

	$d=1$	$d=2$	$d=3$
d-cell	Edge	Face	Volume
$(d-1)$-cell	Vertex	Edge	Face
$(d-2)$-cell	-	Vertex	Edge
$(d-3)$-cell	-	-	Vertex

When a dimension-agnostic description is appropriate for a concept, we employ the generic terminology d-cell, $(d-1)$-cell, etc. For simplicity, we occasionally refer to d-cells, or elements, by E, $(d-1)$-cells, or interfaces, by I, $(d-2)$-cells by w, 2-cells, or faces, by f, 1-cells, or edges, by e, and 0-cells, or vertices, by v. We denote the set of cells of dimension k in the mesh $\mathbf{B}$ by $\mathsf{C}^k(\mathbf{B})$.

3.1.1 Adjacencies

It is useful to define a set of k-cell adjacency operators. Given a cell c and a d-dimensional mesh $\mathbf{B}$, the set of k-cells adjacent to c is denoted by

$$\mathsf{ADJ}^k(\mathsf{c}) = \left\{\mathsf{a} \in \mathsf{C}^k(\mathbf{B}) : \mathsf{a} \cap \mathsf{c} \neq \varnothing\right\} \tag{8}$$

and $\left|\mathsf{ADJ}^k(\mathsf{c})\right|$ denotes the number of k-cells adjacent to c. Chained adjacency sets are written as

$$\mathsf{ADJ}^{k_1} \circ \mathsf{ADJ}^{k_0}(\mathsf{c}) = \bigcup_{\mathsf{a} \in \mathsf{ADJ}^{k_0}(\mathsf{c})} \mathsf{ADJ}^{k_1}(\mathsf{a}). \tag{9}$$

The boundary of a k-cell c is denoted by

$$\partial \mathsf{c} = \mathsf{ADJ}^{k-1}(\mathsf{c}). \tag{10}$$

Given a k-cell a^k, $k < d$, and an adjacent $(k+1)$-cell $\mathsf{b}^{k+1} \in \mathsf{ADJ}^{k+1}(\mathsf{a}^k)$, the set $\mathbf{PC}(\mathsf{a}^k, \mathsf{b}^{k+1})$ contains all $(k+1)$-cells which are both adjacent to the given k-cell a^k and perpendicular to the given $(k+1)$-cell $\mathsf{b}^{k+1} \in \mathsf{ADJ}^{k+1}(\mathsf{a}^k)$ and is defined as

$$\mathbf{PC}(\mathsf{a}^k, \mathsf{b}^{k+1}) = \left(\mathsf{ADJ}^{k+1}(\mathsf{a}^k) \cap \mathsf{ADJ}^{k+1} \circ \mathsf{ADJ}^{d}(\mathsf{b}^{k+1})\right) \setminus \mathsf{b}^{k+1}\,. \tag{11}$$

3.1.2 k-cell types

A k-cell c^k is an interior cell if every interface adjacent to c^k is adjacent to two elements. Otherwise, it is a *boundary* cell. We say that c^k is *regular* if it is possible to imbed the adjacent elements into a regular grid. More specifically, in a d-dimensional mesh a vertex v is *regular* if all elements in $\mathsf{ADJ}^d(\mathsf{v})$ are box-type and $\left|\mathsf{ADJ}^d(\mathsf{v})\right| = 2^{(\left|\mathsf{ADJ}^1(\mathsf{v})\right| - d)}$, and a $(d-2)$-cell w is *regular* if all elements in $\mathsf{ADJ}^d(\mathsf{w})$ are box-type and $\left|\mathsf{ADJ}^d(\mathsf{w})\right| = 2^{(\left|\mathsf{ADJ}^{d-1}(\mathsf{w})\right| - 2)}$. Otherwise, it is an *extraordinary* cell. Note that all vertices and $(d-2)$-cells adjacent to simplex elements are considered to be extraordinary for this work. On d-dimensional meshes, an extraordinary $(d-2)$-cell w is said to be valence-m if $\left|\mathsf{ADJ}^{d-1}(\mathsf{w})\right| = m$.

3.2 Cell domains and parameterization

Building splines over a Bézier mesh requires that a domain and right-handed coordinate system be specified over each cell. These coordinate systems may change from cell to cell to accommodate extraordinary cells or cells of different type, such as between box-like and simplicial cells.

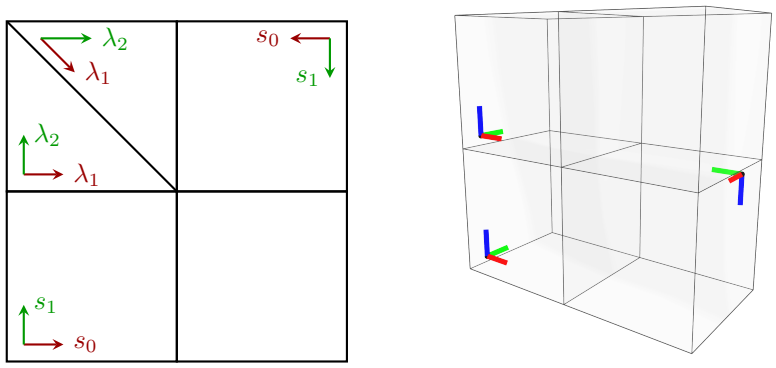
Each cell c is assigned a *parent* domain $\overline{\Omega}$ and a right-handed coordinate system $\boldsymbol{\xi} \in \overline{\Omega}$ or, when referencing a particular cell c, $\overline{\Omega}^{\mathsf{c}}$ and $\boldsymbol{\xi}^{\mathsf{c}}$, respectively. We define $\overline{\Omega}^{\mathbf{B}} = \bigcup_{\mathsf{c} \in \mathbf{B}} \overline{\Omega}^{\mathsf{c}}$. For box cells, $\overline{\Omega}$ is assumed to be a unit hypercube with a cartesian coordinate system and, for simplicial cells, $\overline{\Omega}$ is taken to be the convex hull of a unit simplex with barycentric coordinates. The parent domain is defined in this way to simplify or standardize the implementation and evaluation of a Bernstein basis.

Many situations require the use of nonuniform cell dimensions. The canonical example is nonuniform B-splines; the knot vector that defines the basis possesses intervals of varying length. Although we do not use a knot vector in the definition of U-splines, we do require the flexibility of nonuniform parametric cell dimension. Each cell c is assigned a *parametric* domain $\hat{\Omega}$ and a right-handed coordinate system $\boldsymbol{s} \in \hat{\Omega}$ or, when referencing a particular cell c, $\hat{\Omega}^{\mathsf{c}}$ and $\boldsymbol{s}^{\mathsf{c}}$, respectively. We define $\hat{\Omega}^{\mathbf{B}} = \bigcup_{\mathsf{c}\in\mathbf{B}} \hat{\Omega}^{\mathsf{c}}$. The parametric domain of a box cell is assumed to be a hyperrectangle with Cartesian coordinates. It is a potentially nonuniform scaling of a unit hypercube (the length in each direction may be different). The parametric domain of a simplicial cell will be the convex hull described by the simplex whose edges have been assigned arbitrary lengths but which are usually set to be equal to the parametric size of adjacent box cells. Barycentric coordinates are again assumed for the simplicial parametric domain. Although not discussed further in this work, the relative parametric sizes of adjacent cells must be chosen carefully so as to admit a well-defined smooth spline basis [9].

In two dimensions, we will often refer to the parametric coordinate $\boldsymbol{s}$ as $[s_0, s_1]$ on a quadrilateral face and $[\lambda_0, \lambda_1, \lambda_2]$ on a triangular face, as shown in fig. 2a. In three dimensions, we will refer to the parametric coordinate $\boldsymbol{s}$ as $[s_0, s_1, s_2]$ on a hexahedral volume, as shown in fig. 2b. The operator $\boldsymbol{s}^{\perp}_{\mathsf{E}}(\mathsf{I})$ returns the set of parametric coordinates on element E that are normal to the interface $\mathsf{I} \in \mathsf{ADJ}^{d-1}(\mathsf{E})$ and the operator $\boldsymbol{s}^{\parallel}_{\mathsf{E}}(\mathsf{c}^k)$ returns the set of parametric coordinates on element E that are parallel to an adjacent k-cell $\mathsf{c}^k \in \mathsf{ADJ}^k(\mathsf{E})$. Similarly, $\boldsymbol{s}^{\parallel}_{\Omega}(\mathsf{c}^k)$ returns the set of coordinates in submesh domain Ω (section 7.2) that are parallel to c^k.

If required, the orientation of a cell's parametric coordinate system will be specified by a small axis located at the origin of the coordinate system. If the small axis is omitted, a cell on a two-dimensional mesh is assumed to be oriented with the page, with the origin of the parametric coordinate system at the bottom-left corner (see fig. 2a). Note that on triangles, typically only the barycentric axes associated with λ_1 and λ_2 are shown, since $\lambda_0 = 1 - \lambda_1 - \lambda_2$. On volumetric meshes, as shown in fig. 2b, a similar small axis may be used to specify the orientation of the parameterization. The parametric coordinate systems on a volumetric Bézier mesh can be rotated relative to each other as shown on the bottom two cells in fig. 2b.

For every cell c, we assume there exists a *linear* transformation between the parent and parametric domains wherein the parametric domain can be described in terms of the parent coordinates $\boldsymbol{\xi}^{\mathsf{c}} \in \overline{\Omega}^{\mathsf{c}}$. We denote this linear mapping by $\boldsymbol{\phi}^{\mathsf{c}} : \overline{\Omega}^{\mathsf{c}} \to \hat{\Omega}^{\mathsf{c}}$ where $\boldsymbol{s}^{\mathsf{c}} = \boldsymbol{\phi}^{\mathsf{c}}(\boldsymbol{\xi}^{\mathsf{c}}) = \boldsymbol{A}^{\mathsf{c}}\boldsymbol{\xi}^{\mathsf{c}}$. Note that for box cells $\boldsymbol{\phi}^{\mathsf{c}}$ is a simple scaling, i.e., the matrix $\boldsymbol{A}^{\mathsf{c}}$ is diagonal.



(a) (b)

Figure 2: Examples of parametric coordinate systems specified on quadrilaterals and triangles in two dimensions (left) and on hexahedra in three dimensions (right) using small axes.

3.3 Cell space and degree

A box or simplicial Bernstein space $\mathcal{B}$ is assigned to each cell c and is denoted by $\mathcal{B}^{\mathsf{c}}$. We denote the total degree of polynomial completeness of the Bernstein space on c by $|p^{\mathsf{c}}|$.

The following derived quantities will be used extensively in the description of the U-spline algorithm. Let $p^{\perp}_{\mathsf{I}}(\mathsf{E})$ denote the degree on a d-cell E in the direction perpendicular to the adjacent interface I and $p^{\parallel}_{\mathsf{e}}(\mathsf{E})$

denote the degree on a d-cell E in the direction parallel to the adjacent edge e. Let $\mathsf{p}^{\parallel}_{\mathsf{c}^k}(\mathsf{E}), 0 \leq k \leq d$ denote the k-dimensional tuple p containing the degrees on E in the directions parallel to the cell c^k. We can then define the useful operators $p^{\perp}_{\max}(\mathsf{I})$, $p^{\perp}_{\min}(\mathsf{I})$, $p^{\parallel}_{\max}(\mathsf{e})$, and $p^{\parallel}_{\min}(\mathsf{e})$ as

$$p^{\perp}_{\max}(\mathsf{I}) = \max_{\mathsf{E} \in \mathsf{ADJ}^d(\mathsf{I})} p^{\perp}_{\mathsf{I}}(\mathsf{E}), \tag{12}$$

$$p^{\perp}_{\min}(\mathsf{I}) = \min_{\mathsf{E} \in \mathsf{ADJ}^d(\mathsf{I})} p^{\perp}_{\mathsf{I}}(\mathsf{E}), \tag{13}$$

$$p^{\parallel}_{\max}(\mathsf{e}) = \max_{\mathsf{E} \in \mathsf{ADJ}^d(\mathsf{e})} p^{\parallel}_{\mathsf{e}}(\mathsf{E}), \tag{14}$$

$$p^{\parallel}_{\min}(\mathsf{e}) = \min_{\mathsf{E} \in \mathsf{ADJ}^d(\mathsf{e})} p^{\parallel}_{\mathsf{e}}(\mathsf{E}). \tag{15}$$

To measure the the minimum or maximum degree in a direction parallel to an interface I and perpendicular to a $(d-2)$-cell w adjacent to the interface we define $p^{\parallel,\perp}_{\min}(\mathsf{I},\mathsf{w})$ and $p^{\parallel,\perp}_{\max}(\mathsf{I},\mathsf{w})$ as

$$p^{\parallel,\perp}_{\min}(\mathsf{I},\mathsf{w}) = \min_{\mathsf{E} \in \mathsf{ADJ}^d(\mathsf{I})} p^{\perp}_{\mathsf{I}'}(\mathsf{E}), \mathsf{I}' \in \left(\mathsf{ADJ}^{d-1}(\mathsf{E}) \cap \mathsf{ADJ}^{d-1}(\mathsf{w}) \setminus \mathsf{I}\right), \tag{16}$$

$$p^{\parallel,\perp}_{\max}(\mathsf{I},\mathsf{w}) = \max_{\mathsf{E} \in \mathsf{ADJ}^d(\mathsf{I})} p^{\perp}_{\mathsf{I}'}(\mathsf{E}), \mathsf{I}' \in \left(\mathsf{ADJ}^{d-1}(\mathsf{E}) \cap \mathsf{ADJ}^{d-1}(\mathsf{w}) \setminus \mathsf{I}\right). \tag{17}$$

3.4 Interface continuity

Given a d-dimensional mesh $\mathbf{B}$, each interface I is assigned a required minimum continuity ϑ. We denote the continuity ϑ assigned to an interface I by ϑ^{I}. Note that for certain mesh configurations, the U-spline basis may be smoother than the specified conditions on the interfaces. We say that an interface I has reduced continuity with respect to an adjacent d-cell E if $\vartheta^{\mathsf{I}} < p^{\perp}_{\mathsf{I}}(\mathsf{E}) - 1$ where $p^{\perp}_{\mathsf{I}}(\mathsf{E})$ is the degree on a d-cell E in the direction perpendicular to the adjacent interface I. We say that an interface is creased if it has been assigned $\mathcal{C}^0$ or $\mathcal{C}^{-1}$ continuity and a $(d-2)$-cell is creased if all adjacent interfaces are creased. The operator $\vartheta^{\perp}_{\max}(\mathsf{a}^k, \mathsf{b}^{k+1})$ returns the maximum continuity of all the interfaces that are both adjacent to the given k-cell a^k and perpendicular to the given $(k+1)$-cell $\mathsf{b}^{k+1} \in \mathsf{ADJ}^{k+1}(\mathsf{a}^k)$, and is defined as

$$\vartheta^{\perp}_{\max}(\mathsf{a}^k, \mathsf{b}^{k+1}) = \max_{\mathsf{I} \in \mathbf{PI}(\mathsf{a}^k, \mathsf{b}^{k+1})} \vartheta^{\mathsf{I}} \tag{18}$$

where

$$\mathbf{PI}(\mathsf{a}^k, \mathsf{b}^{k+1}) = \left\{ \mathsf{I} \in \mathsf{ADJ}^{d-1}(\mathsf{a}^k) \cap \mathsf{ADJ}^{d-1} \circ \mathsf{ADJ}^{d}(\mathsf{b}^{k+1}) \,:\, \forall \mathsf{E} \in \mathsf{ADJ}^d(\mathsf{b}^{k+1}), \boldsymbol{s}^{\perp}_{\mathsf{E}}(\mathsf{I}) \subseteq \boldsymbol{s}^{\parallel}_{\mathsf{E}}(\mathsf{b}^{k+1}) \right\}. \tag{19}$$

The continuity of interfaces on two-dimensional figures will be indicated by an accompanying legend or a description in the caption. The continuity of interfaces on volumetric meshes will be indicated by the description in the caption and will follow the pattern indicated in fig. 3. The most common continuity in the mesh and boundary interfaces are often left colorless for greater clarity.

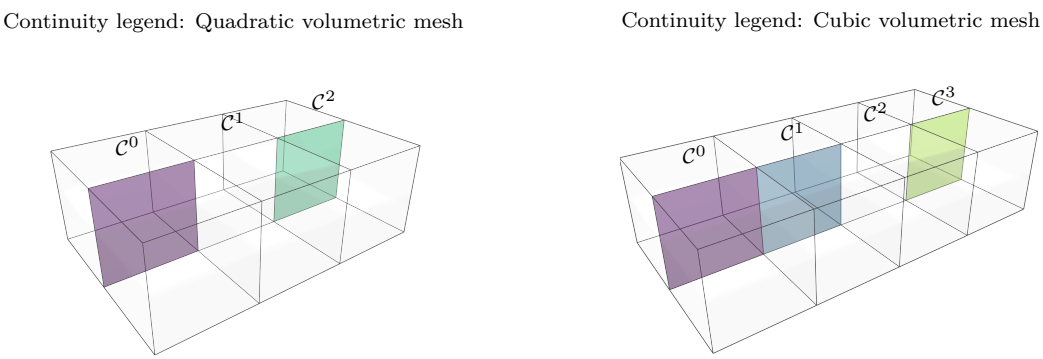


Figure 3: The continuity of interfaces on volumetric Bézier meshes are indicated by the color of a semi-transparent surface drawn on the interface.

3.4.1 Supersmooth interfaces

When setting the smoothness of an interface we normally require that

$$\vartheta^{\mathsf{I}} < p^{\perp}_{\min}(\mathsf{I}) \tag{20}$$

and say that ϑ^{I} is *maximally* smooth if

$$\vartheta^{\mathsf{I}} = p^{\perp}_{\min}(\mathsf{I}) - 1. \tag{21}$$

Additional options are possible with U-splines. We say an interface I is *supersmooth* if

$$\vartheta^{\mathsf{I}} = p^{\perp}_{\mathsf{I}}(\mathsf{E}) \tag{22}$$

for some element E adjacent to I. If, additionally

$$\vartheta^{\mathsf{I}} = p^{\perp}_{\max}(\mathsf{I}) \tag{23}$$

then the supersmooth interface is in fact $\mathcal{C}^{\infty}$. Note that a supersmooth interface is a generalization of the concept of T-junctions [55].

We do not explore this concept further but rather restrict supersmoothness to the simple case where, given two elements a and b that share interface I, we allow $\vartheta^{\mathsf{I}} = p^{\perp}_{\max}(\mathsf{I})$ and require that $p^{\perp}_{\mathsf{I}}(\mathsf{a}) = p^{\perp}_{\mathsf{I}}(\mathsf{b})$ and $\mathsf{p}^{\parallel}_{\mathsf{I}}(\mathsf{a}) = \mathsf{p}^{\parallel}_{\mathsf{I}}(\mathsf{b})$. Figure 4 shows several examples of supersmooth interfaces. On the left, an example of a two-dimensional Bézier mesh with maximally smooth interfaces (eq. (21)) is shown. The two meshes shown in the center have supersmooth interfaces with differing perpendicular degree (eq. (22)). On the right, a $\mathcal{C}^{\infty}$ supersmooth interface in a configuration which is equivalent to a T-junction (eq. (23)) is shown.

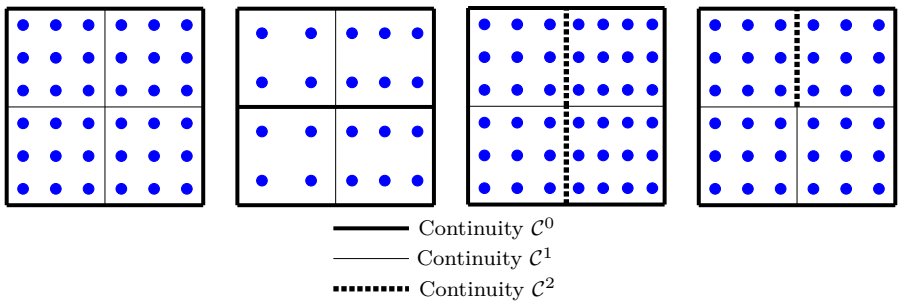


Figure 4: Examples of supersmooth interfaces.

4 Bernstein representations

We now turn our attention to how spline functions are defined over a Bézier mesh. As discussed previously, classical spline technologies, such as NURBS and T-splines, rely on a certain level of global structure to define basis functions. In the case of NURBS, all cells in one direction must use the same polynomial degree and higher-dimensional splines are constructed by global tensor products of univariate NURBS. T-splines also require the specification of global polynomial degree and, while T-splines require less global structure than NURBS, each T-spline basis function is constructed on a local tensor product region. For T-splines, it can be difficult to infer the properties of the underlying spline space by examining the associated T-mesh.

A key practical and conceptual development in the field of IGA was the advent of Bézier extraction as an analysis technology. Bézier extraction allows global spline functions to be evaluated locally on a Bézier cell [8, 54]. More generally, Bézier extraction is a method of providing a *Bernstein representation* of spline functions while maintaining the connection to global spline representation. Bernstein representations are central to representing U-splines, and this section serves to present the necessary notation that will be used throughout the rest of this paper.

4.1 Indexing

The ith Bernstein polynomial on a k-cell c in the mesh forms a unique index, denoted by i^{c}, that specifies both c and the local Bernstein function index i. For simplicity, and when the meaning is unambiguous, we will often just use i to denote both the function index and cell. The set of all indices for all Bernstein polynomials defined over all cells in $\mathbf{B}$ is denoted by $\mathsf{ID}(\mathbf{B})$ and $\mathsf{ID}(\mathsf{c})$ is used to represent the indices for c. We denote the number of indices in any index set by $|\mathsf{ID}|$. The cell associated with a given index is denoted by $\mathrm{cell}(\mathsf{i})$ and the set of cells associated with an index set ID is written as

$$\mathsf{C}(\mathsf{ID}) := \bigcup_{\mathsf{i} \in \mathsf{ID}} \mathrm{cell}(\mathsf{i}). \tag{24}$$

We will often denote a set of index sets by $\mathbf{ID}$.

Figure 5 shows an example of indices on two-dimensional cells. The positioning and quantity of indices indicates the polynomial degree of the Bernstein functions on each cell in each parametric direction. For example, the quadrilateral cell in fig. 5 is quadratic in s_0 and cubic in s_1. On triangular cells, we use the convention to list only indices i_1 and i_2 since on a two-dimensional simplicial Bernstein polynomial, one of the indices is fixed by the choice of polynomial degree and the other two indices: $i_0 = p - i_1 - i_2$, $i_k \in \mathsf{i}$. In figures where the specific index is either apparent from context or not required, we represent indices as small circles or spheres as shown in figs. 5 and 6. Note that we draw these circles inset from the boundary of the cell to avoid confusion between functions on adjacent cells.

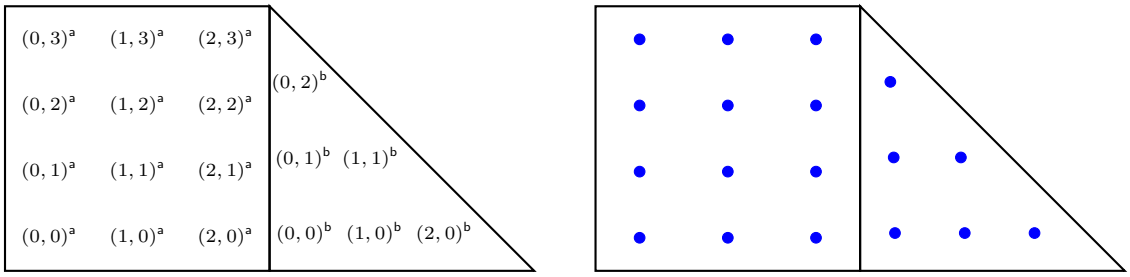


Figure 5: The Bernstein indices (on the left) and corresponding circles (on the right) for a two-dimensional mesh with a quadrilateral and trianglar cell, with degrees $\mathsf{p} = (2, 3)$ and $p = 2$, respectively.

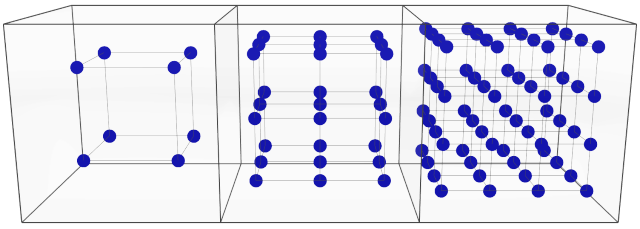


Figure 6: We depict Bernstein indices in volumetric cells as small spheres, the positioning and quantity of which indicate the polynomial degree of the Bernstein functions on each cell in each parametric direction. The cells from left to right have degrees $\mathsf{p} = (1, 1, 1), (2, 2, 2), (3, 3, 3)$, respectively.

4.2 Bernstein form

We say that a piecewise function $f : \hat{\Omega}^{\mathbf{B}} \to \mathbb{R}$ can be written in Bernstein form on the mesh $\mathbf{B}$ if it may be expressed as a linear combination of the Bernstein polynomials on each d-cell E in the mesh. In other words, given a set of Bernstein coefficients

$$\mathbf{c}[\mathbf{B}] = \{c_{\mathsf{i}}\}_{\mathsf{i} \in \mathsf{ID}(\mathbf{B})} \tag{25}$$

the Bernstein form of f is

$$f(\boldsymbol{s}) = \sum_{\mathsf{i} \in \mathsf{ID}(\mathbf{B})} c_\mathsf{i} B_\mathsf{i}(\boldsymbol{\phi}^{-1}(\boldsymbol{s})) \quad \forall \boldsymbol{s} \in \hat{\Omega}^{\mathbf{B}}. \tag{26}$$

We will often use the Bernstein coefficients $\mathbf{c}[\mathbf{B}]$ of f and the function f interchangeably.

The index set of f, or equivalently, $\mathbf{c}[\mathbf{B}]$, is $\mathsf{ID}(\mathbf{B})$ by definition since the domain of f is $\hat{\Omega}^{\mathbf{B}}$. Functions that are nonzero on only a portion of the Bézier mesh $\mathsf{S} \subset \mathbf{B}$ will typically have a large number of zero coefficients and so we adopt a sparse representation that contains only the indices that correspond to nonzero coefficients. The indices associated with the *nonzero* Bernstein coefficients in $\mathbf{c}[\mathbf{B}]$ or f is denoted by

$$\mathsf{ID}(\mathbf{c}[\mathbf{B}]) = \{\mathsf{i} \in \mathsf{ID}(\mathbf{B}) : \mathbf{c}[\mathbf{B}] \ni |c_\mathsf{i}| > 0\}. \tag{27}$$

The nonzero support of f, denoted by $\mathrm{supp}_+(f)$ or $\mathrm{supp}_+(\mathbf{c}[\mathbf{B}])$, is the parametric domain over which the function is nonzero

$$\mathrm{supp}_+(f) = \bigcup_{\mathsf{i} \in \mathsf{ID}(f)} \hat{\Omega}^{\mathsf{c}(\mathsf{i})}. \tag{28}$$

Given a function f in Bernstein form and an index set ID, we define the restriction of f, denoted by $f|_{\mathsf{ID}}$ or, equivalently, $\mathbf{c}[\mathbf{B}]|_{\mathsf{ID}}$, to be the function having the same Bernstein coefficient values as f for all indices in ID and zero for all indices not in ID.

4.3 The trace mapping matrix

Assume that we are given an element E, with associated parametric domain $\hat{\Omega}^{\mathsf{E}}$, and an adjacent (lower-dimensional) interface I, with parametric domain $\hat{\Omega}^{\mathsf{I}} \subset \hat{\Omega}^{\mathsf{E}}$, and a map $\boldsymbol{\phi}_{\mathsf{I}\to\mathsf{E}} : \hat{\Omega}^{\mathsf{I}} \to \hat{\Omega}^{\mathsf{E}}$, and Bernstein bases $B^{\mathsf{E}}_\mathsf{j} : \hat{\Omega}^{\mathsf{E}} \to \mathbb{R}$ and $B^{\mathsf{I}}_\mathsf{i} : \hat{\Omega}^{\mathsf{I}} \to \mathbb{R}$ satisfying $B^{\mathsf{E}}_\mathsf{i} \circ \boldsymbol{\phi}_{\mathsf{I}\to\mathsf{E}} = \sum_{\mathsf{j}\in\mathsf{ID}(\mathsf{I})} c_\mathsf{j} B^{\mathsf{I}}_\mathsf{j}$ (all functions from the element basis can be represented as linear combinations of interface basis functions). Then, the components of the *trace mapping matrix* $\boldsymbol{M}$ are

$$[\boldsymbol{M}]_{\mathsf{ij}} = \left\langle \bar{B}^{\mathsf{I}}_\mathsf{i}, B^{\mathsf{E}}_\mathsf{j} \right\rangle_{\mathsf{I}} = \int_{\hat{\Omega}^{\mathsf{I}}} \bar{B}^{\mathsf{I}}_\mathsf{i}(\boldsymbol{s}) B^{\mathsf{E}}_\mathsf{j}(\boldsymbol{\phi}_{\mathsf{I}\to\mathsf{E}}(\boldsymbol{s}))\, \mathrm{d}\hat{\Omega} \tag{29}$$

where $\bar{B}^{\mathsf{I}}_\mathsf{i}$ is *dual* to $B^{\mathsf{I}}_\mathsf{i}$ in the sense that $\left\langle \bar{B}^{\mathsf{I}}_\mathsf{i}, B^{\mathsf{I}}_\mathsf{j} \right\rangle_{\mathsf{I}} = \delta_{\mathsf{ij}}$ [29].

We use $D^n_{\mathsf{I}\perp} f$ to indicate the nth directional derivative of the function $f : \hat{\Omega}^{\mathsf{c}} \to \mathbb{R}$ with respect to the direction perpendicular to the interface I. the components of the *trace derivative mapping matrix* $\boldsymbol{M}^n$ are

$$[\boldsymbol{M}^n]_{\mathsf{ij}} = \left\langle \bar{B}^{\mathsf{I}}_\mathsf{i}, D^n_{\mathsf{I}\perp} B^{\mathsf{E}}_\mathsf{j} \right\rangle = \int_{\hat{\Omega}^{\mathsf{I}}} \bar{B}^{\mathsf{I}}_\mathsf{i}(\boldsymbol{s}) (D^n_{\mathsf{I}\perp} B^{\mathsf{E}}_\mathsf{j})(\boldsymbol{\phi}_{\mathsf{I}\to\mathsf{E}}(\boldsymbol{s}))\, \mathrm{d}\hat{\Omega}\,. \tag{30}$$

For a given $\mathsf{i} \in \mathsf{ID}(\mathsf{I})$ the set of indices on E that correspond to the nonzero coefficients on the ith row of $\boldsymbol{M}$ is denoted NZ and is defined as

$$\mathsf{NZ}(\boldsymbol{M}, \mathsf{i}) = \left\{ \mathsf{j} \in \mathsf{ID}(\mathsf{E}) \;:\; \left| [\boldsymbol{M}]_{\mathsf{ij}} \right| > 0 \right\}. \tag{31}$$

Note that, in practice, there are efficient approaches for determining the indices of the nonzero coefficients of the trace mapping matrix without computing the matrix terms via integration.

5 Continuity constraints

As mentioned in section 3, a Bézier mesh $\mathbf{B}$ has a prescribed minimum continuity ϑ assigned to each interface I. That is, a piecewise function f defined over $\hat{\Omega}^{\mathbf{B}}$ should have at least $\vartheta + 1$ continuous derivatives across I. Given a function f in Bernstein form, we may state a continuity constraint in terms of the function's Bernstein coefficients, $\mathbf{c}[\mathbf{B}]$. We denote a set of continuity constraint coefficients as $\mathbf{r}$. Then, we may write a homogeneous continuity constraint on a piecewise function in Bernstein form as

$$\sum_{\mathsf{i} \in \mathsf{ID}(\mathbf{r})} r_\mathsf{i} c_\mathsf{i} = 0. \tag{32}$$

An abstract approach to assembling the constraint sets is presented here, which may be applied to determine the constraint equations for meshes of any dimension.

5.1 Constraint sets

The set of continuity constraints associated with an interface I is denoted by $\mathsf{R}(\mathsf{I})$ and the set of all continuity constraints defined over $\mathbf{B}$ is denoted by

$$\mathsf{R}(\mathbf{B}) = \bigcup_{\mathsf{I} \in \mathbf{B}} \mathsf{R}(\mathsf{I}). \tag{33}$$

Given a k-dimensional cell $\mathsf{c}^k \in \mathsf{C}^k(\mathbf{B})$, $0 \leq k < d-1$, the associated set of continuity constraints is the union of all constraint systems associated with interfaces satisfying $\mathsf{I} \cap \mathsf{c}^k \neq \varnothing$:

$$\mathsf{R}(\mathsf{c}^k) = \bigcup_{\mathsf{I} \in \mathsf{ADJ}^{d-1}(\mathsf{c}^k)} \mathsf{R}(\mathsf{I}). \tag{34}$$

Given an index set ID and a continuity constraint, $\mathbf{r}$, the restriction of the constraint to the index set is the constraint that results from keeping only the coefficients associated with indices in the index set. It is denoted by $\mathbf{r}|_{\mathsf{ID}}$. Given an index set ID and a set of continuity constraints R, we denote the restricted set of continuity constraints by

$$\mathsf{R}|_{\mathsf{ID}} = \{\mathbf{r}|_{\mathsf{ID}} : \|\mathbf{r}|_{\mathsf{ID}}\| > 0, \mathbf{r} \in \mathsf{R}\}. \tag{35}$$

In other words, any constraints that are empty after the restriction to the index set has occurred are removed.

5.2 Constraint matrices

Given a set of continuity constraints R, an index set ID, and a function i_{ID} that assigns a unique integer $i \in 0, ..., |\mathsf{ID}|-1$ to every index in ID, we can construct a matrix $\boldsymbol{R}|_{\mathsf{ID}} \in \mathbb{R}^{|\mathsf{R}| \times |\mathsf{ID}|}$, called a (restricted) constraint matrix. For simplicity, the constraint matrix associated with a k-cell c is denoted by $\boldsymbol{R}(\mathsf{c})$ and the constraint matrix for the entire Bézier mesh $\mathbf{B}$ is denoted by $\boldsymbol{R}(\mathbf{B})$.

5.3 Constraint construction

We consider constructing constraints on the interface I between two elements a and b on a d-dimensional Bézier mesh, where $\mathsf{I} = \mathsf{a} \cap \mathsf{b}$. Given two functions defined in Bernstein form, f^{a} defined over a as

$$f^{\mathsf{a}}(\boldsymbol{s}) = \sum_{\mathsf{j} \in \mathsf{ID}(\mathsf{a})} c_{\mathsf{j}} B^{\mathsf{a}}_{\mathsf{j}}(\boldsymbol{s}) \tag{36}$$

and f^{b} defined over b as

$$f^{\mathsf{b}}(\boldsymbol{s}) = \sum_{\mathsf{j} \in \mathsf{ID}(\mathsf{b})} c_{\mathsf{j}} B^{\mathsf{b}}_{\mathsf{j}}(\boldsymbol{s}). \tag{37}$$

We also require the maps from the shared interface I to a and b: $\boldsymbol{\phi}_{\mathsf{I}\to\mathsf{a}} : \hat{\Omega}^{\mathsf{I}} \to \hat{\Omega}^{\mathsf{a}}$ and $\boldsymbol{\phi}_{\mathsf{I}\to\mathsf{b}} : \hat{\Omega}^{\mathsf{I}} \to \hat{\Omega}^{\mathsf{b}}$.

If the functions f^{a} and f^{b} satisfy

$$f^{\mathsf{a}} \circ \boldsymbol{\phi}_{\mathsf{I}\to\mathsf{a}}(\boldsymbol{s}) = f^{\mathsf{b}} \circ \boldsymbol{\phi}_{\mathsf{I}\to\mathsf{b}}(\boldsymbol{s}) \tag{38}$$

for any $\boldsymbol{s} \in \hat{\Omega}^{\mathsf{I}}$ then we say that they are value continuous or $\mathcal{C}^0$. This pointwise constraint over the interface can be converted to a constraint on the coefficients that define the two function through the use of the trace mapping matrix defined previously.

We choose a basis defined on the interface that is sufficient to generate the trace mapping matrices for both a and b: $\boldsymbol{M}^{0,\mathsf{a}}$ and $\boldsymbol{M}^{0,\mathsf{b}}$. Equipped with these matrices, the value constraint given in eq. (38) is equivalent to

$$\sum_{\mathsf{j} \in \mathsf{ID}(\mathsf{a})} \left[\boldsymbol{M}^{0,\mathsf{a}}\right]_{\mathsf{ij}} c_{\mathsf{j}} = \sum_{\mathsf{k} \in \mathsf{ID}(\mathsf{b})} \left[\boldsymbol{M}^{0,\mathsf{b}}\right]_{\mathsf{ik}} c_{\mathsf{k}} \tag{39}$$

for all indices i associated with the basis selected for the interface. It is often convenient to express constraints in homogeneous form:

$$\sum_{\mathsf{j}\in\mathsf{ID}(\mathsf{a})} [\boldsymbol{M}^{\mathsf{a}}]_{\mathsf{ij}}\, c_{\mathsf{j}} - \sum_{\mathsf{k}\in\mathsf{ID}(\mathsf{b})} [\boldsymbol{M}^{\mathsf{b}}]_{\mathsf{ik}}\, c_{\mathsf{k}} = 0. \tag{40}$$

Continuity constraints associated with the nth derivative can be constructed by using the matrices $\boldsymbol{M}^{n,\mathsf{a}}$ and $\boldsymbol{M}^{n,\mathsf{b}}$.

Other work has been done that allows for higher-order continuity constraints for simplex cells to be stated similarly [48]. Thus, the $\mathcal{C}^n$ continuity constraint equations are

$$\sum_{\mathsf{j}\in\mathsf{ID}(\mathsf{a})} \left[\boldsymbol{M}^{\mathcal{C}^n,\mathsf{a}}\right]_{\mathsf{ij}} c_{\mathsf{j}} - \sum_{\mathsf{k}\in\mathsf{ID}(\mathsf{b})} \left[\boldsymbol{M}^{\mathcal{C}^n,\mathsf{b}}\right]_{\mathsf{ik}} c_{\mathsf{k}} = 0, \quad \mathsf{i}\in\mathsf{ID}(\mathsf{I})\ . \tag{41}$$

For detailed derivations of the continuity constraints in two dimensions see appendix B. A few simple exercises that introduce continuity constraints are found in appendix A.1.

6 Splines and the nullspace problem

Given $\boldsymbol{R}(\mathbf{B})$ we can form the corresponding *vector* nullspace problem: Determine the nullspace $\mathsf{N}\subseteq\mathbb{R}^{|\mathsf{ID}(\mathbf{B})|}$ where

$$\mathsf{N}(\mathbf{B}) = \operatorname{span}\{\boldsymbol{c}[\mathbf{B}]\in\mathbb{R}^{|\mathsf{ID}(\mathbf{B})|} : \boldsymbol{R}(\mathbf{B})\boldsymbol{c}[\mathbf{B}] = \mathbf{0}\}. \tag{42}$$

The rank-nullity theorem tells us that the dimension of the nullspace is $\dim\mathsf{N} = |\mathsf{ID}(\mathbf{B})| - \operatorname{rank}(\boldsymbol{R}(\mathbf{B}))$.

6.1 Basis vectors

As with any vector space, any vector in N can be represented as a linear combination of the members of a linearly independent set of vectors. The Ath such vector is called a *basis vector*, denoted by $\boldsymbol{u}_A$, and the set of basis vectors, denoted by $\mathbf{UV}(\mathbf{B})$, form a basis for the nullspace. Of particular importance is determining a *sparse positive basis* for $\mathsf{N}(\mathbf{B})$. This means that we seek to find a set of basis vectors where the set of Bernstein coefficients that define the basis vector has a small number of positive nonzero coefficients (and no negative coefficients). The set of indices associated with the nonzero coefficients of a basis vector $\mathsf{ID}(\boldsymbol{u})$ is sufficiently important that we introduce the symbol id to represent these sets. We also define the set of index sets associated with the basis vectors

$$\mathbf{UI} = \{\mathsf{ID}(\boldsymbol{u}), \boldsymbol{u}\in\mathbf{UV}\}\,. \tag{43}$$

6.2 Basis functions

Since the Bernstein coefficients that compose each basis vector correspond to a set of Bernstein basis functions we can also formulate an equivalent *operator* nullspace problem. In this case, the commonly used terminology changes slightly: Given the Bernstein space $\mathcal{B}(\mathbf{B})$, the constraint space $\mathcal{C}(\mathbf{B})$, and the linear constraint operator $R(\mathbf{B}) : \mathcal{B}(\mathbf{B}) \to \mathcal{C}(\mathbf{B})$, determine the linear subspace of $\mathcal{B}(\mathbf{B})$, called the kernel or nullspace of $\boldsymbol{R}(\mathbf{B})$ and denoted by $\mathcal{N}(\mathbf{B})$, where

$$\mathcal{N}(\mathbf{B}) = \{f\in\mathcal{B}(\mathbf{B}) : R(\mathbf{B})(f) = 0\}. \tag{44}$$

In this case, there is a set of functions, denoted by $\mathsf{UF}(\mathbf{B})$, that span the null space $\mathcal{N}(\mathbf{B})$. The Ath such function is called a *basis function*, denoted by N_A.

6.3 Spline form

Due to the equivalence between eqs. (42) and (44), the basis functions $N_A : \hat{\Omega}^{\mathbf{B}} \to \mathbb{R}$ in $\mathsf{UF}(\mathbf{B})$ not only define $\mathcal{N}(\mathbf{B})$ directly as

$$\mathcal{N}(\mathbf{B}) = \left\{ f : f = \sum_{N_A\in\mathsf{UF}(\mathbf{B})} c_A N_A, c_A\in\mathbb{R} \right\} \tag{45}$$

but are also written in terms of the basis vectors in $\mathbf{UV}(\mathbf{B})$ as

$$N_A(\boldsymbol{s}) = \sum_{\mathsf{i} \in \mathsf{ID}(\mathbf{B})} u_{\mathsf{i}}^A B_{\mathsf{i}}(\boldsymbol{s}) \quad \forall \boldsymbol{s} \in \hat{\Omega}^{\mathsf{c}} \quad \forall \mathsf{c} \in \mathbf{B} \tag{46}$$

where $u_{\mathsf{i}}^A \in \boldsymbol{u}_A \in \mathbf{UV}(\mathbf{B})$.

Importantly, since each N_A satisfies the continuity constraints on $\hat{\Omega}^{\mathbf{B}}$ it is immediately obvious that N_A is not only a function in Bernstein form but also a spline, i.e., it is a piecewise function that satisfies the continuity constraints across all cell interfaces. In other words, finding a sparse positive basis for the nullspace N is equivalent to finding a set of spline basis functions that defines an equivalent spline space. This observation is fundamental to the U-spline construction algorithms presented in this paper.

6.4 Extracted form

For convenience, we often arrange the Bernstein and U-spline basis functions that are nonzero over a k-cell c into vectors, denoted by $\boldsymbol{B}^{\mathsf{c}}$ and $\boldsymbol{N}^{\mathsf{c}}$, respectively. We can then arrange the Bernstein basis vector coefficients on c for each U-spline basis function into corresponding rows in a matrix $\boldsymbol{C}^{\mathsf{c}} \in \mathbb{R}^{|\boldsymbol{N}^{\mathsf{c}}| \times |\boldsymbol{B}^{\mathsf{c}}|}$, called a cell or element extraction matrix. We then have the *extracted form* of the U-spline basis:

$$\boldsymbol{N}^{\mathsf{c}}(\boldsymbol{\xi}^{\mathsf{c}}) = \boldsymbol{C}^{\mathsf{c}} \boldsymbol{B}^{\mathsf{c}}(\boldsymbol{\xi}^{\mathsf{c}}), \quad \boldsymbol{\xi}^{\mathsf{c}} \in \bar{\Omega}^{\mathsf{c}}. \tag{47}$$

In this case, we call the Bernstein basis vector coefficients *extraction coefficients*. Note that at times it is convenient to combine the cell extraction matrices into a global extraction matrix $\boldsymbol{C}$. Representing U-spline basis functions in extracted form is a powerful and convenient abstraction when generalizing finite element frameworks to accommodate smooth spline bases like U-splines. In particular, it is the preferred and simplest representation for smooth splines when the underlying algorithms used to generate the spline basis are not the primary concern.

A few simple exercises that introduce splines and the nullspace problem are found in appendix A.2.

7 Bernstein basis metrics and index measurements

It will be necessary to perform various measurements on submeshes to define relationships between Bernstein bases on adjacent cells. Foundational to these techniques are the Greville points associated with the Bernstein basis functions.

7.1 Greville points

The ith Bernstein polynomial on a k-cell c in the mesh can be associated with a parent point known as a Greville point $\bar{\mathsf{g}}$ or a domain point. For example, for a univariate Bernstein basis function of degree p defined on $\bar{\Omega} = [0, 1]$ this point or abscissa is given by $i/p \in \bar{\Omega}$ and, for a tensor product function, the point is given by the ordered tuple of parent Greville points for each of the Bernstein functions appearing in the product:

$$\bar{\mathsf{g}}(\mathsf{i}) = \left(\mathsf{c}, \left(\frac{i_k}{[\mathsf{p}^{\mathsf{c}}]_k}, i_k \in \mathsf{i} \right) \right). \tag{48}$$

An example is shown in fig. 7.

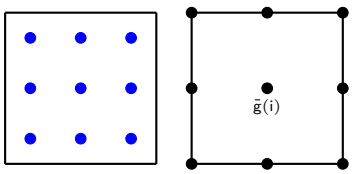


Figure 7: Each Bernstein function in a cell has an associated point known as a Greville point $\bar{\mathsf{g}}$ in the parent domain of the cell, depicted on the right as small black circles.

7.2 Submesh domains

We now need to identify a subdomain and coordinate system that encompasses multiple cells and that allows us to easily take advantage of the ordered derivative property of the Bernstein basis defined over each cell. To accomplish this, given a submesh $\mathsf{K} \subseteq \mathsf{B}$, a *submesh* domain Ω and right-handed orthogonal coordinate system $\alpha \in \Omega$ is constructed. In other words, for a given coordinate $\alpha \in \Omega$ we have that

$$\alpha = \sum_k \alpha_k \boldsymbol{e}_k \tag{49}$$

where

$$\boldsymbol{e}_i \cdot \boldsymbol{e}_j = \delta_{ij}. \tag{50}$$

Note that the origin can be placed anywhere in Ω but is usually placed to simplify the problem at hand. When referencing a particular submesh K, we use Ω^{K} and α^{K}, respectively. Note that the submesh domain corresponding to a set of indices ID is created by setting $\mathsf{K} = \mathsf{C}(\mathsf{ID})$.

To construct Ω^{K} we define a set of *affine* transformations, denoted by $\{\boldsymbol{\varphi}^{\mathsf{c}}\}_{\mathsf{c}\in\mathsf{K}}$, $\boldsymbol{\varphi}^{\mathsf{c}} : \overline{\Omega}^{\mathsf{c}} \to \Omega^{\mathsf{c}}$, where

$$\boldsymbol{\varphi}^{\mathsf{c}}(\boldsymbol{\xi}^{\mathsf{c}}) = \boldsymbol{A}^{\mathsf{c}}\boldsymbol{\xi}^{\mathsf{c}} + \alpha^{\mathsf{c}} \quad \forall \boldsymbol{\xi}^{\mathsf{c}} \in \overline{\Omega}^{\mathsf{c}} \tag{51}$$

and $\boldsymbol{A}^{\mathsf{c}} = \boldsymbol{S}^{\mathsf{c}}\boldsymbol{R}^{\mathsf{c}}$ is a transformation composed of a scaling $\boldsymbol{S}^{\mathsf{c}}$ and a rotation $\boldsymbol{R}^{\mathsf{c}}$. The submesh domain is now constructed as

$$\Omega^{\mathsf{K}} = \bigcup_{\mathsf{c}\in\mathsf{K}} \boldsymbol{\varphi}^{\mathsf{c}}(\overline{\Omega}^{\mathsf{c}}). \tag{52}$$

7.2.1 Indexed submesh domains

The choice of the scaling operator $\boldsymbol{S}^{\mathsf{c}}$ is particularly important and can be chosen to expose certain properties of the Bernstein basis. In particular, we can define a particularly simple scaling that can be leveraged to take advantage of the ordered derivative property of the Bernstein basis. In this case, we define the scaling $\boldsymbol{S}^{\mathsf{c}}$ to be

$$\boldsymbol{S}^{\mathsf{c}} = \begin{bmatrix} {p_0}^{\mathsf{c}} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & {p_{d-1}}^{\mathsf{c}} \end{bmatrix}. \tag{53}$$

In this case, we have that

$$\mathsf{i}^{\mathsf{c}} = \boldsymbol{S}^{\mathsf{c}}\bar{\mathsf{g}}(\mathsf{i}^{\mathsf{c}}) \tag{54}$$

and the submesh Greville points $\mathsf{g} \in \mathbb{Z}^d$ are defined by the mapping $\boldsymbol{\varphi}^{\mathsf{c}}$ as

$$\mathsf{g} = \boldsymbol{\varphi}^{\mathsf{c}}(\bar{\mathsf{g}}). \tag{55}$$

Due to the fact that, under the scaling chosen, all the submesh Greville points are composed of integers, we call this type of submesh domain an *index* domain and that the domain is indexed by the submesh Greville points g. A set of submesh Greville points is denoted by G and a set of sets of submesh Greville points is denoted by $\mathbf{G}$.

We will often choose the transformations $\boldsymbol{\varphi}^{\mathsf{c}}$ so that both the indices and certain chosen features align. For example, when considering two cells of differing degree, the index map will not preserve all topological relationships that exist between the elements (the same vertex may be mapped to two different points under two adjacent maps, etc.) and so we choose to preserve relative orientation and require that a single vertex be preserved under adjacent maps. This is shown in fig. 8. To help differentiate between Greville points on adjacent elements, we often will draw the circles representing the Greville points with respect to a scaled element inset from the boundary, as seen in fig. 9.

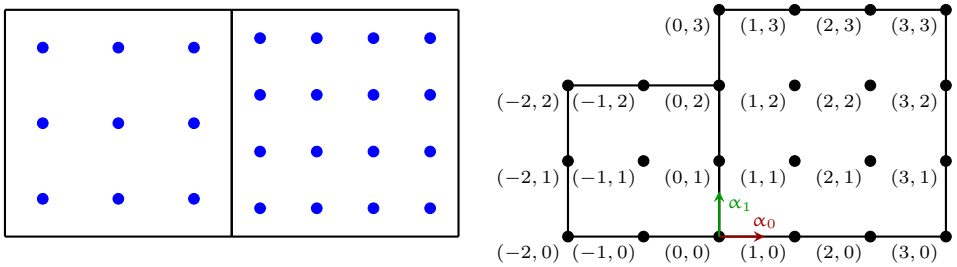


Figure 8: The submesh Greville points on an indexed submesh domain composed of two adjacent cells with differing degree. Each Greville point is labeled with its corresponding coordinate index. The origin chosen for this submesh domain is indicated by the small axis.

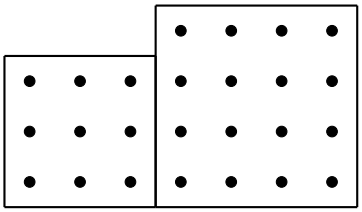


Figure 9: To help differentiate between Greville points on adjacent elements, we often will draw the circles representing the Greville points with respect to a scaled element inset from the boundary, as seen here.

7.3 Equivalence relations and classes

Recall that an equivalence relation partitions a set into equivalence classes. An equivalence class is a set of objects which are defined as equivalent by a given equivalence relation. The set of equivalence classes induced by an equivalence relation is commonly called a partition.

The equivalence relations used here will be based on a classical result in finite-dimensional vector spaces. Given any subspace $\mathcal{W} \subseteq \Omega$, any vector $\alpha \in \Omega$ can be written uniquely in terms of the parallel and perpendicular projectors $\pi^{\parallel}_{\mathcal{W}} : \Omega \to \mathcal{W}$ and $\pi^{\perp}_{\mathcal{W}} : \Omega \to \mathcal{N}(\pi^{\parallel}_{\mathcal{W}})$ as

$$\alpha = \pi^{\parallel}_{\mathcal{W}}(\alpha) + \pi^{\perp}_{\mathcal{W}}(\alpha). \tag{56}$$

Now, given two Greville points g and h we define the equivalence relation $\varpi^{\parallel}_{\mathcal{W}}$ to be true if $\pi^{\parallel}_{\mathcal{W}}(\mathsf{g}) = \pi^{\parallel}_{\mathcal{W}}(\mathsf{h})$ and false otherwise and the equivalence relation $\varpi^{\perp}_{\mathcal{W}}$ to be true if $\pi^{\perp}_{\mathcal{W}}(\mathsf{g}) = \pi^{\perp}_{\mathcal{W}}(\mathsf{h})$ and false otherwise. When distinguishing between parallel and perpendicular projectors is not important we will use the superscript $*$ to convey that the corresponding statement holds for either case.

These equivalence relations can then be used to create corresponding partitions of a set of Greville points G denoted by $\mathsf{G}/\varpi^{*}_{\mathcal{W}}$. We see this in fig. 10 where the Greville points on a cell c are partitioned with respect to an edge e using the equivalence relations $\varpi^{\parallel}_{\mathsf{e}}$ and $\varpi^{\perp}_{\mathsf{e}}$, forming partitions $\mathsf{G}(\mathsf{c})/\varpi^{\parallel}_{\mathsf{e}}$ and $\mathsf{G}(\mathsf{c})/\varpi^{\perp}_{\mathsf{e}}$. We define the shorthand

$$\mathbf{G}^{*}_{\mathcal{W}} = \mathsf{G}/\varpi^{*}_{\mathcal{W}} \tag{57}$$

and for any equivalence class $\mathsf{G} \in \mathbf{G}^{*}_{\mathcal{W}}$ we define

$$\pi^{*}_{\mathcal{W}}(\mathsf{G}) = \pi^{*}_{\mathcal{W}}(\mathsf{g}), \quad \mathsf{g} \in \mathsf{G} \in \mathbf{G}^{*}_{\mathcal{W}} \tag{58}$$

since every point in an equivalence class has the same projected value.

We also extend the equivalence relation $\varpi^{*}_{\mathcal{W}}$ to sets of points. We say that a set of points A is equivalent to another set of points B if they project to the same set of points under the projection

$$\pi^{*}_{\mathcal{W}}(\mathsf{A}) = \pi^{*}_{\mathcal{W}}(\mathsf{B}). \tag{59}$$

We can partition sets containing sets of points with this relation.

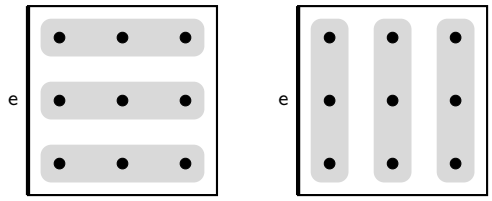


Figure 10: The parallel and perpendicular equivalence relations $\varpi_{\mathsf{e}}^{\parallel}$ and $\varpi_{\mathsf{e}}^{\perp}$ are used to partition the Greville points on a cell c with respect to the adjacent edge e, forming partitions $\mathsf{G}(\mathsf{c})/\varpi_{\mathsf{e}}^{\parallel}$ and $\mathsf{G}(\mathsf{c})/\varpi_{\mathsf{e}}^{\perp}$.

7.4 Alignment

In order to compare indices on two or more adjacent elements, we define the concept of *alignment*. We first introduce the concept in two dimensions where an intuitive geometric description is available and then generalize those ideas to arbitrary dimensions.

7.4.1 Alignment in two dimensions

We consider two elements a and b on a two-dimensional mesh. In order to compare indices on a and b, we construct an index domain on the two elements so that a set of boundary edges $\{\mathsf{e} \in \partial\mathsf{a} \cap \partial\mathsf{b} : \mathsf{v} \cap \mathsf{e} \neq \varnothing\}$ that share a common vertex v are aligned and so that v lies at the origin. An example of an index domain satisfying these conditions is shown in fig. 8. Next, we form the set of points shared between both elements under this index mapping $\mathsf{GI} = \mathsf{GA} \cap \mathsf{GB}$, where GA are the points on element a and GB are the points on element b. Figure 11 shows an example of these sets on two adjacent elements with degrees $(1, 1)$ and $(1, 2)$.

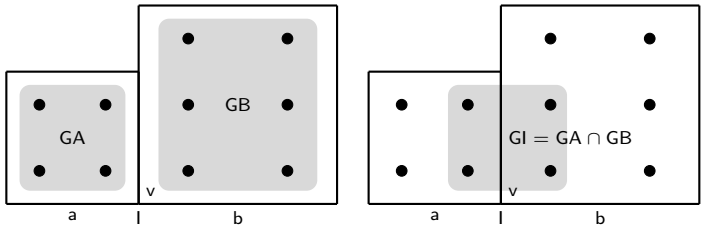


Figure 11: An example of the sets GA, GB, and GI on two adjacent elements a and b with degrees $(1, 1)$ and $(1, 2)$.

For each of the three sets GA, GB, and GI we determine the projected point that is farthest from v:

$$\mathsf{g}^{\mathsf{a}}_{\max} = \mathsf{g} \in \pi_{\mathsf{I}}^{\parallel}(\mathsf{GA}) : \|\mathsf{g} - \mathsf{v}\|_2 = \max_{\mathsf{g} \in \pi_{\mathsf{I}}^{\parallel}(\mathsf{GA})} \|\mathsf{g} - \mathsf{v}\|_2 \,, \tag{60}$$

$$\mathsf{g}^{\mathsf{b}}_{\max} = \mathsf{g} \in \pi_{\mathsf{I}}^{\parallel}(\mathsf{GB}) : \|\mathsf{g} - \mathsf{v}\|_2 = \max_{\mathsf{g} \in \pi_{\mathsf{I}}^{\parallel}(\mathsf{GB})} \|\mathsf{g} - \mathsf{v}\|_2 \,, \tag{61}$$

$$\mathsf{g}^{\mathsf{I}}_{\max} = \mathsf{g} \in \mathsf{GI} : \|\mathsf{g} - \mathsf{v}\|_2 = \max_{\mathsf{g} \in \mathsf{GI}} \|\mathsf{g} - \mathsf{v}\|_2 \,. \tag{62}$$

These points can be used to define difference vectors:

$$\Delta\mathsf{g}^{\mathsf{a}}_{\max} = \mathsf{g}^{\mathsf{a}}_{\max} - \mathsf{g}^{\mathsf{I}}_{\max} \tag{63}$$

$$\Delta\mathsf{g}^{\mathsf{b}}_{\max} = \mathsf{g}^{\mathsf{b}}_{\max} - \mathsf{g}^{\mathsf{I}}_{\max}. \tag{64}$$

Figure 12 shows the projected points for the elements in fig. 11 as black circles along the shared interface I. The associated difference vector $\Delta\mathsf{g}^{\mathsf{b}}_{\max}$ is shown with a curly bracket. The difference vector $\Delta\mathsf{g}^{\mathsf{a}}_{\max}$ in this example has length zero and is not shown.

Given a difference vector $\Delta\mathsf{g}^{a}_{\max}$ we define a set of offset points given a point g as

$$\mathsf{Offset}_{\mathsf{g}}(\Delta\mathsf{g}^{a}_{\max}) = \{\mathsf{g}\} \cup \{\mathsf{g} + \Delta\mathsf{g}^{a}_{\max}\} \,. \tag{65}$$

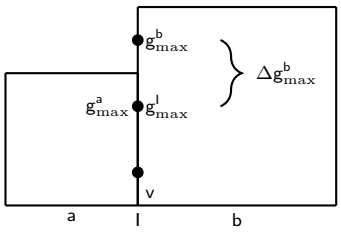


Figure 12: The projected points for the two elements in fig. 11 are shown as black circles along the shared interface I. The difference vector $\Delta \mathsf{g}^{\mathsf{b}}_{\max}$ is shown with a curly bracket.

Now, using $\mathsf{AABB}(\mathsf{P})$ to denote an axis-aligned bounding box associated with a set of points P, for every point $\mathsf{g}^{\mathsf{I}} \in \mathsf{GI}$ there are unique regions associated with GA and GB

$$\Omega^{\mathsf{a}}_{\mathsf{g}^{\mathsf{I}}} = \mathsf{AABB}(\mathsf{Offset}_{\mathsf{g}^{\mathsf{I}}}(\Delta \mathsf{g}^{\mathsf{a}}_{\max})), \tag{66}$$

$$\Omega^{\mathsf{b}}_{\mathsf{g}^{\mathsf{I}}} = \mathsf{AABB}(\mathsf{Offset}_{\mathsf{g}^{\mathsf{I}}}(\Delta \mathsf{g}^{\mathsf{b}}_{\max})). \tag{67}$$

Each region defines a subset of the equivalence classes on the associated faces:

$$\mathbf{GA}^{\parallel}_{\mathsf{I},\mathsf{g}^{\mathsf{I}}} = \left\{ \mathsf{G} \in \mathbf{GA}^{\parallel}_{\mathsf{I}} : \pi^{\parallel}_{\mathsf{I}}(\mathsf{G}) \subseteq \Omega^{\mathsf{a}}_{\mathsf{g}^{\mathsf{I}}}, \mathbf{GA}^{\parallel}_{\mathsf{I}} = \mathsf{GA}/\varpi^{\parallel}_{\mathsf{I}} \right\}, \tag{68}$$

$$\mathbf{GB}^{\parallel}_{\mathsf{I},\mathsf{g}^{\mathsf{I}}} = \left\{ \mathsf{G} \in \mathbf{GB}^{\parallel}_{\mathsf{I}} : \pi^{\parallel}_{\mathsf{I}}(\mathsf{G}) \subseteq \Omega^{\mathsf{b}}_{\mathsf{g}^{\mathsf{I}}}, \mathbf{GB}^{\parallel}_{\mathsf{I}} = \mathsf{GB}/\varpi^{\parallel}_{\mathsf{I}} \right\}. \tag{69}$$

For every $\mathsf{g}^{\mathsf{I}} \in \mathsf{GI}$ there is a unique set of equivalence classes from $\mathbf{GA}^{\parallel}_{\mathsf{I}}$ and another unique set from $\mathbf{GB}^{\parallel}_{\mathsf{I}}$. We say that these two sets of equivalence classes are aligned and define a set containing the aligned sets of equivalence classes as

$$\mathbf{Align}_{\mathsf{g}^{\mathsf{I}}} = \mathbf{GA}^{\parallel}_{\mathsf{I},\mathsf{g}^{\mathsf{I}}} \cup \mathbf{GB}^{\parallel}_{\mathsf{I},\mathsf{g}^{\mathsf{I}}}. \tag{70}$$

Figure 13 shows each set of aligned sets of equivalence classes for the elements in fig. 11 along the shared interface I. We use superscripts to refer to subsets of $\mathbf{Align}_{\mathsf{g}^{\mathsf{I}}}$ formed from equivalence classes associated with a single cell. $\mathbf{Align}^{\mathsf{a}}_{\mathsf{g}^{\mathsf{I}}}$ is the set of equivalence classes in $\mathbf{Align}_{\mathsf{g}^{\mathsf{I}}}$ associated with points on a. This approach may appear ambiguous for simplicial cells but because we restrict ourselves to $\mathcal{C}^0$ interfaces around simplicial cells there is no ambiguity.

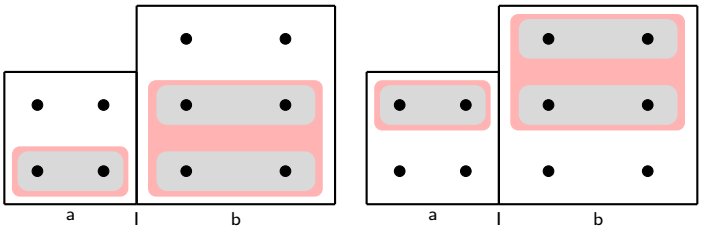


Figure 13: Given a Greville point on the shared interface $\mathsf{g}^{\mathsf{I}} \in \mathsf{GI}$, a set of equivalence classes on element a (denoted $\mathbf{GA}^{\parallel}_{\mathsf{I},\mathsf{g}^{\mathsf{I}}}$) is aligned with a corresponding set of equivalence classes on element b (denoted $\mathbf{GB}^{\parallel}_{\mathsf{I},\mathsf{g}^{\mathsf{I}}}$), together forming a set of aligned sets of equivalence classes (denoted $\mathbf{Align}_{\mathsf{g}^{\mathsf{I}}}$).

7.4.2 Alignment in arbitrary dimensions

Consider $m \geq 2$ neighboring elements $\mathsf{E}_k, k \in \{1, ..., m\}$ on a d-dimensional Bézier mesh that meet at a common lower-dimensional adjacent cell c. Recall that each element E_k has a prescribed Bernstein space and set of Greville points GE_k. The cell c is assigned a Bernstein space with polynomial degree p_{c} equal to the *minimum* of all degrees on E_k in each parametric direction parallel to c. In the case that c is a vertex, the Bernstein basis is assumed to be constant.

The set GE_k can be partitioned into equivalence classes with respect to c as

$$\mathbf{GE}^{\|}_{k,\mathsf{c}} = \mathsf{GE}_k / \varpi^{\|}_{\mathsf{c}}. \tag{71}$$

We construct trace mapping matrices $\boldsymbol{M}_k$ for each E_k with respect to c and for each $\mathsf{i} \in \mathsf{ID}(\mathsf{c})$ collect indices on E_k that correspond to the nonzero coefficients on the ith row of $\boldsymbol{M}_k$, denoted $\mathsf{NZ}(\boldsymbol{M}_k, \mathsf{i})$, as described in section 4.3.

For each $\mathsf{i} \in \mathsf{ID}(\mathsf{c})$ there is a unique set of equivalence classes from each $\mathbf{GE}^{\|}_{k,\mathsf{c}}$ defined as

$$\mathbf{GE}^{\|}_{k,\mathsf{c},\mathsf{i}} = \left\{ \mathsf{G} \in \mathbf{GE}^{\|}_{k,\mathsf{c}} \,:\, \pi^{\|}_{\mathsf{c}}(\mathsf{G}) \subseteq \pi^{\|}_{\mathsf{c}}(\bar{\mathsf{g}}(\mathsf{NZ}(\boldsymbol{M}_k, \mathsf{i}))) \right\}. \tag{72}$$

We say that these sets of equivalence classes, one set from each element E_k for a given i, are aligned and define a set containing the aligned sets of equivalence classes as

$$\mathbf{Align}_{\mathsf{c},\mathsf{i}} = \bigcup_k \mathbf{GE}^{\|}_{k,\mathsf{c},\mathsf{i}}. \tag{73}$$

As we did in the two-dimensional case, we use superscripts to refer to subsets of $\mathbf{Align}_{\mathsf{c},\mathsf{i}}$ formed from equivalence classes associated with a single element. For example, $\mathbf{Align}^{\mathsf{E}_k}_{\mathsf{c},\mathsf{i}}$ is the set of equivalence classes in $\mathbf{Align}_{\mathsf{c},\mathsf{i}}$ associated with points on element E_k. In this context, we refer to $\mathsf{i} \in \mathsf{ID}(\mathsf{c})$ as an alignment index. See figs. 14 and 15 for simple illustrations of these ideas.

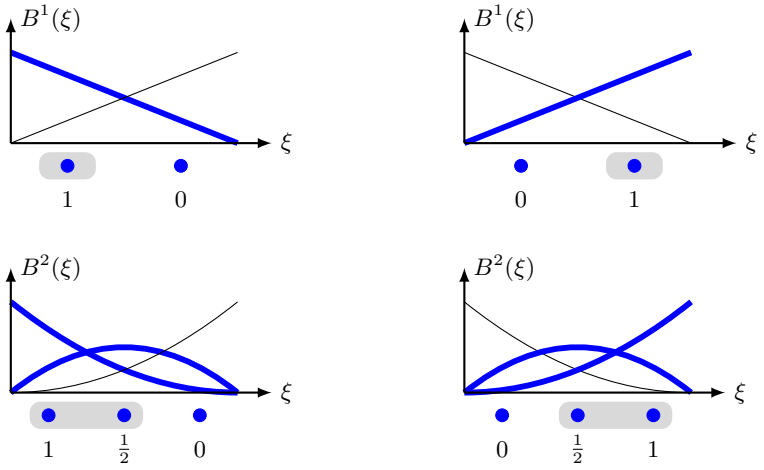


Figure 14: The concept of alignment follows naturally from the fact that multiple nonzero coefficients corresponding to higher-degree Bernstein basis functions are required to represent a single Bernstein function of lower degree (see section 2.2).

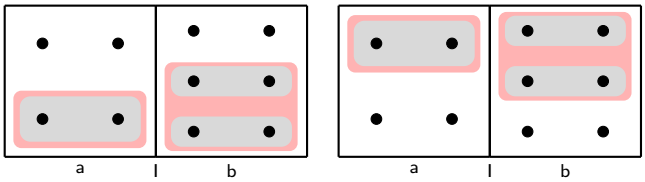


Figure 15: Given an alignment index i on a shared interface I, $\mathsf{i} \in \mathsf{ID}(\mathsf{I})$, a set of equivalence classes on element a (denoted $\mathbf{Align}^{\mathsf{a}}_{\mathsf{I},\mathsf{i}}$) is aligned with a corresponding set of equivalence classes on element b (denoted $\mathbf{Align}^{\mathsf{b}}_{\mathsf{I},\mathsf{i}}$), together forming a set of aligned sets of equivalence classes (denoted $\mathbf{Align}_{\mathsf{I},\mathsf{i}}$).

8 Basis vectors for k-cell nullspaces

Given $\mathsf{R}(\mathsf{c})$ for any k-cell $\mathsf{c} \in \mathbf{B}$ we want to determine the set or system of basis vectors, denoted by $\mathbf{BV}(\mathsf{c})$, that span the nullspace of the corresponding restricted constraint matrix $\boldsymbol{R}(\mathsf{c})$. Importantly, we require that a set of admissibility conditions defined in section 9.2 be satisfied by the mesh $\mathbf{B}$. With this restriction we can use the ordered derivative property of the Bernstein basis to efficiently identify the index support of each basis vector through a series of operations on $\mathsf{ID}(\mathbf{B})$.

The set of index sets associated with a set of basis vectors is used with sufficient frequency that we use $\mathbf{BI}(\mathsf{c}) = \{\mathsf{ID}(\boldsymbol{v}), \boldsymbol{v} \in \mathbf{BV}(\mathsf{c})\}$ to represent it. We also define the set of mapped index sets associated with the basis vectors of the cell c to be $\mathbf{BG}(\mathsf{c}) = \{\{\boldsymbol{\phi}(\mathsf{i}), \mathsf{i} \in \mathsf{ID}\}, \mathsf{ID} \in \mathbf{BI}(\mathsf{c})\}$ for a suitably chosen index mapping $\boldsymbol{\phi}$.

To aid the reader in understanding these concepts we begin in one dimension and then move to interface and vertex basis vectors in two dimensions followed last by the technical description of k-cell basis vectors in arbitrary dimensions. Most of the needed intuition for the general setting can be developed in the one- and two-dimensional settings.

8.1 Basis vectors in one dimension

The constraint matrix to enforce $\mathcal{C}^{\vartheta}$ across an interface $\boldsymbol{R}(\mathsf{I})$ in a one-dimensional mesh has rank $\vartheta + 1$. This is a consequence of the ordering of the Bernstein constraints. A suitable choice of permutation of the $\vartheta + 1$ rows will produce a constraint matrix in lower triangular form and therefore the matrix has rank $\vartheta + 1$.

Given an interface I in a one-dimensional mesh where I has been assigned a continuity ϑ, the minimum number of nonzero contiguous Bernstein coefficients in any basis vector having nonzero indices in both edges adjacent to I is $\vartheta + 2$. Let $\boldsymbol{r}_\rho$ represent the row of the constraint matrix $\boldsymbol{R}(\mathsf{I})$ associated with the constraint for continuity of level $\rho, 0 \leq \rho \leq \vartheta$; there are a total of $\vartheta + 1$ rows in the matrix. Begin with $\rho = 0$ and pick a unit vector $\boldsymbol{u}_{-1}$ with a single positive nonzero entry that satisfies $|\boldsymbol{r}_0 \boldsymbol{u}_{-1}| > 0$. Now pick a second unit vector $\boldsymbol{u}_0'$ with a single nonzero entry that satisfies $|\boldsymbol{r}_0 \boldsymbol{u}_0'| > 0$ and $\langle \boldsymbol{u}_{-1}, \boldsymbol{u}_0' \rangle = 0$. The two vectors can be combined to form a new vector $\boldsymbol{u}_0 = \boldsymbol{u}_{-1} + a\boldsymbol{u}_0'$ that satisfies $\boldsymbol{r}_0 \boldsymbol{u}_0 = 0$. The system can be solved to find that

$$a = -\frac{\boldsymbol{r}_0 \boldsymbol{u}_{-1}}{\boldsymbol{r}_0 \boldsymbol{u}_0'} \tag{74}$$

and so

$$\boldsymbol{u}_0 = \boldsymbol{u}_{-1} - \boldsymbol{u}_0' \frac{\boldsymbol{r}_0 \boldsymbol{u}_{-1}}{\boldsymbol{r}_0 \boldsymbol{u}_0'}. \tag{75}$$

By construction, the vector $\boldsymbol{u}_0$ has two nonzero entries. The same procedure can be applied to obtain solutions to the higher-order continuity constraints. This gives rise to the recursive definition:

$$\boldsymbol{u}_\rho = \boldsymbol{u}_{\rho-1} - \boldsymbol{u}_\rho' \frac{\boldsymbol{r}_\rho \boldsymbol{u}_{\rho-1}}{\boldsymbol{r}_\rho \boldsymbol{u}_\rho'} \tag{76}$$

where at each stage $\boldsymbol{u}_\rho'$ is chosen so that $\langle \boldsymbol{u}_{\rho-1}, \boldsymbol{u}_\rho' \rangle = 0$ and so that the nonzero entry in $\boldsymbol{u}_\rho'$ is adjacent to the nonzero entries of $\boldsymbol{u}_{\rho-1}$. This means that each additional constraint applied adds one additional nonzero entry and so the final vector of coefficients will always have $\vartheta + 2$ nonzero entries. At every stage there are exactly two choices for the next vector $\boldsymbol{u}_\rho'$. This is due to the ordered increase in the size of constraint vectors in Bernstein form. Each constraint vector has two more nonzero entries than the previous one. Note that this construction relies on the fact that $\boldsymbol{r}_\rho \boldsymbol{u}_{\rho-1} \neq 0$ which is not established rigorously here.

Because the number of nonzero contiguous coefficients in the basis vectors of a smoothness constraint system is given by the maximum smoothness of the system, it is possible to determine the support of the basis vectors without solving the constraint system. It is sufficient to select all unique contiguous vectors with $\vartheta + 2$ entries that have entries corresponding to functions on either side of the interface. Examples of these one-dimensional basis vectors for various continuities are shown in fig. 16. Restricting the constraint system to these nonzero entries will result in a rank one nullspace problem (see section 10.2 for an approach to solving rank one nullspace problems).

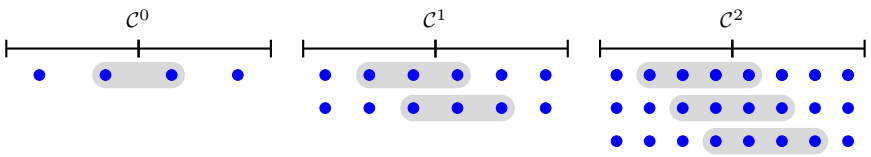


Figure 16: The nonzero coefficients of one-dimensional basis vectors on $\mathcal{C}^0$, $\mathcal{C}^1$, and $\mathcal{C}^2$ interfaces, respectively. Each basis vector has $\vartheta + 2$ contiguous nonzero entries.

8.2 Interface basis vectors in two dimensions

Equation (76) can easily be generalized to an edge interface between faces with matching tensor product bases along the interface. It can be seen that the constraints separate into separate systems with exactly the same nonzero constraint values as would occur in the univariate setting repeated for each equivalence class. A simple example of a constraint matrix is given in eq. (77). The indices corresponding to nonzero entries are highlighted with gray boxes in fig. 17. In this setting, the results of eq. (76) can be applied directly to obtain the nonzero entries of each basis vector that has nonzero indices in both faces. The indices of the nonzero entries can similarly be obtained if one does not wish to solve for the values. The indices corresponding to two representative basis vectors are shown in fig. 18. Each one was obtained by choosing three coefficients from a row with contiguous indices with at least one coefficient on each side of the interface.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & -2 & 2 & 0 \end{bmatrix} \tag{77}$$

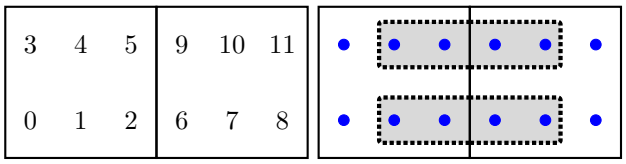


Figure 17: Coefficients corresponding to the nonzero entries in the rows of the constraint matrix. Each grey block surrounds the nonzero entries of a pair of rows in the matrix given in eq. (77).

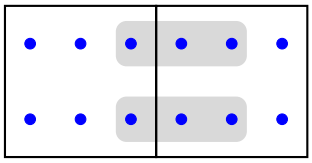


Figure 18: The indices associated with nonzero entries of two representative basis vectors for eq. (77) are shown. Each gray box indicates the nonzero entries in one basis vector.

If the Bernstein basis in each element does not match along the interface edge then the situation is significantly more complex. Consider a system consisting of two elements, one with degree $(1, 1)$ and the other with degree $(1, 2)$ connected by a $\mathcal{C}^0$ interface such that the bases do not match on the shared interface. This is shown in fig. 19. The constraint matrix is given in eq. (78). A pictorial representation of the nonzero entries associated with each row in the constraint matrix is given in fig. 19. The constraints are now fully coupled and hence the one-dimensional result may not be directly applied to compute the indices of the nonzero entries of the basis vectors.

$$\begin{bmatrix} 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{78}$$

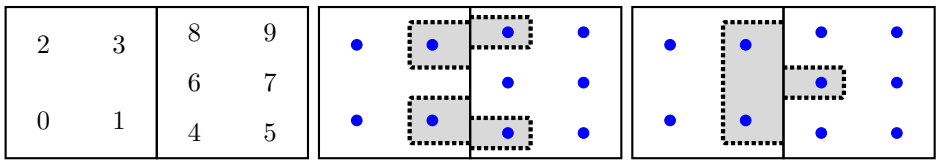


Figure 19: Indexing convention and coefficients corresponding to the nonzero entries in the constraint matrix given in eq. (78). The mesh has two elements with degrees $(1,1)$ and $(1,2)$, connected by a $\mathcal{C}^0$ interface. Each contiguous gray region surrounds the nonzero entries of a row in the matrix given in eq. (78).

For every interface in a Bézier mesh there is a natural association between the equivalence classes on one side of the interface and the equivalence classes on the other side as described by eq. (70). This association allows the results of eq. (76) to be generalized to the construction of interface basis vectors for higher-dimensional elements with local tensor product polynomial bases.

For two elements a and b sharing an interface $\mathsf{I} = \mathsf{a} \cap \mathsf{b}$ with assigned smoothness ϑ and having mapped index sets $\mathsf{GA} = \pi_{\mathsf{I}}^{\parallel}(\varphi^{\mathsf{a}}(\mathsf{ID}(\mathsf{a})))$ and $\mathsf{GB} = \pi_{\mathsf{I}}^{\parallel}(\varphi^{\mathsf{b}}(\mathsf{ID}(\mathsf{b})))$ with intersection set $\mathsf{GI} = \mathsf{GA} \cap \mathsf{GB}$, there are $\vartheta + 1$ basis vectors with indices from both elements for every index in GI. The indices of the nonzero entries of these basis vectors are defined for all integers $0 \leq m \leq \vartheta$. For every $\mathsf{g}^{\mathsf{I}} \in \mathsf{GI}$ form the sets

$$\mathsf{ID}^{\mathsf{a}}_{\mathsf{g}^{\mathsf{I}},m} = \bigcup_{\mathsf{G} \in \mathbf{Align}^{\mathsf{a}}_{\mathsf{g}^{\mathsf{I}}}} \left\{ \mathsf{i}^{\mathsf{a}} : \mathsf{g} = \varphi^{\mathsf{a}}(\bar{\mathsf{g}}(\mathsf{i}^{\mathsf{a}})) \in \mathsf{G}, \left|\pi_{\mathsf{I}}^{\perp}(\mathsf{g})\right| \leq m \right\}, \tag{79}$$

$$\mathsf{ID}^{\mathsf{b}}_{\mathsf{g}^{\mathsf{I}},\vartheta-m} = \bigcup_{\mathsf{G} \in \mathbf{Align}^{\mathsf{b}}_{\mathsf{g}^{\mathsf{I}}}} \left\{ \mathsf{i}^{\mathsf{b}} : \mathsf{g} = \varphi^{\mathsf{b}}(\bar{\mathsf{g}}(\mathsf{i}^{\mathsf{b}})) \in \mathsf{G}, \left|\pi_{\mathsf{I}}^{\perp}(\mathsf{g})\right| \leq \vartheta - m \right\}. \tag{80}$$

The indices for the mth function associated with g^{I} are given by the union of these two sets:

$$\mathsf{ID}_{\mathsf{g}^{\mathsf{I}},m} = \mathsf{ID}^{\mathsf{a}}_{\mathsf{g}^{\mathsf{I}},m} \cup \mathsf{ID}^{\mathsf{b}}_{\mathsf{g}^{\mathsf{I}},\vartheta-m}. \tag{81}$$

Once the index set has been obtained, the nonzero coefficients associated with the basis vector may be obtained by solving the smoothness constraints restricted to the index set. This also holds for $\mathcal{C}^0$ interfaces between triangle-triangle or quad-triangle pairs. The set of all interface basis vectors with indices from both elements adjacent to I is obtained by considering all possible values of g^{I} and m.

The interface basis vectors with indices from both elements in fig. 19 are highlighted in fig. 20. These basis vectors are in the nullspace of the constraint matrix in eq. (78). A slightly more complex example is seen in fig. 21, where two faces with degrees $\mathsf{p} = (2,2)$ and $\mathsf{p} = (3,3)$ are separated by an interface with $\mathcal{C}^1$ continuity. The six basis vectors with indices from both faces are highlighted.

See appendix A.3 for a few simple exercises that provide greater insight into basis vectors and rank one nullspace problems.

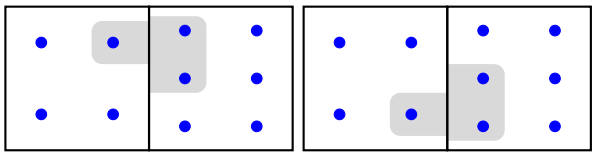


Figure 20: The basis vectors with indices from both elements adjacent to the interface in fig. 19. These basis vectors span the null space of the constraint matrix given in eq. (78).

8.3 k-cell basis vector preliminaries

The construction of basis vectors on k-cells of dimension lower than an interface is more complex. Before proceeding with the construction of basis vectors for an arbitrary k-cell a^k we define several preliminary concepts.

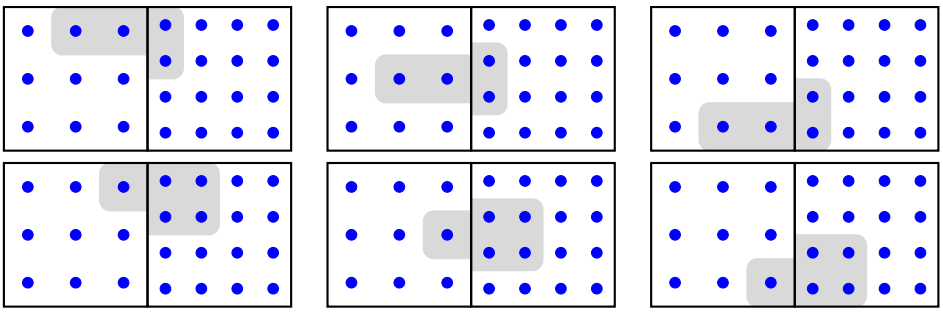


Figure 21: The basis vectors with indices from both faces adjacent to a $\mathcal{C}^1$ interface. The faces have degrees $\mathsf{p} = (2, 2)$ and $\mathsf{p} = (3, 3)$.

8.3.1 Spokes and interface-element pairs

A *spoke*, denoted by ρ, is a pair containing an element E and an adjacent $(k+1)$-cell b^{k+1}, both of which are adjacent to a^k. More specifically, a spoke is defined as

$$\rho = \left(\mathsf{a}^k, \mathsf{E}, \mathsf{b}^{k+1}\right) \text{ s.t. } \mathsf{E} \in \mathsf{ADJ}^d(\mathsf{a}^k), \mathsf{b}^{k+1} \in \mathsf{ADJ}^{k+1}(\mathsf{a}^k), \mathsf{b}^{k+1} \in \mathsf{ADJ}^{k+1}(\mathsf{E}). \tag{82}$$

Each spoke ρ has a corresponding perpendicular interface-element pair, denoted by t, and defined as

$$\mathsf{t} = \left(\mathsf{a}^k, \mathsf{E}, \mathsf{I}\right) \text{ s.t. } \mathsf{E} \in \mathsf{ADJ}^d(\mathsf{a}^k), \mathsf{I} \in \mathsf{ADJ}^{d-1}(\mathsf{a}^k), \mathsf{I} \in \mathsf{ADJ}^{d-1}(\mathsf{E}). \tag{83}$$

In fig. 22a, the spokes adjacent to a vertex v are depicted as dashed lines.

We define several operations on spokes and interface-element pairs. The operator $\mathsf{t}^\perp(\rho)$ gives the interface-element pair perpendicular to a spoke on the same element and is defined as

$$\mathsf{t}^\perp(\mathsf{a}^k, \mathsf{E}, \mathsf{b}^{k+1}) = \left(\mathsf{a}^k, \mathsf{E}, \mathsf{I}\right) \text{ s.t. } \mathsf{I} \in \mathsf{ADJ}^{d-1}(\mathsf{E}), \boldsymbol{s}_{\mathsf{E}}^\perp(\mathsf{I}) \subseteq \boldsymbol{s}_{\mathsf{E}}^\parallel(\mathsf{b}^{k+1}). \tag{84}$$

The operator $\rho^\perp(\mathsf{t})$ gives the spoke perpendicular to an interface-element pair on the same element and is defined as

$$\rho^\perp(\mathsf{a}^k, \mathsf{E}, \mathsf{I}) = \left(\mathsf{a}^k, \mathsf{E}, \mathsf{b}^{k+1}\right) \text{ s.t. } \mathsf{b}^{k+1} \in \mathsf{ADJ}^{k+1}(\mathsf{a}^k), \mathsf{b}^{k+1} \in \mathsf{ADJ}^{k+1}(\mathsf{E}), \boldsymbol{s}_{\mathsf{E}}^\perp(\mathsf{I}) \subseteq \boldsymbol{s}_{\mathsf{E}}^\parallel(\mathsf{b}^{k+1}). \tag{85}$$

The operator $\mathsf{t}^\parallel(\mathsf{t})$ gives the element-interface pair on the adjacent element sharing the same interface and is defined as

$$\mathsf{t}^\parallel(\mathsf{a}^k, \mathsf{E}_a, \mathsf{I}) = \left(\mathsf{a}^k, \mathsf{E}_b, \mathsf{I}\right) \text{ s.t. } \mathsf{E}_a \cap \mathsf{E}_b = \mathsf{I}, \mathsf{E}_b \in \mathsf{ADJ}^d(\mathsf{I}). \tag{86}$$

The operator $\rho^\parallel(\mathsf{a}^k, \mathsf{b}^{k+1})$ gives the set of spokes that share the same $(k+1)$-cell b^{k+1} and is defined as

$$\rho^\parallel(\mathsf{a}^k, \mathsf{b}^{k+1}) = \left\{ \left(\mathsf{a}^k, \mathsf{E}, \mathsf{b}^{k+1}\right) \text{ s.t. } \mathsf{E} \in \mathsf{ADJ}^d(\mathsf{b}^{k+1}) \right\}. \tag{87}$$

We let $\vartheta_{\max}^\perp(\rho) = \vartheta_{\max}^\perp(\mathsf{a}^k, \mathsf{b}^{k+1}) : \mathsf{a}^k, \mathsf{b}^{k+1} \in \rho$.

8.3.2 Inclusion distances

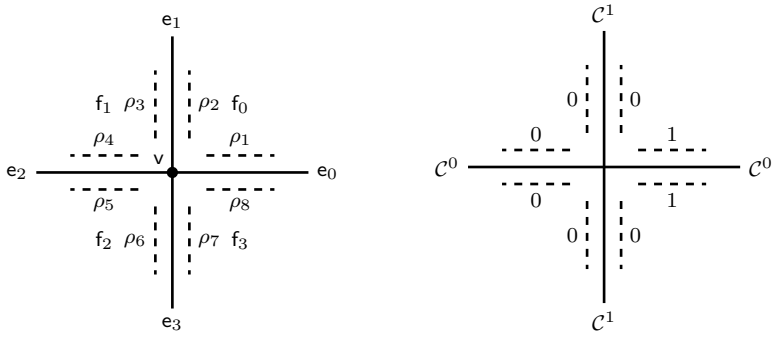
We assign a number called an *inclusion distance* to each spoke, denoted $\mathsf{inc}(\rho)$, which will be used to determine the set of $(k+1)$-cell basis vectors from each $(k+1)$-cell adjacent to a^k that are included in the k-cell basis vector.

Each inclusion distance has the following properties:

- $0 \leq \mathsf{inc}(\rho) \leq \vartheta_{\max}^\perp(\rho)$
- $\mathsf{inc}(\rho) = \vartheta_{\max}^\perp(\rho) - \mathsf{inc}(\rho^\perp(\mathsf{t}^\parallel(\mathsf{t}^\perp(\rho))))$
- $\forall \{\rho_a, \rho_b\} \subseteq \rho^\parallel(\mathsf{b}^{k+1}),\ \mathsf{inc}(\rho_a) = \mathsf{inc}(\rho_b)$.

For the construction of a k-cell basis vector on a d-dimensional mesh, there are $(d-k)$ spokes that share each element adjacent to the k-cell. We begin the process of constructing a set of inclusion distances for a k-cell a^k by choosing one of these elements E and then choosing $(d-k)$ inclusion distances $\mathsf{inc}_1 ... \mathsf{inc}_{d-k}$, one for each of the spokes on E, $\rho_j, j \in \{1, ..., d-k\}, \mathsf{E} \in \rho_j$, that satisfy the requirement $0 \leq \mathsf{inc}_j \leq \vartheta^{\perp}_{\max}(\rho_j), j \in \{1, ..., d-k\}$. The set of inclusion distances for each $(k+1)$-cell adjacent to a^k are fixed by the choice of $\mathsf{inc}_1 ... \mathsf{inc}_{d-k}$ on E. We denote the set of inclusion distances for all $(k+1)$-cells adjacent to a^k by $\mathsf{INC}^{\mathsf{inc}_1,...,\mathsf{inc}_{d-k}}$ and denote the value associated with the $(k+1)$-cell b^{k+1} by $\mathsf{INC}^{\mathsf{inc}_1,...,\mathsf{inc}_{d-k}}_{\mathsf{b}^{k+1}}$.

In fig. 22b, a set of inclusion distances have been chosen for the spokes adjacent to a vertex v, on a system where each face has a biquadratic basis, and the continuity on each edge is as indicated.



(a) Each spoke ρ_i on a regular vertex v in a two-dimensional mesh is depicted as a dashed line.

(b) An example of a choice of inclusion distances, labeled adjacent to each spoke.

Figure 22: Examples of spokes and inclusion distances.

8.3.3 Alignment sets

We define the alignment sets on the k-cell a^k. A k-dimensional Bernstein basis is placed in $\hat{\Omega}^{\mathsf{a}^k}$ with polynomial degree $\mathsf{p}_{\mathsf{a}^k}$ equal to the *minimum* on the adjacent elements in each parametric direction parallel to a^k, and for each $\mathsf{i} \in \mathsf{ID}(\mathsf{a}^k)$ we construct the sets of aligned equivalence classes, contained in $\mathbf{Align}_{\mathsf{a}^k,\mathsf{i}}$ (see section 7.4). For each $\mathsf{i} \in \mathsf{ID}(\mathsf{a}^k)$ there exists a set of $(k+1)$-cell basis vectors $\mathbf{HBV}_{\mathsf{i}}(\mathsf{a}^k)$ such that

$$\mathbf{HBV}_{\mathsf{i}}(\mathsf{a}^k) = \left\{ \boldsymbol{n} \in \mathbf{HBV}(\mathsf{a}^k) \,:\, \mathsf{G}(\mathsf{ID}(\boldsymbol{n})) \subseteq \mathbf{Align}_{\mathsf{a}^k,\mathsf{i}} \right\} \tag{88}$$

where

$$\mathbf{HBV}(\mathsf{a}^k) = \bigcup_{\mathsf{b}^{k+1} \in \mathsf{ADJ}^{k+1}(\mathsf{a}^k)} \mathbf{BV}(\mathsf{b}^{k+1}). \tag{89}$$

8.4 Overview of k-cell basis vector construction

In general, the algorithm for constructing k-cell basis vectors is recursive. It assumes the basis vectors on the adjacent $(k+1)$-cells have already been constructed, the base case being the d-cell basis vectors (the Bernstein functions). Each k-cell basis vector is a linear combination of $(k+1)$-cell basis vectors on adjacent $(k+1)$-cells. *Composite* k-cell basis vectors are formed by taking a linear combination of multiple $(k+1)$-cell basis vectors, while *simple* k-cell basis vectors are formed from just one $(k+1)$-cell basis vector.

The algorithm for constructing the basis vectors on a k-cell a^k is summarized as follows:

1. Construct the basis vectors on adjacent $(k+1)$-cells $\mathsf{b}^{k+1} \in \mathsf{ADJ}^{k+1}(\mathsf{a}^k)$ by calling this algorithm recursively. The base case is a d-cell, where the basis vectors are the Bernstein functions which span the Bernstein space assigned to the element.

2. Composite basis vectors are constructed by considering all possible sets of inclusion distances $\mathsf{INC}^{\mathsf{inc}_1,...,\mathsf{inc}_{d-k}}$ (section 8.3.2) and all possible alignment indices $\mathsf{i} \in \mathsf{ID}(\mathsf{a}^k)$ (section 8.3.3) on a^k. For each,

- Collect the set of $(k+1)$-cell basis vectors associated with the given set of inclusion distances and alignment index. (section 8.5.1 and appendix C.1)
- The index set of the new k-cell basis vector is the union of the index sets of this collection of $(k+1)$-cell basis vectors.

3. Each simple k-cell basis vector is constructed from a single $(k+1)$-cell basis vector on an adjacent $(k+1)$-cell, one for each $(k+1)$-cell basis vector whose index set is not a subset of a composite k-cell basis vector (section 8.5.2 and appendix C.2).
4. If desired, once the index set for each k-cell basis vector has been obtained, the constraint system can be solved to obtain the coefficient values and construct the basis vector.

8.5 Vertex basis vectors in two dimensions

We illustrate the algorithm by constructing two-dimensional basis vectors on vertices. The algorithm for arbitrary dimensions is found in appendix C. Examples of k-cell basis vectors of various dimension, on both two- and three-dimensional meshes are seen in figs. 26 to 32.

8.5.1 Composite vertex basis vectors

The *composite* basis vectors on a vertex v are formed from multiple adjacent edge basis vectors. Each composite vertex basis vector is associated with a set of inclusion distances $\mathsf{INC}^{\mathsf{inc}_1,\mathsf{inc}_2}$ (section 8.3.2) and alignment index $\mathsf{i} \in \mathsf{ID}(\mathsf{v})$ (section 8.3.3). In the case of a vertex, there is only one alignment index. The spokes (section 8.3.1) in this case consist of every edge-face pair adjacent to the vertex v. In fig. 23a we see a two-dimensional mesh with four elements. A set of spokes on the edges adjacent to the center vertex are depicted in fig. 23b. We assign an inclusion distance to every spoke adjacent to the vertex v, as described in section 8.3.2, beginning with two initial inclusion distance choices denoted inc_1 and inc_2 (corresponding to the spokes ρ_1 and ρ_2, respectively), that satisfy the requirement $0 \leq \mathsf{inc}_j \leq \vartheta^{\perp}_{\max}(\rho_j), j \in \{1,2\}$. The inclusion distances associated with every spoke adjacent to v, denoted $\mathsf{INC}^{\mathsf{inc}_1,\mathsf{inc}_2}$, are fixed by the choice of inc_1 and inc_2, as determined by the properties of inclusion distances outlined in section 8.3.2. One possible set of inclusion distances for the mesh in fig. 23a is shown in fig. 23c. The inclusion distance associated with a particular edge $\mathsf{e} \in \mathsf{ADJ}^1(\mathsf{v})$ is denoted by $\mathsf{INC}^{\mathsf{inc}_1,\mathsf{inc}_2}_{\mathsf{e}}$.

We place indexed submesh domains over each pair of elements adjacent to each edge adjacent to v (with their origins at v), and partition the mapped index sets of the basis vectors associated with each edge e into equivalence classes with respect to the parallel equivalence relation on the edge $\mathbf{BG}(\mathsf{e})/\varpi^{\parallel}_{\mathsf{e}}$. We then identify the equivalence classes for which the minimum projection onto the edge is less than or equal to $\mathsf{INC}^{\mathsf{inc}_1,\mathsf{inc}_2}_{\mathsf{e}}$:

$$\mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}) = \left\{ \mathbf{EBG}(\mathsf{e}) \in \mathbf{BG}(\mathsf{e})/\varpi^{\parallel}_{\mathsf{e}} \,:\, \min_{\mathsf{g}\in\mathsf{G}} \left(\pi^{\parallel}_{\mathsf{e}}(\mathsf{g})\right) \leq \mathsf{INC}^{\mathsf{inc}_1,\mathsf{inc}_2}_{\mathsf{e}}, \mathsf{G} \in \mathbf{EBG}(\mathsf{e}) \right\}. \tag{90}$$

An example of these equivalence classes is shown in fig. 24.

Let $\mathsf{e}^{\perp}_0 \in \mathbf{PC}(\mathsf{v},\mathsf{e})$, and $\mathsf{e}^{\perp}_1 \in \mathbf{PC}(\mathsf{v},\mathsf{e})$. ($\mathbf{PC}$ is defined in section 3.1.1.) We form the sets containing all indices whose associated cell is adjacent to e and whose associated submesh Greville point is a part of elements of $\mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}^{\perp}_0)$ and $\mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}^{\perp}_1)$

$$\mathsf{ID}^{\perp}_{\mathsf{e}^{\perp}_0} = \left\{ \mathsf{i} \in \mathsf{ID}(\mathsf{f}) \,:\, \mathsf{f} \in \mathsf{ADJ}^2(\mathsf{e}), \mathsf{g}(\mathsf{i}) \in \mathsf{G} \in \mathbf{EBG} \in \mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}^{\perp}_0) \right\}, \tag{91}$$

$$\mathsf{ID}^{\perp}_{\mathsf{e}^{\perp}_1} = \left\{ \mathsf{i} \in \mathsf{ID}(\mathsf{f}) \,:\, \mathsf{f} \in \mathsf{ADJ}^2(\mathsf{e}), \mathsf{g}(\mathsf{i}) \in \mathsf{G} \in \mathbf{EBG} \in \mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}^{\perp}_1) \right\}. \tag{92}$$

We form the set of Greville points that are fixed points of the parallel projectors onto $\mathsf{e}^{\perp}_0$ or $\mathsf{e}^{\perp}_1$:

$$\mathsf{G}^{\perp} = \left\{ \mathsf{g}(\mathsf{i}) : \mathsf{g}(\mathsf{i}) = \pi^{\perp}_{\mathsf{e}}(\mathsf{g}(\mathsf{i})), \mathsf{i} \in \mathsf{ID}^{\perp}_{\mathsf{e}^{\perp}_0} \right\} \cup \left\{ \mathsf{g}(\mathsf{i}) : \mathsf{g}(\mathsf{i}) = \pi^{\perp}_{\mathsf{e}}(\mathsf{g}(\mathsf{i})), \mathsf{i} \in \mathsf{ID}^{\perp}_{\mathsf{e}^{\perp}_1} \right\}. \tag{93}$$

We take all basis vectors whose projections onto the edges perpendicular to e lie in $\mathsf{G}^{\perp}$:

$$\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}) = \left\{ \mathsf{G} \in \mathbf{EBG} \in \mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}) \,:\, \forall_{\mathsf{g}\in\mathsf{G}}\, \pi^{\perp}_{\mathsf{e}}(\mathsf{g}) \subseteq \mathsf{G}^{\perp} \right\}. \tag{94}$$

In fig. 25, we see an example of these basis vector subsets marked with dotted lines.

The set of indices associated with $\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e})$ is

$$\mathsf{ID}_{\mathsf{e}}^{\mathsf{inc}_1,\mathsf{inc}_2} = \bigcup_{\mathsf{G}\in\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e})} \left\{\mathsf{i}\in\mathsf{ID}(\mathsf{f}) \,:\, \mathsf{f}\in\mathsf{ADJ}^2(\mathsf{e}), \mathsf{g}(\mathsf{i})\in\mathsf{G}\right\}. \tag{95}$$

The full set of indices associated with inc_0 and inc_1 is

$$\mathsf{ID}_{\mathsf{v}}^{\mathsf{inc}_1,\mathsf{inc}_2} = \bigcup_{\mathsf{e}\in\mathsf{ADJ}^1(\mathsf{v})} \mathsf{ID}_{\mathsf{e}}^{\mathsf{inc}_1,\mathsf{inc}_2}. \tag{96}$$

In the case of a vertex, there is only one alignment index, so the set $\mathsf{ID}_{\mathsf{v}}^{\mathsf{inc}_1,\mathsf{inc}_2}$ represents the index set of the composite basis vector. Note that for higher-dimensional cells, one additional step is required to find the set associated with a given alignment index and inclusion distances (see eq. (206) in appendix C.1).

We consider all possible values of inc_0 and inc_1 on v to construct the set of all composite vertex basis vectors. We use $\mathbf{BV}''(\mathsf{v})$ to represent this set. The full set of indices contained in this set is

$$\mathsf{UID}_{\mathsf{v}} = \bigcup_{\boldsymbol{n}\in\mathbf{BV}''(\mathsf{v})} \mathsf{ID}(\boldsymbol{n}). \tag{97}$$

An example of a set of composite vertex basis vectors is shown in fig. 26.

8.5.2 Simple vertex basis vectors

Each *simple* vertex basis vector is formed from a single edge basis vector on an adjacent edge, one for each edge basis vector whose index set is not subsumed by $\mathsf{UID}_{\mathsf{v}}$. We use $\mathbf{BV}'(\mathsf{v})$ to represent this set:

$$\mathbf{BV}'(\mathsf{v}) = \bigcup_{\mathsf{e}\in\mathsf{ADJ}^{k+1}(\mathsf{v})} \left\{\boldsymbol{n}\in\mathbf{BV}(\mathsf{e}) \,:\, \mathsf{ID}(\boldsymbol{n}) \not\subseteq \mathsf{UID}_{\mathsf{v}}\right\}. \tag{98}$$

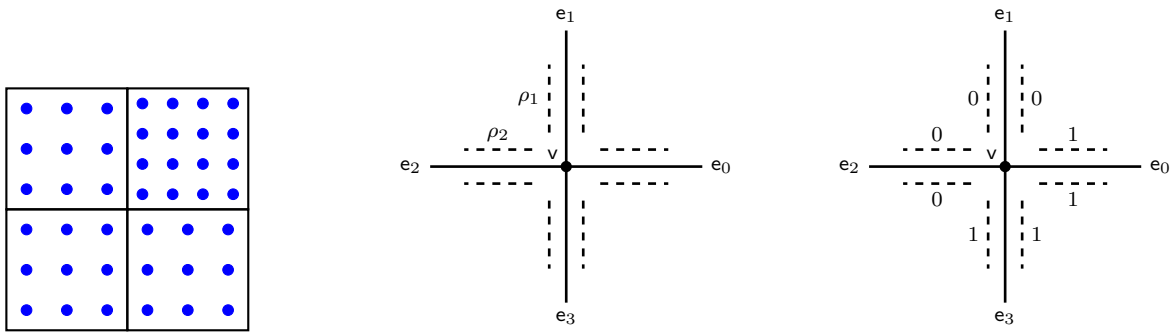
An example of a set of simple vertex basis vectors is shown in fig. 27.

8.5.3 The full set of vertex basis vectors

The full set of vertex basis vectors is found by combining the set of *composite* vertex basis vectors with the set of *simple* vertex basis vectors:

$$\mathbf{BV}(\mathsf{v}) = \mathbf{BV}''(\mathsf{v}) \cup \mathbf{BV}'(\mathsf{v}). \tag{99}$$

Once the index set for each vertex basis vector has been obtained, the constraint system can be solved to obtain the coefficient values and construct the basis vector.

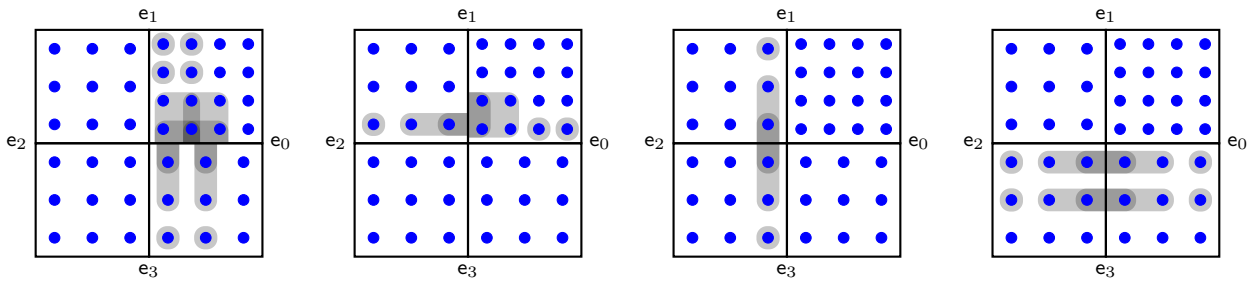


(a) A two-dimensional mesh with four cells. Three cells are biquadratic and one is bicubic. All edges are $\mathcal{C}^1$.

(b) We select an intitial element E on the top-left, and label the two initial spokes ρ_1 and ρ_2.

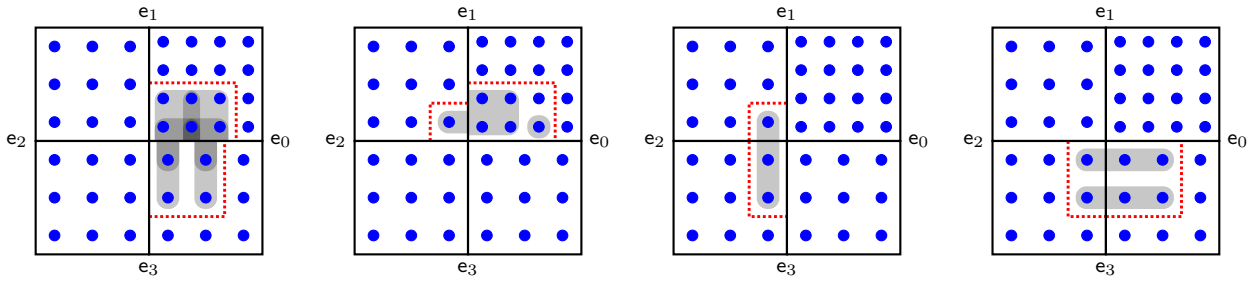
(c) We choose $\mathsf{inc}(\rho_1) = 0$ and $\mathsf{inc}(\rho_2) = 0$, thus determining the inclusion distances on all the spokes.

Figure 23: Choosing an initial element E and two initial spokes ρ_1 and ρ_2, and choosing the inclusion distances which uniquely identify a vertex basis vector. The fully constructed basis vector is seen in fig. 26b.



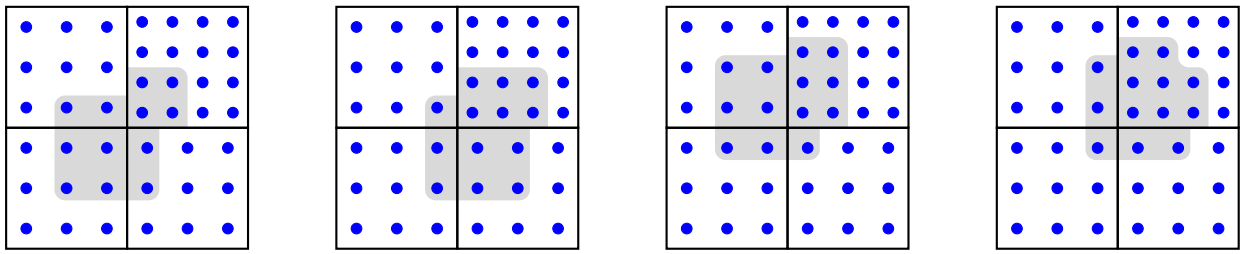
(a) $\mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}_0)$ (b) $\mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}_1)$ (c) $\mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}_2)$ (d) $\mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}_3)$

Figure 24: The basis vectors contained in $\mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}_i)$ for each edge $\mathsf{e}_i, i \in \{0,1,2,3\}$, the sets of equivalence classes for which the minimum projection onto the edge is less than or equal to $\mathsf{INC}^{\mathsf{inc}_1,\mathsf{inc}_2}_{\mathsf{e}_i}$. Overlapping regions are depicted with darker gray. The fully constructed basis vector is seen in fig. 26b.



(a) $\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}_0)$ (b) $\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}_1)$ (c) $\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}_2)$ (d) $\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}_3)$

Figure 25: The basis vectors contained in $\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\mathsf{inc}_2}(\mathsf{e}_i)$ for each edge $\mathsf{e}_i, i \in \{0,1,2,3\}$. The dotted line represents the span of indices whose projections onto the edges perpendicular to e_i lie in $\mathsf{G}^{\perp}$. Overlapping regions are depicted with darker gray. The fully constructed basis vector is seen in fig. 26b.



(a) The vertex basis vector with index support $\mathsf{ID}^{0,1}_{\mathsf{v}}$. (b) The vertex basis vector with index support $\mathsf{ID}^{0,0}_{\mathsf{v}}$. (c) The vertex basis vector with index support $\mathsf{ID}^{1,1}_{\mathsf{v}}$. (d) The vertex basis vector with index support $\mathsf{ID}^{1,0}_{\mathsf{v}}$.

Figure 26: The composite basis vectors on a vertex adjacent to four quadrilateral cells, three with degree $\mathsf{p} = (2,2)$ and one with degree $\mathsf{p} = (3,3)$. The continuity on the interfaces is $\mathcal{C}^1$.

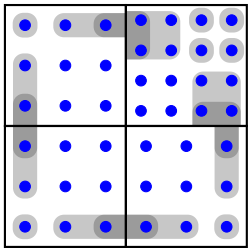


Figure 27: The simple basis vectors on a vertex adjacent to four quadrilateral cells, three with degree $\mathsf{p} = (2, 2)$ and one with degree $\mathsf{p} = (3, 3)$. The continuity on the interfaces is $\mathcal{C}^1$. Overlapping regions are depicted with darker gray. Each simple vertex basis vector is constructed from an adjacent edge basis vector that is not subsumed by a composite vertex basis vector.

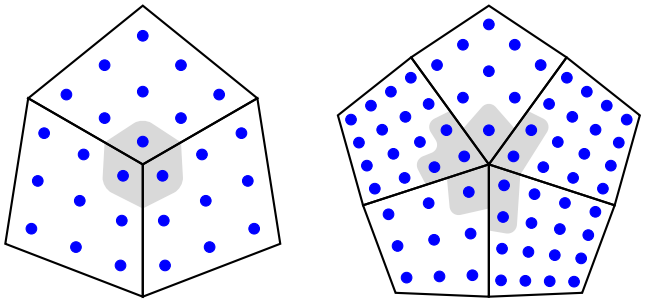


Figure 28: Examples of composite basis vectors on extraordinary vertices. On the left we see a valence-3 extraordinary vertex where all the cells have degree $\mathsf{p} = (2, 2)$. On the right we see a valence-5 extraordinary vertex where some cells have degree $\mathsf{p} = (2, 2)$ and others have degree $\mathsf{p} = (3, 3)$. In each case, the continuity of each edge is $\mathcal{C}^0$.

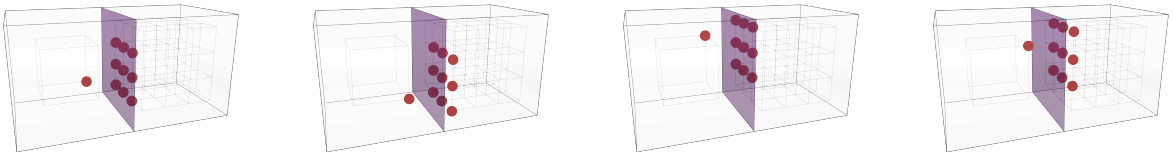


Figure 29: Basis vectors on a $\mathcal{C}^0$ interface between two hexahedron on a volumetric mesh. The element on the left is linear and the element on the right is cubic. This results in four alignment sets, and one interface-overlapping basis vector per alignment index.

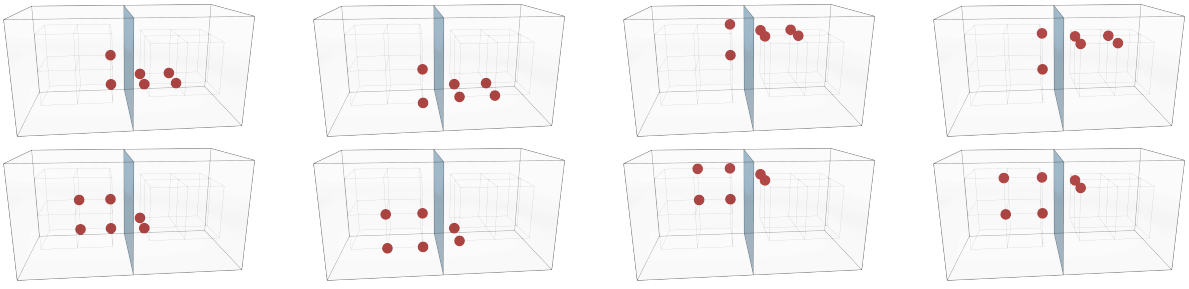


Figure 30: Basis vectors on a $\mathcal{C}^1$ interface between two hexahedron on a volumetric mesh. The two elements are each quadratic in the direction normal to the interface, and then are given degrees $(1, 2)$ and $(2, 1)$, respectively, in the directions parallel to the interface. This results in four alignment sets, and two interface-overlapping basis vectors per alignment index.

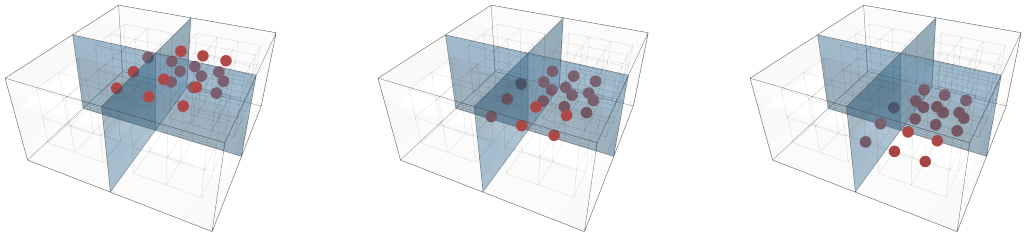
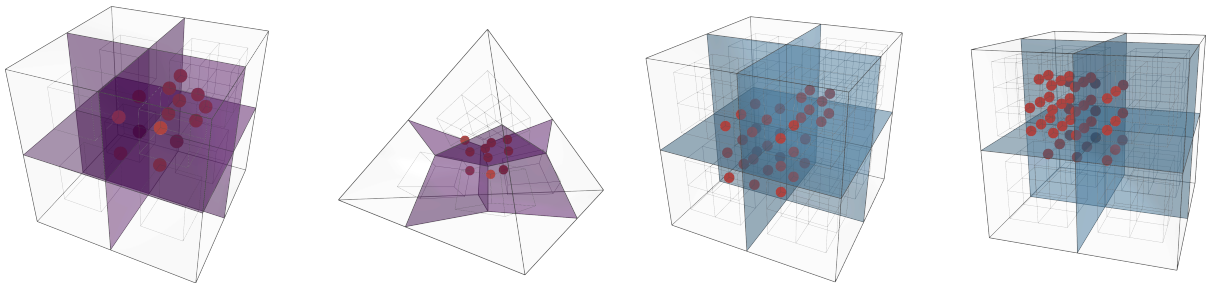


Figure 31: Examples of composite basis vectors on an edge adjacent to four hexahedron on a volumetric mesh. Three elements are quadratic and one is cubic. The continuity is $\mathcal{C}^1$ everywhere. These $(d-2)$-cell basis vectors are the volumetric analog to the two-dimensional vertex basis vector shown in fig. 26b, one for each alignment index along the edge.



(a) Seven linear and one quadratic hexahedron meet at a regular vertex. The continuity is $\mathcal{C}^0$ everywhere.

(b) Three linear and one quadratic hexahedron meet at an extraordinary vertex. The continuity is $\mathcal{C}^0$ everywhere.

(c) Seven quadratic and one cubic hexahedron meet at a regular vertex. The continuity is $\mathcal{C}^1$ everywhere.

(d) Seven quadratic and one cubic hexahedron meet at a regular vertex. The continuity is $\mathcal{C}^1$ everywhere.

Figure 32: Several examples of composite vertex basis vectors on volumetric meshes. The degree and continuity for each example is indicated. Figure 32b is an example of an extraordinary vertex on a volumetric mesh, constructed by filling a tetrahedron with hexahedral cells. Figure 32c is analogous to the two-dimensional example in fig. 26a. In fig. 32d notice the far corner Bernstein index is missing on the cubic cell, analogous to the two-dimensional example in fig. 26d.

8.6 Subordinate basis vectors

By definition (see eq. (34)), the constraint set for a k-cell c, $\mathsf{R}(\mathsf{c})$, is a superset of the constraint set of any higher dimensional cell having c in its boundary. Consequently, the basis vectors in $\mathbf{BV}(\mathsf{c})$ can be expressed locally in terms of the basis vectors associated with the $(k+1)$-cells having c in their shared boundary. Given $\boldsymbol{n} \in \mathbf{BV}(\mathsf{c})$, we define the set of $(k+1)$-cell basis vectors associated with $\boldsymbol{n}$ as

$$\mathbf{SBV}(\boldsymbol{n}) = \left\{ \boldsymbol{m} \in \left\{ \bigcup_{\mathsf{a} \in \mathsf{ADJ}^{k+1}(\mathsf{c})} \mathbf{BV}(\mathsf{a}) \right\} \,:\, \mathsf{ID}(\boldsymbol{m}) \subseteq \mathsf{ID}(\boldsymbol{n}) \right\} \tag{100}$$

and say that the members of $\mathbf{SBV}(\boldsymbol{n})$ are subordinate basis vectors or basis vectors subordinate to $\boldsymbol{n}$. The subordinate basis vectors of a set of basis vectors is also defined as $\mathbf{SBV}(\mathbf{BV}(\mathsf{c})) = \bigcup_{\boldsymbol{n} \in \mathbf{BV}(\mathsf{c})} \mathbf{SBV}(\boldsymbol{n})$.

We also define the subset of basis vectors subordinate to $\boldsymbol{n}$ that are also basis vectors on an adjacent cell c as

$$\mathbf{SBV}_{\mathsf{c}}(\boldsymbol{n}) = \mathbf{SBV}(\boldsymbol{n}) \cap \mathbf{BV}(\mathsf{c}). \tag{101}$$

An example of subordinate basis vectors in one dimension are shown in fig. 33.

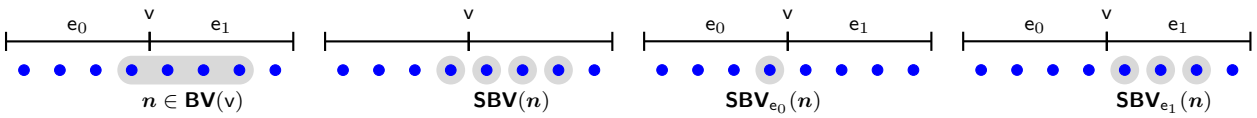


Figure 33: An example of subordinate basis vectors in one dimension. The set of subordinate basis vectors of $\boldsymbol{n}$ is $\mathbf{SBV}(\boldsymbol{n})$, and $\mathbf{SBV}_{\mathsf{e}_i}(\boldsymbol{n}), i \in \{0,1\}$ are the subsets of subordinate basis vectors associated with edges e_0 and e_1, respectively.

8.7 Basis vector boundaries

The U-spline algorithm relies on relationships between basis vectors associated with adjacent topological features in order to construct basis vectors from the global set. These relationships are most conveniently expressed by introducing a notion of boundary to the mapped indices associated with nonzero entries in the basis vector.

It has been shown that basis vectors can be represented in terms of the basis vectors associated with higher-dimensional adjacent cells. Having constructed a basis vector, $\boldsymbol{n} \in \mathbf{BV}(\mathsf{a}^k)$, for a k-cell, a^k, we loosely define the boundary with respect to an adjacent $(k+1)$-cell, b^{k+1}, as the most distant elements chosen from the set of subordinate basis vectors $\mathbf{SBV}_{\mathsf{b}^{k+1}}(\boldsymbol{n})$. We will denote this set by $\mathbf{BD}_{\mathsf{b}^{k+1}}(\boldsymbol{n})$ and make the definition more precise. We first give definitions for the three cases relevant to one and two dimensions and then define basis vector boundaries in arbitrary dimension.

8.7.1 Basis vector boundaries in one dimension

For the case of a basis vector $\boldsymbol{n}_{\mathsf{v}} \in \mathbf{BV}(\mathsf{v})$ associated with a vertex v in the one-dimensional setting, we begin by defining an indexed submesh domain over the two edges adjacent to the vertex, such that the origin lies at the vertex, and the associated index mappings are $\boldsymbol{\phi}^{\mathsf{e}}$ for each edge e. Then, the boundary with respect to the edge e is given by

$$\mathbf{BD}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}}) = \left\{ \boldsymbol{q} \in \mathbf{SBV}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}}) \;:\; \max_{\mathsf{i} \in \mathsf{ID}(\boldsymbol{q})} \left\| \boldsymbol{\phi}^{\mathsf{e}}(\mathsf{i}) \right\|_2 = q_{\max} \right\} \tag{102}$$

where

$$q_{\max} = \max_{\boldsymbol{q} \in \mathbf{SBV}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}})} \max_{\mathsf{i} \in \mathsf{ID}(\boldsymbol{q})} \left\| \boldsymbol{\phi}^{\mathsf{e}}(\mathsf{i}) \right\|_2 . \tag{103}$$

In fig. 34 we see a basis vector $\boldsymbol{n}_{\mathsf{v}}$ on a vertex on a one-dimensional mesh. Associated with each adjacent edge e_1 and e_2 is a boundary set, $\mathbf{BD}_{\mathsf{e}_1}(\boldsymbol{n}_{\mathsf{v}})$ and $\mathbf{BD}_{\mathsf{e}_2}(\boldsymbol{n}_{\mathsf{v}})$, which contains the subordinate basis vector that makes up the boundary in the direction of the respective edge. These are represented as black circles. The full set of boundaries is obtained by taking the union of all boundary sets generated by the edges adjacent to v:

$$\mathbf{BD}(\boldsymbol{n}_{\mathsf{v}}) = \bigcup_{\mathsf{e} \in \mathsf{ADJ}^1(\mathsf{v})} \mathbf{BD}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}}). \tag{104}$$

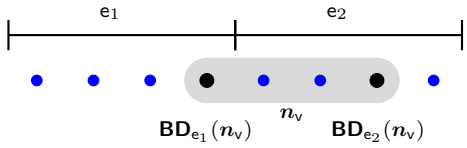


Figure 34: An example of basis vector boundaries on a vertex null vector on a one-dimensional mesh.

8.7.2 Basis vector boundaries in two dimensions

In a two-dimensional setting we must consider the boundaries of edge basis vectors and vertex basis vectors. The boundaries of edge basis vectors are constructed from the basis vectors of the adjacent face systems, which are just the standard Bernstein basis functions. Given an edge $\mathsf{e} \in \mathsf{ADJ}^1(\mathsf{f})$ with basis vector $\boldsymbol{n}_{\mathsf{e}} \in \mathbf{BV}(\mathsf{e})$

adjacent to a face f, the subordinate subset is $\mathbf{SBV}_{\mathsf{f}}(\boldsymbol{n}_{\mathsf{e}})$ and the associated index set is $\mathsf{ID}(\mathbf{SBV}_{\mathsf{f}}(\boldsymbol{n}_{\mathsf{e}}))$. We form the mapped points associated with $\mathsf{ID}(\mathbf{SBV}_{\mathsf{f}}(\boldsymbol{n}_{\mathsf{e}}))$ and partition it with respect to the projection onto the edge e:

$$\mathbf{G}^{\parallel} = \left\{\boldsymbol{\phi}^{\mathsf{f}}(\mathsf{i}), \mathsf{i} \in \mathsf{ID}(\mathbf{SBV}_{\mathsf{f}}(\boldsymbol{n}_{\mathsf{e}}))\right\} / \varpi_{\mathsf{e}}^{\parallel}. \tag{105}$$

The boundary set is formed by taking the index associated with the most distant point in each equivalence class:

$$\mathbf{BD}_{\mathsf{f}}(\boldsymbol{n}_{\mathsf{e}}) = \bigcup_{\mathsf{G} \in \mathbf{G}^{\parallel}} \left\{ \mathsf{i} \in \mathsf{ID}(\mathbf{SBV}_{\mathsf{f}}(\boldsymbol{n}_{\mathsf{e}})) \,:\, \pi_{\mathsf{e}}^{\perp}(\boldsymbol{\phi}^{\mathsf{f}}(\mathsf{i})) = \max_{\mathsf{g} \in \mathsf{G}} \pi_{\mathsf{e}}^{\perp}(\mathsf{g}) \right\}. \tag{106}$$

Again, the full boundary is given by

$$\mathbf{BD}(\boldsymbol{n}_{\mathsf{e}}) = \bigcup_{\mathsf{f} \in \mathsf{ADJ}^2(\mathsf{e})} \mathbf{BD}_{\mathsf{f}}(\boldsymbol{n}_{\mathsf{e}}). \tag{107}$$

The boundary set for a vertex basis vector $\boldsymbol{n}_{\mathsf{v}}$ in the two-dimensional setting is formed from the basis vectors associated with an edge e adjacent to the vertex v. The construction is similar to the construction presented for the boundary of edge basis vectors. We form the set containing the set of mapped index points for each vector in $\mathbf{SBV}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}})$:

$$\mathbf{G}(\mathbf{SBV}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}})) = \left\{ \left\{ \boldsymbol{\phi}^{\mathsf{e}}(\mathsf{i}), \mathsf{i} \in \mathsf{ID}(\boldsymbol{n}_{\mathsf{e}}) \right\}, \boldsymbol{n}_{\mathsf{e}} \in \mathbf{SBV}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}}) \right\}. \tag{108}$$

The chart $\boldsymbol{\phi}^{\mathsf{e}}$ is chosen so that both elements adjacent to the edge e are mapped consistently and that v lies at the origin. We partition the set of Greville point sets into equivalence classes with respect to the projection perpendicular to the edge e and form the boundary set by taking the most distant element from each equivalence class. Given an equivalence class $\mathsf{H} \in \mathbf{G}(\mathbf{SBV}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}}))/\varpi_{\mathsf{e}}^{\perp}$, we define the maximum of this equivalence class as the set of points whose projection onto the line in the direction of e is greatest:

$$\max \mathsf{H} = \mathsf{G} \in \mathsf{H} \,:\, \max_{\mathsf{g} \in \mathsf{G}} \pi_{\mathsf{e}}^{\parallel}(\mathsf{g}) = \max_{\mathsf{G}' \in \mathsf{H}} \max_{\mathsf{g}' \in \mathsf{G}'} \pi_{\mathsf{e}}^{\parallel}(\mathsf{g}'), \tag{109}$$

$$\mathbf{BD}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}}) = \left\{ \boldsymbol{n}_{\mathsf{e}} \in \mathbf{SBV}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}}) \,:\, \boldsymbol{\phi}^{\mathsf{e}}(\mathsf{ID}(\boldsymbol{n}_{\mathsf{e}})) = \max \mathsf{H}, \boldsymbol{\phi}^{\mathsf{e}}(\mathsf{ID}(\boldsymbol{n}_{\mathsf{e}})) \in \mathsf{H}, \mathsf{H} \in \mathbf{G}(\mathbf{SBV}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}}))/\varpi_{\mathsf{e}}^{\perp} \right\}. \tag{110}$$

In fig. 35 a vertex basis vector on a two-dimensional mesh with uniform degree is seen on the left. An equivalence class with respect to the projection perpendicular to the edge e contains two subordinate basis vectors, as seen in the middle. The rightmost subordinate basis vector from the equivalence class makes up the basis vector boundary, as seen on the right. Similarly, in fig. 36 on the left we see a vertex basis vector on a two-dimensional mesh with mixed degree. In this case there are two equivalence classes with respect to the projection perpendicular to the edge e, each of which contain two subordinate basis vectors. The boundary set is made up of the rightmost subordinate basis vectors from each equivalence class, as seen on the right. The full boundary is once again obtained by uniting the boundary sets associated with every edge adjacent to v

$$\mathbf{BD}(\boldsymbol{n}_{\mathsf{v}}) = \bigcup_{\mathsf{e} \in \mathsf{ADJ}^1(\mathsf{v})} \mathbf{BD}_{\mathsf{e}}(\boldsymbol{n}_{\mathsf{v}}). \tag{111}$$

8.7.3 Basis vector boundaries in arbitrary dimensions

The boundary set for a k-cell basis vector $\boldsymbol{n}_{\mathsf{a}^k} \in \mathbf{BV}(\mathsf{a}^k)$ in the d-dimensional setting is formed from the basis vectors associated with a $(k+1)$-cell b^{k+1} adjacent to a^k, $\mathsf{b}^{k+1} \in \mathsf{ADJ}^{k+1}(\mathsf{a}^k)$. The description for constructing the boundary set for a basis vector in arbitrary dimensions is a direct generalization of the description in section 8.7.2 for vertex basis vectors in two dimensions, by taking eqs. (108) to (111) and replacing the vertex v with a^k and the edge e with b^{k+1}. In the general description, the chart $\boldsymbol{\phi}^{\mathsf{b}^{k+1}}$ is chosen so that all elements adjacent to b^{k+1} are mapped consistently and that one of the vertices adjacent to a^k lies at the origin.

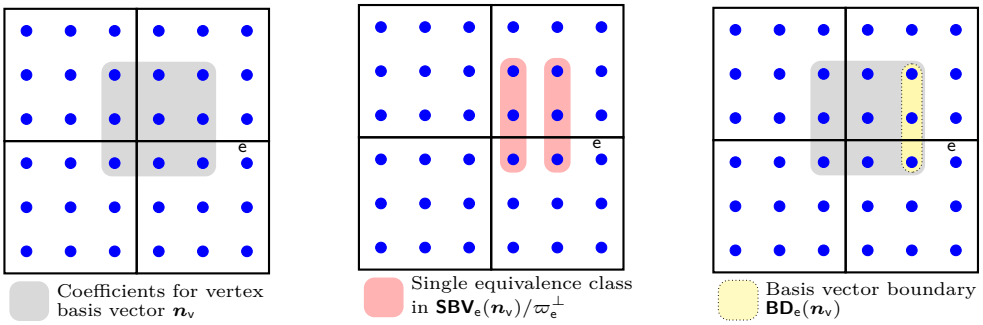


Figure 35: The boundary of a vertex basis vector $\boldsymbol{n}_\mathsf{v}$ on edge e, on a quadratic mesh with $\mathcal{C}^1$ continuity. In this case, there is only one equivalence class, which contains two subordinate basis vectors. The rightmost subordinate basis vector from the equivalence class makes up the basis vector boundary.

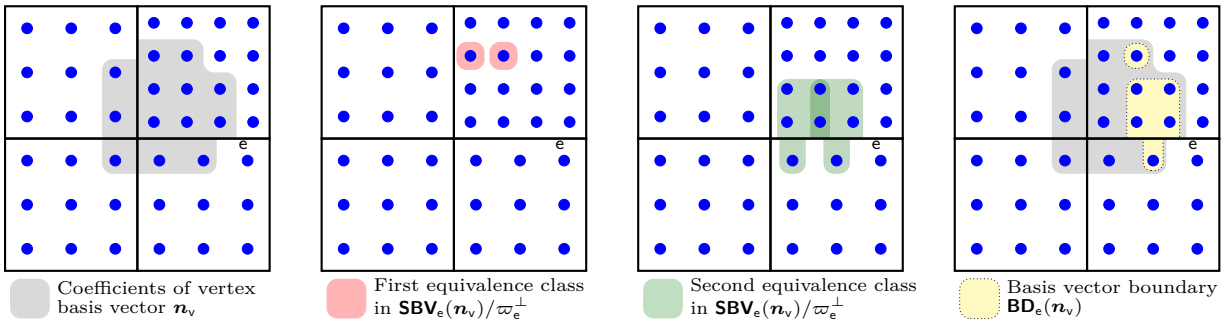


Figure 36: In the presence of local variation in degree, the boundary of a vertex basis vector $\boldsymbol{n}_\mathsf{v}$ on edge e may be determined from more than one equivalence class. The rightmost subordinate basis vectors from each equivalence class make up the basis vector boundary. All the interfaces have $\mathcal{C}^1$ continuity.

9 The U-spline mesh

As described previously in section 6, a spline space can be constructed directly from the nullspace of the global constraint matrix $\boldsymbol{R}(\mathbf{B})$. However, for large meshes, analyzing and constructing a basis for the global nullspace is usually not computationally feasible.

To overcome this issue, we will instead seek to build the index sets corresponding to k-cell basis vectors as described in section 8 and arrange them into basis vectors for U-spline basis functions as described in section 10. The nullspace problem associated with each U-spline basis function will be rank one in all cases.

To simplify the construction of a U-spline basis, we define a class of *admissible* Bézier meshes, which we call U-spline meshes and denote by $\mathbf{U}$. A U-spline mesh is a Bézier mesh with admissibility constraints placed on the layout of cells, the degree, and the smoothness of interfaces. Admissibility constraints are imposed through appropriate *separation* and *grading* conditions on degree and continuity transitions throughout the Bézier mesh. The mathematical properties of the corresponding admissible U-spline space $\mathcal{U}$ can be controlled *a priori* by specifying the properties of the underlying Bézier mesh topology.

Specifically, admissibility ensures that the following properties are satisfied:

- The index sets of k-cell basis vectors can be constructed directly through topological equivalence relations based on the derivative ordering property of a Bernstein-like basis on each cell and mesh topology local to the k-cell as described in section 8,
- The basis vector that defines each U-spline basis function can be determined from the relationships between a set of k-cell basis vectors to determine the indices of nonzero values. Coefficient values can then be determined by solving a relatively small rank one nullspace problem. The details of this approach are described in section 8 and section 10.1.
- The detailed mathematical properties satisfied by the U-spline space are described in section 11.

The admissibility conditions are written in terms of *ribbons*. A ribbon r is an ordered set of interfaces from a Bézier mesh segment length is controlled by both the degree and continuity of adjacent elements and interfaces, respectively. Ribbons are used as a measuring instrument on a Bézier mesh to quantify the separation distance between local variations in degree and continuity.

We note, however, that we have studied U-spline basis functions constructed over Bézier meshes that are more complex than those presented here and plan to present more generalized admissibility conditions in a forthcoming work. We anticipate that certain admissibility requirements will always be necessary to construct spline spaces with desirable mathematical properties.

9.1 Ribbons

A ribbon $\mathsf{r} = \left\{\mathsf{I}^h, \mathsf{o}, \boldsymbol{t}\right\}$ where the *tail* $\boldsymbol{t} = \left[\mathsf{I}_0, \mathsf{I}_1, ..., \mathsf{I}_{|\boldsymbol{t}|-1}\right]$, is composed of $|\boldsymbol{t}|$ consecutive interfaces I_j, originating at an *origin* $(d-2)$-cell o, and I^h is the *head* interface that is opposite the tail $\boldsymbol{t}$. The origin $(d-2)$-cell o must be regular and interior to the mesh. We say a ribbon with $|\boldsymbol{t}|$ interfaces in the tail is a ribbon of length $|\boldsymbol{t}|$. The skeleton of a ribbon, denoted by skel(r), is the array of $|\boldsymbol{t}|+1$ $(d-2)$-cells which are attached to the $|\boldsymbol{t}|$ interfaces in the tail and parallel to the origin $(d-2)$-cell o, including the origin $(d-2)$-cell. The fact that the definition of a ribbon is defined using $(d-2)$-cells prevents a meaningful definition of ribbons for univariate U-splines. Fortunately, all univariate U-splines of maximal continuity are admissible and so this is not a problem.

Figure 37 illustrates a ribbon composed of several consecutive interfaces in both the two- and three-dimensional mesh cases. A small solid circle or sphere near the head of the ribbon represents an initial Bernstein coefficient, adjacent to which is the origin $(d-2)$-cell o. The interfaces in the tail extend to the right. The head of the ribbon I^h is the interface immediately opposite the tail.

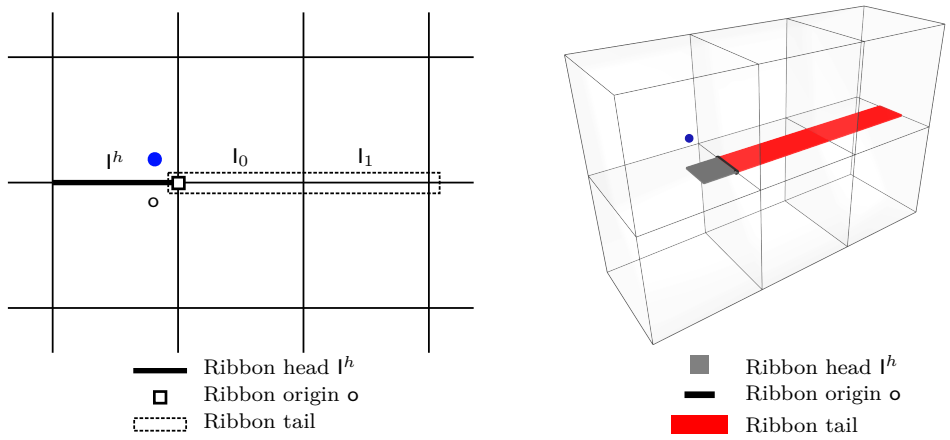


Figure 37: An example of a ribbon on a two-dimensional mesh (left) and on a three-dimensional mesh (right). In each case, a small solid circle or sphere near the head of the ribbon represents an initial Bernstein coefficient.

9.1.1 Maximum coupling length

A ribbon r can originate from any interior regular $(d-2)$-cell in a U-spline mesh, and can be of any length $|\boldsymbol{t}|, \boldsymbol{t} \in \mathsf{r}$. However, we will primarily use ribbons to measure the maximum distance, measured by the length of the ribbon, between a specified Bernstein coefficient and any other coefficient coupled to it. We call a ribbon constructed in this way a *ribbon of maximum coupling length*. In this case, the length of the tail is determined by how far that coefficient can couple with neighboring coefficients in the direction of an interface which is adjacent to the origin $(d-2)$-cell (this becomes the first interface in the tail). Algorithm 2 in appendix D describes the procedure used to determine the interfaces in the tail of a ribbon of maximum coupling length. A ribbon of maximum coupling length is said to be *truncated* if the length of the ribbon is shortened due to reduced continuity on the final interface encountered by the tail.

See appendix A.4 for an intuitive example of constructing a ribbon of maximum coupling length.

9.1.2 Continuity transitions

A *continuity transition ribbon* (designated by cr) is a ribbon of maximum coupling length with the additional property that $\vartheta^{\mathsf{l}^h} > \vartheta^{[\boldsymbol{t}]_0}$. An example of this type of ribbon for both two and three dimensions is seen in fig. 38. Two perpendicular continuity transition ribbons cr_i and cr_j where $i \neq j$ and cr_i is length $|\boldsymbol{t}_i|$ and cr_j is length $|\boldsymbol{t}_j|$, are said to be intersecting if they share a $(d-2)$-cell (i.e., $\text{skel}(\mathsf{cr}_i) \cap \text{skel}(\mathsf{cr}_j) \neq \varnothing$). A tail-to-tail intersection occurs when the shared $(d-2)$-cell is $\mathsf{w}_{|\boldsymbol{t}_i|+1}$ in cr_i and $\mathsf{w}_{|\boldsymbol{t}_j|+1}$ in cr_j. A head-to-tail intersection occurs when the shared $(d-2)$-cell is w_0 in cr_i and $\mathsf{w}_{|\boldsymbol{t}_j|+1}$ in cr_j, or vice-versa. The minimum perpendicular degree of a continuity transition ribbon is defined as $p^{\perp}_{\min}(\mathsf{cr}) = \min_k(p^{\perp}_{\min}([\boldsymbol{t}]_k))$.

9.1.3 Degree transitions

A *degree transition ribbon* (designated by dr) is a ribbon of maximum coupling length with the additional property that $p^{\perp}_{\min}(\mathsf{l}^h) < p^{\perp}_{\min}([\boldsymbol{t}]_0)$. An example of this type of ribbon for both two and three dimensions is seen in fig. 39.

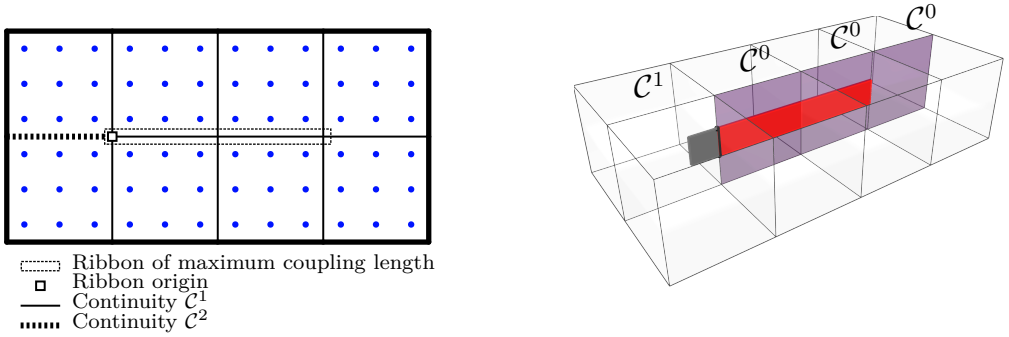


Figure 38: Continuity transition ribbons on a two-dimensional mesh (left) and a three-dimensional mesh (right). In each case, the ribbon originates at a $(d-2)$-cell where a continuity transition occurs, and proceeds in the direction of lower continuity. All elements on both meshes are quadratic.

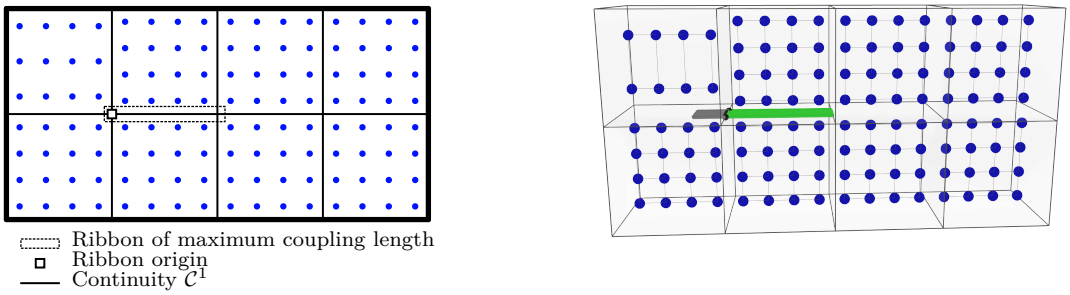


Figure 39: Degree transition ribbons on a two-dimensional mesh (left) and a three-dimensional mesh (right). A degree transition ribbon is a ribbon of maximum coupling length with the additional property that $p^{\perp}_{\min}(\mathsf{l}^h) < p^{\perp}_{\min}([\boldsymbol{t}]_0)$.

9.2 Admissible layouts

As described previously, a U-spline mesh $\mathbf{U}$ is a Bézier mesh with an admissible layout. The admissibility conditions presented here were selected for simplicity, while still ensuring that the resulting U-spline spaces possess the requisite mathematical properties. It should be noted, however, that various generalizations of these conditions exist but are significantly more complex to implement and understand so are omitted from this work. We will explore these generalized conditions in a forthcoming work. For our purposes, an admissible layout satisfies the following three simple constraints:

- **ϑ-separation:** If two perpendicular continuity transition ribbons cr_i and cr_j intersect (that is, share a common $(d-2)$-cell), then it must be a tail-to-tail intersection or a head-to-tail intersection. If a truncated continuity transition ribbon meets with a non-truncated continuity transition ribbon tail-to-tail, then $\vartheta^{\mathsf{l}_i^h} < \mathcal{C}^\infty$ or $\vartheta^{\mathsf{l}_j^h} < \mathcal{C}^\infty$. If two truncated continuity transition ribbons meet tail-to-tail, then $\vartheta^{\mathsf{l}_i^h} < \mathcal{C}^\infty$ and $\vartheta^{\mathsf{l}_j^h} < \mathcal{C}^\infty$.

- **p-separation:** For all continuity transition ribbons cr_i, if $p^\perp_{\min}(\mathsf{l}^h) > \vartheta^{\mathsf{l}^h}$, then $p^\perp_{\min}(\mathsf{cr}_i) > \vartheta^{\mathsf{l}^h}$ and if $p^\perp_{\min}(\mathsf{l}^h) = \vartheta^{\mathsf{l}^h}$, then $p^\perp_{\min}(\mathsf{cr}_i) \geq \vartheta^{\mathsf{l}^h}$. Also, no degree transition ribbon dr perpendicular to cr_i may intersect o_i, the origin of cr_i.

- **ϑ-grading:** A *creased* $(d-2)$*-cell* is any $(d-2)$-cell in a Bézier mesh where all adjacent interfaces are creased to $\mathcal{C}^0$ or $\mathcal{C}^{-1}$. For a creased $(d-2)$-cell w and a set of continuity transition ribbons $\{\mathsf{cr}_j\}$ that all terminate at w, and given any $\mathsf{cr} \in \{\mathsf{cr}_j\}$ of length $|\boldsymbol{t}|$ where $\mathsf{w}_j = \mathsf{ADJ}^{d-2}(\mathsf{l}_j) \cap \mathsf{ADJ}^{d-2}(\mathsf{l}_{j-1})$, $\mathsf{l}_j \in \mathbf{t} \in \boldsymbol{cr}$, then $\vartheta^{\mathsf{l}_j} \leq \vartheta_j$, where ϑ_j is defined recursively as $\vartheta_{|\boldsymbol{t}|-1} = 0$ and $\vartheta_{j-1} = \vartheta_j + \beta_j$, $j = |\boldsymbol{t}| - 1, \ldots, 1$ where

$$\beta_j = \begin{cases} 0 & \text{if } \vartheta^\perp_{\max}(\mathsf{w}_j) = \mathcal{C}^\infty \\ 1 & \text{otherwise} \end{cases} \tag{112}$$

  and

$$\vartheta^\perp_{\max}(\mathsf{w}_j) = \max_{\mathsf{l}^\perp \in \mathsf{ADJ}^{d-1}(\mathsf{w}_j) \setminus \{\mathsf{l}_j, \mathsf{l}_{j-1}\}} \vartheta^{\mathsf{l}^\perp}. \tag{113}$$

  Additionally, on three-dimensional meshes, given three edges on a hexahedral element E adjacent to vertex $\mathsf{v} \in \mathsf{ADJ}^0(\mathsf{E})$, (without loss of generality we label these edges $\{\mathsf{e}_0, \mathsf{e}_1, \mathsf{e}_2\} = \mathsf{ADJ}^1(\mathsf{E}) \cap \mathsf{ADJ}^1(\mathsf{v})$), if e_0 is extraordinary, then all faces adjacent to v and perpendicular to e_1 and e_2, i.e.,

$$\mathsf{f} \in \bigcup_{\mathsf{e}' \in \{\mathsf{e}_1, \mathsf{e}_2\}} \mathsf{ADJ}^2(\mathsf{v}) \cap \mathsf{ADJ}^2 \circ \mathsf{ADJ}^d(\mathsf{e}') \setminus \mathsf{ADJ}^2(\mathsf{e}'), \tag{114}$$

  must be creased. This three-dimensional requirement is illustrated in fig. 45b. Note that all extraordinary $(d-2)$-cells are required to be creased, but regular $(d-2)$-cells may also be creased as seen in fig. 41.

Examples of two-dimensional mesh layouts that satisfy the separation and grading admissibility conditions are shown in figs. 40 and 41 and three-dimensional examples are shown in figs. 42 to 45.

The ϑ-separation condition is demonstrated in fig. 40a and similarly in figs. 42a and 43 where we see admissible configurations with ribbons that do not intersect except tail-to-tail or head-to-tail. We note that ribbons on volumetric meshes that cross in the middle of a face are not considered intersecting, and are admissible.

A common application of the p-separation condition is seen in figs. 40c and 42b where a transition to lower continuity must be sufficiently separated from a transition to lower perpendicular degree. Also notable is the p-separation condition demonstrated in figs. 40b and 44 where a degree transition ribbon cannot be permitted to intersect the origin of a continuity transition.

The ϑ-grading condition is demonstrated in Figures 41 and 45 where we see several mesh configurations that include a creased $(d-2)$-cell. Extraordinary $(d-2)$-cells are always required to be creased, but the same ϑ-grading conditions also apply to the interfaces near a regular $(d-2)$-cell if all adjacent interfaces are creased. Unstructured volumetric configurations often contain many extraordinary edges, as the hex-meshed tetrahedral topology demonstrates in fig. 45a. When an extraordinary edge is near a boundary on a volumetric mesh, sometimes additional faces must be creased as is demonstrated in fig. 45b. In this case, an additional

face to the left of the extraordinary edge was required to be creased (despite itself being adjacent to only regular edges). This is because this face is perpendicular to an edge which, in turn, is perpendicular to the extraordinary edge (see eq. (114)).

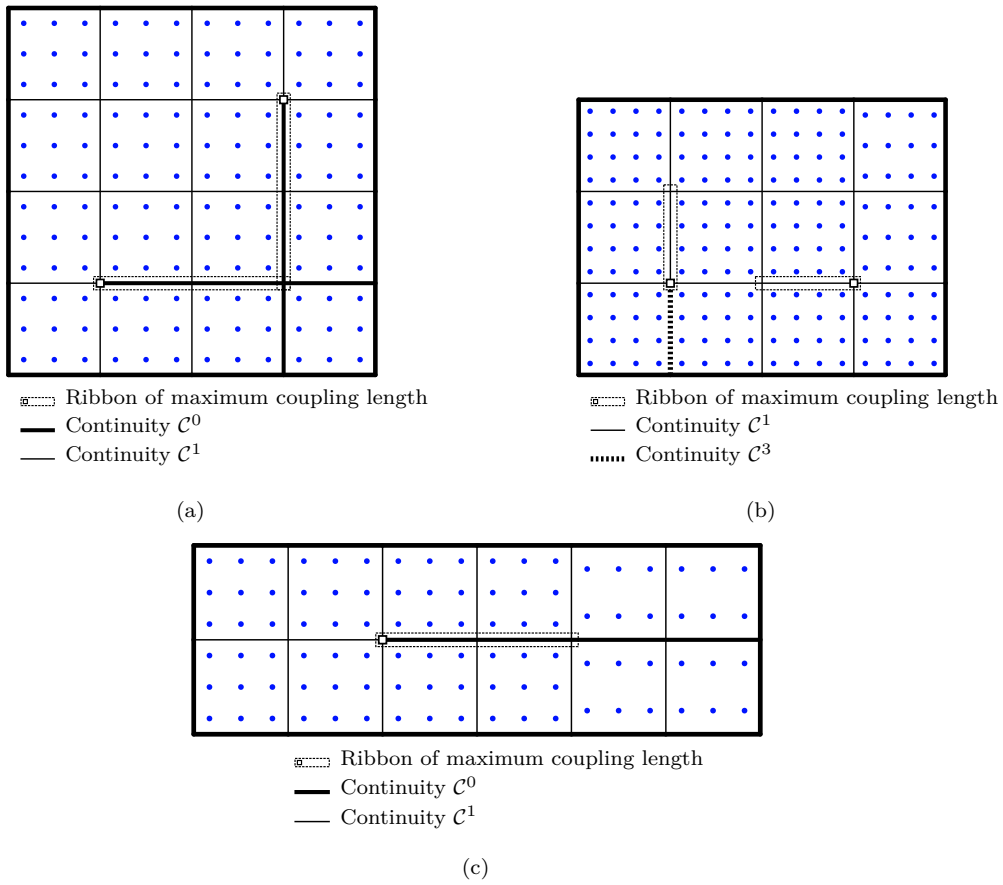


Figure 40: Examples of admissible layouts that satisfy ϑ- and p-separation conditions. On the top left, two continuity transition ribbons meet tail-to-tail. On the top right, a degree transition ribbon dr is sufficently separated from a continuity transition ribbon cr to avoid intersection with the origin of cr. On the bottom, a degree transition is sufficiently separated from the head of a continuity transition ribbon cr so that $p^{\perp}_{\min}(\mathsf{cr}) > \vartheta^{l^h}$.

9.3 Classification

We denote a *class* of U-spline meshes by $\boldsymbol{\mathcal{U}}$, with a superscript that is used to identify key Bézier mesh properties which are common to the meshes contained in the class (see table 1). Using this superscript notation, U-spline meshes sharing certain characteristics can be grouped and denoted by, for example $\boldsymbol{\mathcal{U}}^{RHKP}$, which would denote all structured U-spline meshes where variations in mesh size, smoothness, and degree propagate globally. Examples include the tensor product splines such as B-splines and NURBS. The U-spline class $\boldsymbol{\mathcal{U}}^{rHKP}$ may not admit a global parameterization due to the possible presence of, for example, extraordinary vertices, but all variations in mesh size, smoothness, and degree propagate globally. Spline meshes in this class include unstructured finite element meshes composed of linear elements and multi-patch NURBS meshes. Analysis-suitable T-spline (ASTS) meshes are in $\boldsymbol{\mathcal{U}}^{rhkP}$. We note that in one dimension, the most unstructured class possible is $\boldsymbol{\mathcal{U}}^{RhKP}$.

In table 1, this superscript notation is related to underlying Bézier mesh characteristics.

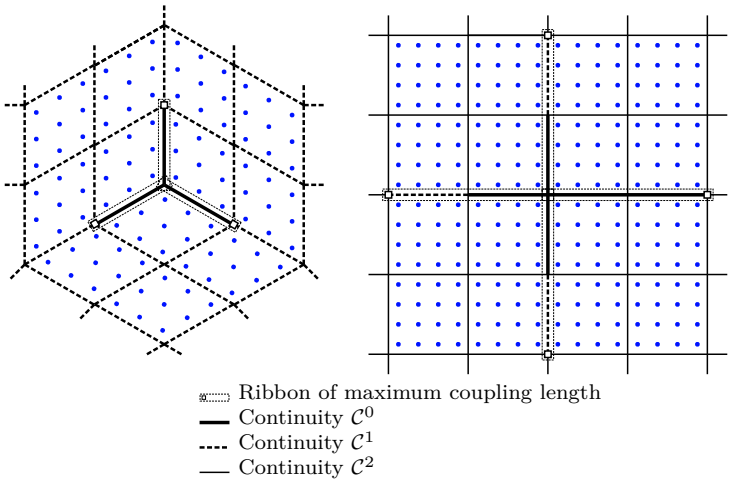
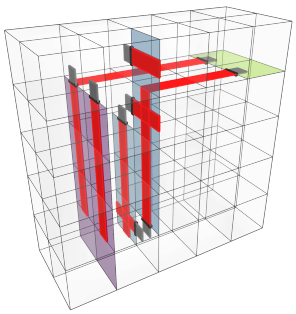
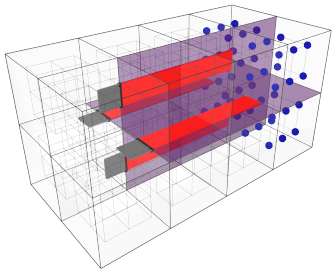


Figure 41: Example of two admissible layouts that satisfy the ϑ-grading condition. Note that while extraordinary $(d-2)$-cells will always be creased, a creased $(d-2)$-cell may also be regular as shown in the example on the right.

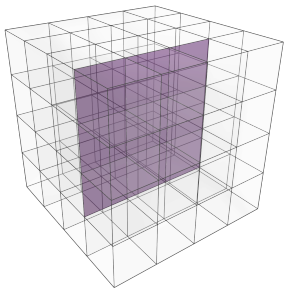


(a) Example of an admissible cubic volumetric mesh that satisfies the ϑ-separation condition. The marked interfaces on the left are $\mathcal{C}^0$, the middle are $\mathcal{C}^1$, and the right are $\mathcal{C}^3$. All other interfaces are $\mathcal{C}^2$.

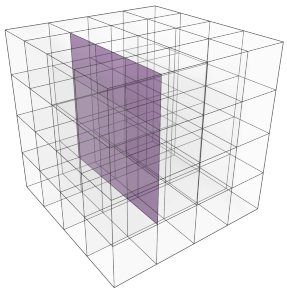


(b) An admissible mesh that satisfies the p-separation condition. The marked interfaces on the right are $\mathcal{C}^0$, and the others are $\mathcal{C}^1$. This mesh is quadratic everywhere except for the cells on the far end, which are set to $p = (2, 1, 1)$, as depicted by the small solid spheres.

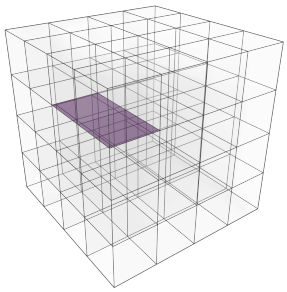
Figure 42: Examples of admissible volumetric meshes that satisfy ϑ- and p-separation conditions. The example on the left contains both a tail-to-tail and head-to-tail ribbon intersection. Ribbons that cross at the center of a face are not considered intersecting and are admissible. The example on the right contains a continuity transition from $\mathcal{C}^1$ to $\mathcal{C}^0$, sufficiently far from the degree transition to prevent the continuity transition ribbons from overlapping the cells with lower degree.



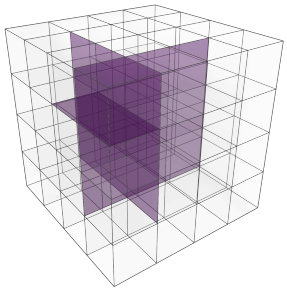
(a) A subset of faces creased to $\mathcal{C}^0$ are marked.



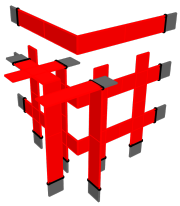
(b) A subset of faces creased to $\mathcal{C}^0$ are marked.



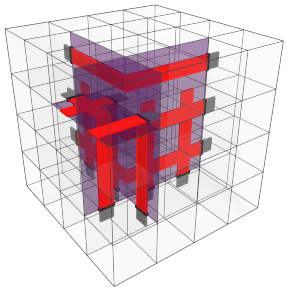
(c) A subset of faces creased to $\mathcal{C}^0$ are marked.



(d) All faces creased to $\mathcal{C}^0$ are shown together.

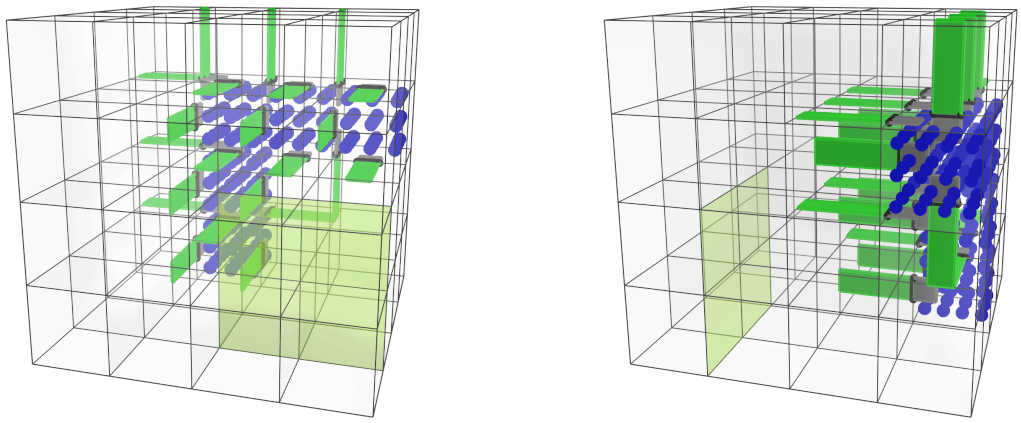


(e) The continuity transition ribbons are shown alone.



(f) The complete admissible volumetric mesh, including the continuity transition ribbons.

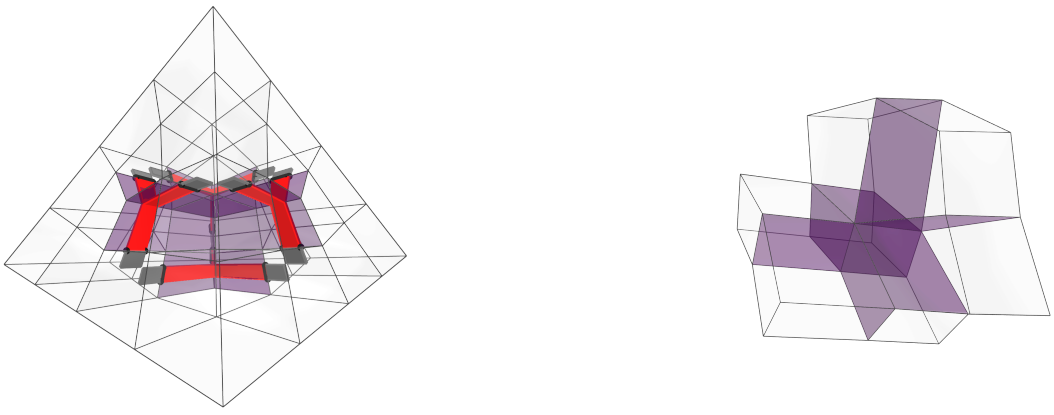
Figure 43: An example of an admissible volumetric mesh that satisfies the ϑ-separation condition. All elements are quadratic and all faces have continuity $\mathcal{C}^1$ except for three mutually orthogonal planes of faces creased to $\mathcal{C}^0$. For clarity, various parts of the mesh are shown separately before they are shown together. Notice the tail-to-tail and head-to-tail ribbon intersections, which are admissible. Ribbons that cross at the center of a face are not considered intersecting and are admissible.



(a) Front view.

(b) Side view.

Figure 44: An admissible mesh that satisfies the p-separation condition. This mesh is cubic $\mathcal{C}^1$ everywhere except for five cells on the far side, which have been set to $p = (3, 2, 2)$, and a few faces on the near side which are set to $\mathcal{C}^3$ (supersmooth). The degree transition ribbons may be seen extending away from the cells with lower degree. Observe that these transition ribbons do not intersect the edge where a supersmooth continuity transition occurs, thus maintaining the admissibility of the mesh. Both the front view and side view of the mesh are shown.



(a)

(b)

Figure 45: Two quadratic volumetric meshes that satisfy the ϑ-grading condition. The mesh on the left was constructed by filling a tetrahedron with hexahedral cells. All faces adjacent to an extraordinary edge must be creased to $\mathcal{C}^0$, but there are cases where a face near but not directly adjacent to an extraordinary edge must also be creased, as seen in the example on the right (see also eq. (114)).

- R: A global parameterization is possible.
  - It is possible to construct a submesh domain $\Omega^{\mathbf{U}}$ and an associated coordinate system $\alpha^{\mathbf{K}}$ on the U-spline mesh $\mathbf{U}$.
- r: A global parameterization may not be possible.

- H: Supersmooth interfaces are not permitted in the mesh.
  - All interfaces in the mesh have continuity less than $p^{\perp}_{\min}(\mathsf{I})$.
  - $\forall \mathsf{I} \in \mathbf{U}, \vartheta^{\mathsf{I}} < p^{\perp}_{\min}(\mathsf{I})$.
- h: Supersmooth interfaces are permitted in the mesh.

- K: Smoothness propagates globally.
  - For any ribbon of maximum coupling length r in the mesh, the continuity of the ribbon head I^h is equal to the continuity on any interface in the tail.
  - $\forall \mathsf{r} \in \mathbf{U}, \vartheta^{\mathsf{I}^h} = \vartheta^{\mathsf{I}_i}, \mathsf{I}^h \in \mathsf{r}, \mathsf{I}_i \in \boldsymbol{t} \in \mathsf{r}$.
- k: Local variation in smoothness is permitted.

- P: Polynomial degree propagates globally.
  - For all interfaces in the mesh, the polynomial degree parallel to the interface is the same on both cells adjacent to the interface.
  - $\forall \mathsf{I} \in \mathsf{C}^{d-1}(\mathbf{U}), p^{\parallel}_{\mathsf{e}}(\mathsf{c}) = p^{\parallel}_{\mathsf{e}}(\mathsf{c}'), \mathsf{e} \in \mathsf{ADJ}^1(\mathsf{I}), \{\mathsf{c}, \mathsf{c}'\} \subseteq \mathsf{ADJ}^d(\mathsf{I})$.
- p: Local variation in polynomial degree is permitted.

Table 1: The definition of each superscript used to identify a U-spline mesh class.

10 The U-spline basis

A U-spline basis is constructed over a U-spline mesh $\mathbf{U}$ as follows:

1. Determine the index support $\mathsf{id}_{\boldsymbol{n}} = \mathsf{ID}(\boldsymbol{n}), \boldsymbol{n} \in \mathbf{BV}(\mathsf{c})$ for each k-cell basis vector $\boldsymbol{n}$ associated with each k-cell $\mathsf{c} \in \mathbf{U}$ (see section 8).
2. Determine the index support id_A of each U-spline basis vector by constructing a corresponding core graph G_A (see section 10.1).
3. Determine the basis vector $\boldsymbol{u}_A$ of each U-spline basis function by solving the rank one null space problem $\boldsymbol{R}|_{\mathsf{id}_A}$ (see section 10.2).
4. Normalize the set of U-spline basis vectors $\mathbf{UV}(\mathbf{U})$ to determine the set of U-spline basis functions $\mathsf{UF}(\mathbf{U})$ (see section 10.3).

10.1 The core graph

In order to construct U-spline basis vectors or, equivalently, U-spline basis functions, we need to create collections of vertex basis vectors and represent relationships between them. We do this by defining a core graph for each U-spline basis function. A core graph G_A is a directed acyclic graph that combines adjacent vertex basis vectors into the index support id_A for a single U-spline basis vector $\boldsymbol{u}_A$. An algorithm to compute G_A is given in algorithm 1.

10.1.1 Cores

Each node in the graph, called a core and denoted by κ, corresponds to a set of vertex basis vectors, i.e., $\kappa \subseteq \mathbf{BV}(\mathsf{v})$. To retrieve the core associated with a vertex v we use $\kappa(\mathsf{v})$. The boundary of a core in the direction of an edge e, denoted by $\mathbf{BD}_{\mathsf{e}}(\kappa)$, is computed in the same manner as basis vector boundaries (section 8.7). The set of children cores of κ is denoted by children(κ). We say that cores κ_i and κ_j are conforming if $\mathbf{BD}_{\mathsf{e}}(\kappa_i) = \mathbf{BD}_{\mathsf{e}}(\kappa_j)$ on a shared edge e. Edges between conforming cores represent parent/child relationships in G_A.

10.1.2 Expansion edges

The core graph algorithm functions by iterating over candidate *expansion edges* on which to expand. An expansion edge is a directed edge from vertex v_i to vertex v_j, denoted $\mathsf{e}_{i,j}$. For the subsequent definitions it is convenient to define several auxiliary sets. The set of *expanding basis vectors* on a vertex adjacent to a core is given by

$$\mathbf{XBV}(\kappa_i, \mathsf{v}_j) = \{\boldsymbol{n} \in \mathbf{BV}(\mathsf{v}_j) : \mathsf{C}(\mathsf{ID}(\boldsymbol{n})) \not\subseteq \mathsf{C}(\mathsf{ID}(\kappa_i))\}. \tag{115}$$

The set of *interacting edges* is given by

$$\mathsf{IE} = \{\mathsf{e}_{i,j} : \mathbf{SBV}(\kappa_i) \cap \mathbf{SBV}(\mathbf{XBV}(\kappa_i, \mathsf{v}_j)) \neq \varnothing\}. \tag{116}$$

The set of *covered edges* is given by

$$\mathsf{CE} = \{\mathsf{e}_{i,j} : \mathbf{BD}_{\mathsf{e}}(\kappa_i) \subseteq \mathbf{SBV}(\kappa_j)\}. \tag{117}$$

Finally, let FE be the set of directed edges for which the algorithm has tried and failed to find a conforming child core. Then, the set of candidate expansion edges are given by:

$$\mathsf{XE} = \mathsf{IE} \setminus (\mathsf{CE} \cup \mathsf{FE}). \tag{118}$$

The core graph algorithm proceeds in a breadth-first manner. That is, in a graph with multiple candidate expansion edges, we prioritize those edges originating from cores closest to the root of the graph.

Algorithm 1 Compute core graph G_A from given vertex basis vector $\boldsymbol{n}$

procedure ComputeCoreGraph($\boldsymbol{n}$)
  $\kappa_r \leftarrow \{\boldsymbol{n}\}$ ▷ The root core consists of the input null vector.
  $\kappa(\mathsf{v}_r) \leftarrow \kappa_r$ ▷ Initialize the graph by inserting the root core.
  while $\mathsf{XE} \neq \varnothing$ **do**
    $\mathsf{e}_{i,j} \leftarrow \mathsf{next}(\mathsf{XE})$ ▷ Get next expansion edge from XE.
    if $\kappa(\mathsf{v}_i)$ is an ancestor of $\kappa(\mathsf{v}_j)$ **then** ▷ A connection from κ_i to κ_j would create a cycle.
      $\mathsf{FE} \leftarrow \mathsf{FE} \cup \mathsf{e}_{i,j}$ ▷ Mark $\mathsf{e}_{i,j}$ as failed and go to next iteration.
      continue
    end if
    $\kappa_{\text{new}} \leftarrow \{\boldsymbol{n} \in \mathbf{BV}(\mathsf{v}_j) : \mathbf{BD}_\mathsf{e}(\boldsymbol{n}) \subseteq \mathbf{BD}_\mathsf{e}(\kappa(\mathsf{v}_i))\}$ ▷ Construct a new core on v_j that conforms to κ_i.
    if $\kappa_{\text{new}} = \varnothing$ **then** ▷ If κ_{new} is empty, expansion failed along this edge.
      $\mathsf{FE} \leftarrow \mathsf{FE} \cup \mathsf{e}_{i,j}$
      continue
    end if
    $\kappa(\mathsf{v}_j) \leftarrow \kappa(\mathsf{v}_j) \cup \kappa_{\text{new}}$ ▷ Merge κ_{new} with any existing null vectors in $\kappa(\mathsf{v}_j)$.
    $\text{children}(\kappa(\mathsf{v}_i)) \leftarrow \text{children}(\kappa(\mathsf{v}_i)) \cup \kappa(\mathsf{v}_j)$ ▷ Add a graph edge from κ_i to κ_j.
    for each $\kappa_c \in \text{children}(\kappa(\mathsf{v}_j))$ **do** ▷ Prune any children of $\kappa(\mathsf{v}_j)$ that no longer conform.
      if $\mathbf{BD}_\mathsf{e}(\kappa_c) \neq \mathbf{BD}_\mathsf{e}(\kappa_j)$ **then**
        remove κ_c and all descendants from G_A
      end if
    end for
    $\mathsf{FE} \leftarrow \mathsf{FE} \setminus (\mathsf{e}_{i,j} \cup \mathsf{e}_{j,i})$ ▷ Remove $\mathsf{e}_{i,j}$ and its opposite from failed edges.
  end while
  $\mathsf{success} \leftarrow \mathsf{FE} = \varnothing$ ▷ The algorithm succeeds only if it terminates with no failed edges.
  return G_A, success
end procedure

10.1.3 Algorithm

Algorithm 1 describes the procedure for constructing a core graph. To illustrate the behavior of the core graph algorithm, first consider the one-dimensional cubic U-spline mesh shown in fig. 46, where each feature of a core graph is depicted, including the root core κ_0, two child cores κ_1 and κ_2, and the expansion candidate edges XE which resulted in the two children being added to the graph. Next, consider the two-dimensional cubic U-spline mesh shown in fig. 47. This mesh has twelve cells and continuity $\mathcal{C}^2$ everywhere except for one edge which is $\mathcal{C}^3$, forming a supersmooth interface. The Bernstein coefficients of the completed basis function are listed in appendix E.1. For additional insight into U-spline basis construction, see the exercises in appendix A.5.

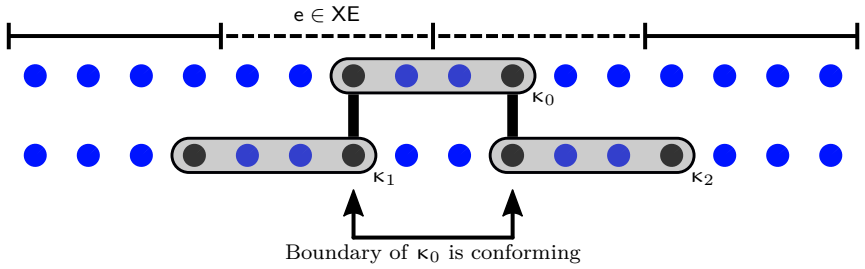
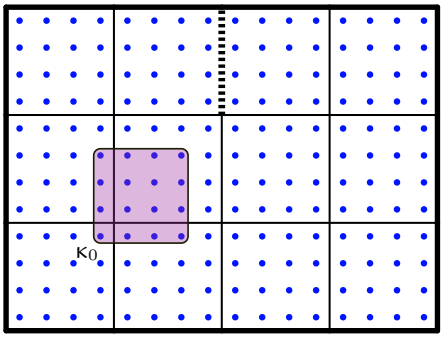
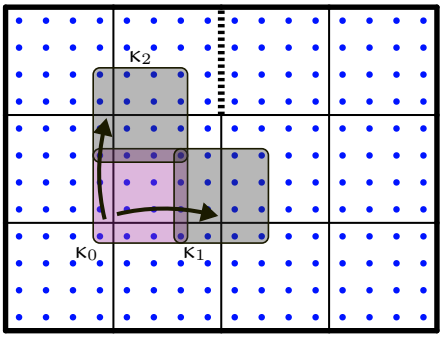


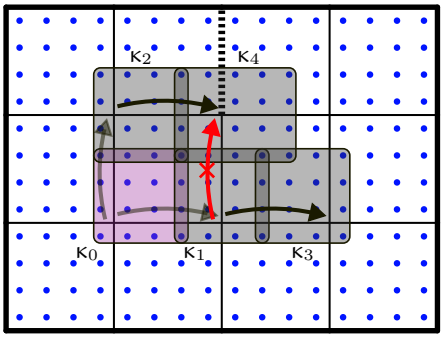
Figure 46: An example of a simple core graph on a one-dimensional cubic U-spline mesh. The continuity on the interfaces is $\mathcal{C}^2$.



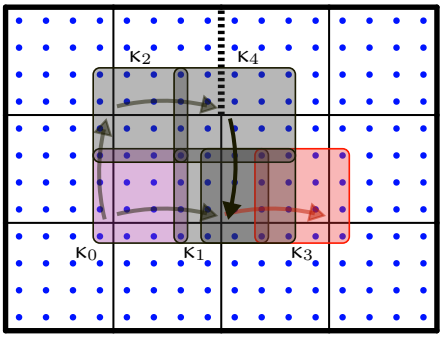
(a) The root core κ_0.



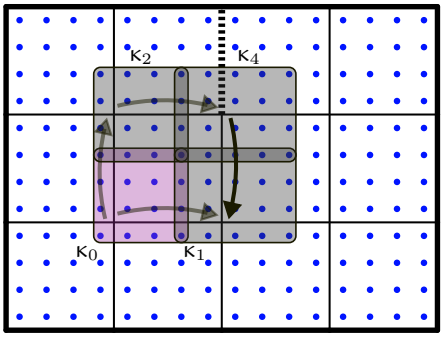
(b) The core graph is expanded along the edges to the right and top, forming cores κ_1 and κ_2.



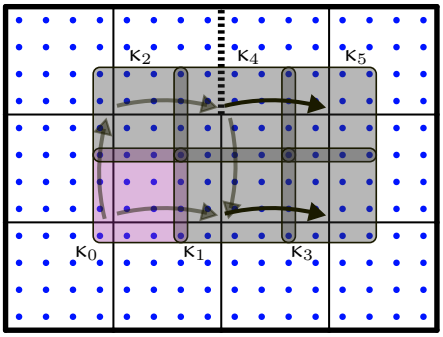
(c) The core graph is expanded a second time, to the right on both legs of the graph, forming cores κ_3 and κ_4. A failed expansion is encountered when trying to expand upwards from κ_1 .



(d) Expanding from the other direction, the failed edge is resolved, but this results in adding an extra basis vector to κ_1, which causes κ_3 to become inconsistent with its parent core.



(e) The inconsistent child core is removed from the graph.



(f) The core graph construction is completed.

Figure 47: A two-dimensional core graph example. In this example, the U-spline mesh $\mathbf{U}$ is a 3-by-4 bicubic mesh with a single supersmooth interface. All interior edges have continuity $\mathcal{C}^2$ (thin solid lines) except for one, which has continuity $\mathcal{C}^3$ (thick dashed line), forming a supersmooth interface.

10.2 The rank one null space problem

The U-spline index support id_A is extracted from the combined index supports of the cores in G_A. We then consider a restricted rank one constraint matrix $\boldsymbol{R}|_{\mathsf{id}_A}$ and associated null space problem. In the multi-dimensional setting, the smoothness constraints often form a system of linearly dependent equations. That is, the constraint matrix $\boldsymbol{R}|_{\mathsf{id}_A}$ is often not square, with the number of rows m being greater than the number of columns n. To solve for the Bernstein coefficients of the U-spline basis vector $\boldsymbol{u}_A$, one approach is

to solve for the reduced row echelon form of $\boldsymbol{R}|_{\mathsf{id}_A}$ via Gaussian elimination, resulting in a matrix with $m-n$ rows equal to $\mathbf{0}^T$. However, it has been shown that Gaussian elimination on floating point numbers leads to unacceptably high accumulated error when analyzing the null space of spline constraint equations [2].

An alternative approach is to cast the problem as the linear program

$$\begin{aligned} \text{minimize} \quad & \mathbf{1}^T \boldsymbol{u}_A && (119)\\ \text{subject to} \quad & \boldsymbol{R}|_{\mathsf{id}_A} \boldsymbol{u}_A = \mathbf{0} && (120)\\ & \boldsymbol{u}_A \geq \mathbf{1}. && (121) \end{aligned}$$

Note that we have enforced the lower bound $\boldsymbol{u}_A \geq \mathbf{1}$ to preclude the trivial solution of $\boldsymbol{u}_A = \mathbf{0}$. This linear program can then be solved using any number of established methods such as simplex methods or interior-point methods. We have used the revised simplex method implemented in the `lp_solve` package [7]. Thus far we have found this approach to be sufficient for solving our problems of interest, and as such have not compared the various algorithms that may be used to solve the above linear program. Instead, a detailed comparison of the solution methods that may be employed to solve the linear system of constraint equations for a given function will be the topic of a future work.

10.3 Normalization

To recover a partition of unity in the U-spline basis we perform a normalization step. In other words, we want to find a set of *positive* coefficients $c_A \in \mathbb{R}_+, A = 1, \ldots, |\mathsf{UF}|$, such that

$$\begin{aligned} 1 &= \sum_A c_A N_A && (122)\\ &= \sum_A \bar{N}_A && (123) \end{aligned}$$

where $\bar{N}_A$ is a *normalized* U-spline basis function. This normalization is always possible due to the underlying structure of the null space N.

This problem can be solved in a variety of ways such as by constructing a full rank linear system by sampling the U-spline basis at $|\mathsf{UF}|$ unique locations or by recasting the problem as a linear programming problem and solving it using a simplex method or similar technique as described in section 10.2.

Note that there exists a set of non-negative coefficients, $c_A \in \mathbb{R}_+, A = 1, \ldots, |\mathsf{UF}|$, such that

$$1 = \sum_A c_A N_A. \tag{124}$$

This can be shown using the following reasoning. Since the function $f = 1$ is an analytic function and $f \in \mathcal{B}^{\mathsf{c}}$ for every $\mathsf{c} \in \mathbf{U}$ we know that $f \in \mathsf{N}$. This means there exists a set of coefficients, $c_A \in \mathbb{R}, A = 1, \ldots, |\mathsf{UF}|$, such that

$$\sum_A c_A \boldsymbol{u}_A = \mathbf{1}. \tag{125}$$

Since by construction $\mathsf{N} \subset \mathbb{R}_+^{|\mathsf{UF}|}$, where $\mathbb{R}_+^{|\mathsf{UF}|}$ is the non-negative orthant, and $\mathbb{R}^{|\mathsf{UF}|}$ is a polyhedral cone then N is a polyhedral cone as well. This means that, in fact, the coefficients are also non-negative, i.e., $c_A \in \mathbb{R}_+$. Consequently, we have that

$$\begin{aligned} \sum_A c_A \boldsymbol{u}_A &= \mathbf{1} && (126)\\ \sum_A c_A \sum_{\mathsf{i} \in \mathsf{id}_A} u_{\mathsf{i}}^A B_{\mathsf{i}} &= \sum_{\mathsf{i} \in \mathsf{ID}(\mathbf{U})} B_{\mathsf{i}} && (127)\\ \sum_A c_A N_A &= 1 && (128)\\ \sum_A \bar{N}_A &= 1. && (129) \end{aligned}$$

11 The U-spline space

Given a U-spline mesh $\mathbf{U}$, a U-spline space, denoted by $\mathcal{U}(\mathbf{U})$ or $\mathcal{U}$ for short, can now be defined. As described in section 6, we will construct the U-spline space by leveraging the nullspace perspective for splines. However, rather than constructing and attempting to find a solution to the global nullspace problem, which can be computationally expensive and numerically unstable [12, 13], we will instead solve a single small, highly localized rank one nullspace problem for each U-spline basis function.

11.1 Completeness and the neighborhood of interaction

Because adjacent cells with differing polynomial degree can occur in admissible meshes, the notion of completeness of the U-spline basis must take into account the way a cell of lower degree affects the completeness on adjacent cells which may have a local Bernstein basis of higher degree, but only have completeness at a lower total degree, bounded by the lower degree of nearby cells.

For example, in fig. 48, we show a U-spline mesh with three quadratic and one linear cell, and $\mathcal{C}^0$ continuity on the interfaces. The indices of the nonzero coefficients of the U-spline basis functions on this mesh are highlighted in gray. Observe that the cells directly to the left and directly below the linear cell each have a local Bernstein basis of degree $\mathsf{p} = (2, 2)$, which has a total of nine functions, yet there are only eight U-spline basis functions which are nonzero on those cells. Due to their close proximity to the linear cell, these cells are complete up to degree $\mathsf{p} = (2, 1)$ and $\mathsf{p} = (1, 2)$, respectively, or complete up to total degree $p = 1$. The cell diagonal from the linear cell, on the other hand, is not impacted by the linear basis and remains complete up to degree $\mathsf{p} = (2, 2)$.

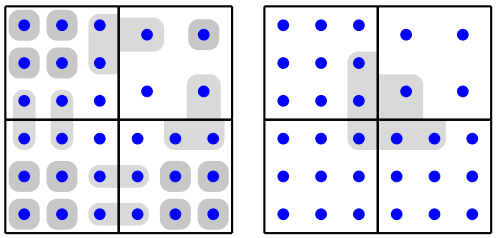


Figure 48: A U-spline mesh with three quadratic and one linear cell, and $\mathcal{C}^0$ continuity on the interfaces. The indices of the nonzero coefficients of the U-spline basis functions are highlighted in gray. The quadratic cells on the top-left and bottom-right have a reduced degree of completeness due to their proximity to the linear cell.

To describe this behavior, we define a submesh $\mathsf{NI}(\mathsf{c})$, called the neighborhood of interaction, of a given d-cell c. Let the set of basis functions that are nonzero over a k-cell c be denoted by $\mathsf{UF}(\mathsf{c})$ and the extended support of a k-cell be denoted by

$$\operatorname{supp}(\mathsf{UF}(\mathsf{c})) = \bigcup_{N_A \in \mathsf{UF}(\mathsf{c})} \operatorname{supp}(N_A) \,. \tag{130}$$

Additionally, we denote the set of all d-cells that can be reached in a cardinal submesh direction that is orthogonal to the interfaces that are adjacent to a d-cell c by $\mathsf{CRD}(\mathsf{c})$. Figure 49 shows $\mathsf{CRD}(\mathsf{f})$ for a face f. Then, the neighborhood of interaction $\mathsf{NI}(\mathsf{c})$ of a given d-cell c is defined as

$$\mathsf{NI}(\mathsf{c}) = \operatorname{supp}(\mathsf{UF}(\mathsf{c})) \cap \mathsf{CRD}(\mathsf{c}). \tag{131}$$

The neighborhood of interaction of the linear cell in the mesh in fig. 48 is highlighted in fig. 50. The completeness of a U-spline space is then defined as described in section 11.2.

11.2 Mathematical properties

The mathematical properties satisfied by a U-spline space are:

- **Local linear independence:** The set of U-spline basis functions are locally linearly independent. This means that, for any submesh $\mathbf{K} \subseteq \mathbf{U}$, $\sum_A c_A N_A(\boldsymbol{s}) = 0$ for all $\boldsymbol{s} \in \hat{\Omega}^{\mathbf{K}}$, where $\boldsymbol{c}[\mathbf{U}] = \{c_A\}$ is a set of real coefficients, if and only if $\boldsymbol{c}[\mathbf{U}] = \mathbf{0}$.

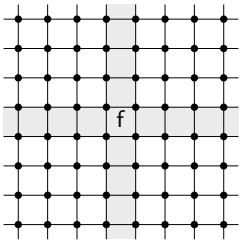


Figure 49: The faces that can be reached in a cardinal submesh direction that is orthogonal to the interfaces adjacent to f, denoted by the set $\mathsf{CRD}(\mathsf{f})$.

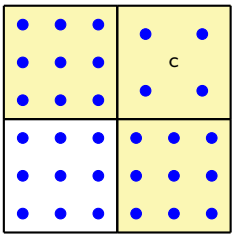


Figure 50: The neighborhood of interaction $\mathsf{NI}(\mathsf{c})$ is highlighted for a cell c on a two-dimensional U-spline mesh with three quadratic and one linear cell and continuity $\mathcal{C}^0$ on the interfaces.

- **Completeness:** A set of U-spline basis functions is complete through total polynomial degree

$$|q^{\mathsf{c}}| = \min_{\mathsf{a} \in \mathsf{NI}(\mathsf{c})} |p^{\mathsf{a}}| \tag{132}$$

  over $\hat{\Omega}^{\mathsf{c}}$. Additionally, a U-spline space is complete through total polynomial degree

$$\left|q^{\mathbf{U}}\right| = \min_{\mathsf{c} \in \mathbf{U}} |q^{\mathsf{c}}| \tag{133}$$

  over $\hat{\Omega}^{\mathbf{U}}$. In other words, there exists a set of real coefficients $\{c_A\}$ such that $\sum_A c_A N_A(\boldsymbol{s}) = \boldsymbol{s}^r$ for any $r \leq |q^{\mathsf{c}}|, \boldsymbol{s} \in \hat{\Omega}^{\mathsf{c}}$ or $r \leq \left|q^{\mathbf{U}}\right|, \boldsymbol{s} \in \hat{\Omega}^{\mathbf{U}}$.

- **Pointwise non-negativity:** A set of U-spline basis functions are pointwise non-negative. More precisely, $N_A(\boldsymbol{s}) \geq 0$ for all $\boldsymbol{s} \in \hat{\Omega}^{\mathbf{U}}$, $A = 1, \ldots, |\mathsf{UF}(\mathbf{U})|$ where $\mathsf{UF}(\mathbf{U})$ is the set of U-spline basis functions that are nonzero over $\hat{\Omega}^{\mathbf{U}}$.

- **Partition of unity:** A set of U-spline basis functions forms a partition of unity. In other words, $\sum_A N_A(\boldsymbol{s}) = 1$ for all $\boldsymbol{s} \in \hat{\Omega}^{\mathbf{U}}$.

- **Compact support:** Compact support simply means that for any U-spline basis function N_A there exists a submesh $\mathbf{K}_A \subseteq \mathbf{U}$ such that for any $\boldsymbol{s} \in \hat{\Omega}^{\mathbf{U}}$

$$\begin{cases} N_A(\boldsymbol{s}) > 0 & \boldsymbol{s} \in \hat{\Omega}^{\mathbf{K}_A}, \\ N_A(\boldsymbol{s}) = 0 & \text{otherwise.} \end{cases} \tag{134}$$

  It is desirable for the submesh $\mathbf{K}_A$ to be as small as possible to preserve the sparsity of linear systems written in terms of the basis functions.

11.3 Numerical verification

The approach to building a basis on a U-spline mesh presented in section 10 is algorithmic in nature. In order to verify that this algorithm will always successfully build a valid U-spline basis with the desired properties on every U-spline mesh, we performed extensive testing by running the U-spline algorithm on a

large number and variety of one- and two-dimensional input U-spline meshes. For three-dimensional meshes, due to computational speed limitations we focused our testing on a specific set of tailor-made volumetric meshes that specifically conform to or violate each admissibility condition. A formal proof of the correctness of the algorithm is beyond the scope of this work.

The set of all admissible U-spline meshes is very large, and it is impractical to test every possible mesh configuration. In order to ensure that our test suite contained sufficient variety to provide reasonable evidence of the validity of the U-spline algorithm, for one- and two-dimensional meshes we leveraged the U-spline mesh classification system presented in section 9.3 to generate test meshes within several of these classes, representing increasing levels of complexity.

11.3.1 Overview of verification procedure

We first generated meshes within the most structured class $\boldsymbol{\mathcal{U}}^{RHKP}$, and then proceeded to remove structure until finally a large number of meshes in the class $\boldsymbol{\mathcal{U}}^{rhkp}$ were tested (see table 1 for a definition of each superscript). In one dimension, we focused on class $\boldsymbol{\mathcal{U}}^{RhKP}$. For each mesh, we used the U-spline algorithm to construct a set of basis functions. These basis functions were then analyzed to ensure they satisfied all the mathematical properties of a U-spline space as described in section 11.2.

One dimension For one-dimensional meshes, the class $\boldsymbol{\mathcal{U}}^{RhKP}$ was selected, which allows any degree on each cell and continuity on each interface up to $\mathcal{C}^\infty$. This is the most unstructured class possible in one dimension.

Two dimensions On two-dimensional meshes, five gradations of structure were selected.

- $\boldsymbol{\mathcal{U}}^{RHKP}$: Tensor-product topology; degree and continuity propagate globally.
- $\boldsymbol{\mathcal{U}}^{RhkP}$: Tensor-product topology; degree propagates globally but continuity may vary locally (including supersmooth interfaces).
- $\boldsymbol{\mathcal{U}}^{RHKp}$: Tensor-product topology; continuity propagates globally but degree may vary locally.
- $\boldsymbol{\mathcal{U}}^{rHkP}$: Meshes may include extraordinary vertices and triangles; degree propagates globally but continuity may vary locally.
- $\boldsymbol{\mathcal{U}}^{rhkp}$: Fully unstructured.

Within each class, meshes were randomly constructed to include as much variation as possible while still conforming to the admissibility conditions described in section 9.2. This included degree up to $p = 3$ and continuity up to $\vartheta = \mathcal{C}^2$ or when permitted, supersmooth continuity. The process of constructing these meshes involved starting with a specific base topology, listed below, and then applying further modifications as the class allowed, such as adding extraordinary vertices, triangles, and variations in degree and continuity. The base topologies used are as follows. Simple representations of these base topologies are shown in fig. 51.

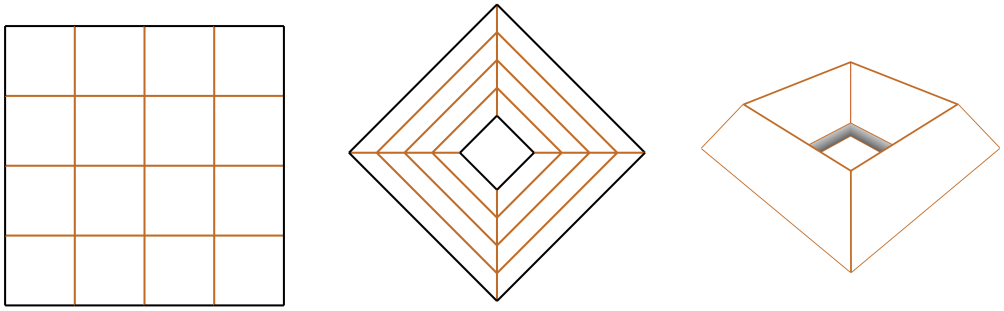
- **Line**. A line topology is a non-periodic one-dimensional sequence of edges, and is denoted by L.
- **Loop** (has periodicity). A loop topology is a periodic one-dimensional sequence of edges, and is denoted by P.
- **Regular grid**. A regular grid topology is a tensor product of two non-periodic one-dimensional mesh topologies, and is denoted
$$\mathsf{G} = \mathsf{L} \otimes \mathsf{L} \,. \tag{135}$$
- **Annulus** (periodicity in one tensor product direction). An annulus topology is the tensor product of a periodic one-dimensional topology with a non-periodic one-dimensional topology, and is denoted
$$\mathsf{A} = \mathsf{P} \otimes \mathsf{L} \,. \tag{136}$$

- **Torus** (periodicity in both tensor product directions). A torus topology is the tensor product of two periodic one-dimensional topologies, and is denoted

$$\mathsf{T} = \mathsf{P} \otimes \mathsf{P} \,. \tag{137}$$

- **Triangle**. A triangle mesh topology is an unstructured mesh containing only triangles, and is denoted Δ.
- **Mixed grid**. A mixed grid mesh is a mesh containing both quadrilateral and trianglar cells that is constructed by taking a tensor product mesh (grid, annulus, or torus), and then splitting a subset of the quadrilateral cells into two or more triangles. A mixed grid mesh is denoted M.
- **Multi-patch**. A multi-patch topology is constructed of multiple regular grid, mixed grid, or triangle meshes, sewn together along conforming boundaries. A multi-patch topology is denoted X.

Following these guidelines, twenty-five thousand two-dimensional admissible U-spline meshes were constructed, with variation in degree, continuity, mesh size, and topology. In addition, several thousand one-dimensional admissible U-spline meshes of all combinations of degree and continuity were also constructed. The U-spline algorithm successfully constructed a basis on each of these meshes that was verified to satisfy all the mathematical properties of a U-spline space described in section 11.2.



(a) A regular grid, no periodicity.

(b) An annulus, periodic in one dimension.

(c) A torus, periodic in two dimensions.

Figure 51: The three base tensor-product topologies used for constructing two-dimensional U-spline test meshes for verification.

Three dimensions Due to computational speed limitations, it was impractical to generate and test volumetric meshes in the same way as was done in one and two dimensions. Instead, a specific set of tailor-made volumetric meshes were created that specifically conform to or violate each admissibility condition. Our tests demonstrated that the U-spline algorithm successfully built a valid U-spline basis with the desired properties on each of the admissible meshes.

12 Notable U-spline examples

12.1 Supersmooth interfaces

Figure 52 shows a U-spline mesh where two perpendicular continuity transition ribbons, emanating from supersmooth interfaces, touch at a common endpoint. The support of a nearby U-spline basis function is also shown. The control points for a linear parameterization of this mesh along with a countour plot of the highlighted basis function are seen in fig. 53. Notice the non-rectangular support of the basis function, required in this configuration to ensure local linear independence of the U-spline basis near the transition.

This example contrasts with T-splines, that leverage a type of supersmooth interface, called a T-junction, which only produce blending functions that have a tensor product structure [55, 35]. U-splines, on the other

hand, are not limited to a tensor product structure, and can therefore produce an analysis-suitable basis on certain meshes which cannot produce analysis-suitable T-splines. This is particularly evident in cases where infinitely smooth interfaces are close enough in proximity to interact, as we see in this example. The values of the Bernstein coefficients of the highlighted basis function are listed in the appendix E.2. See also appendix E.3 to compare with a U-spline that is equivalent to an analysis-suitable T-spline.

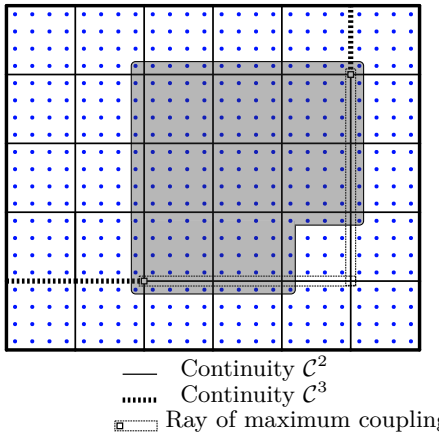
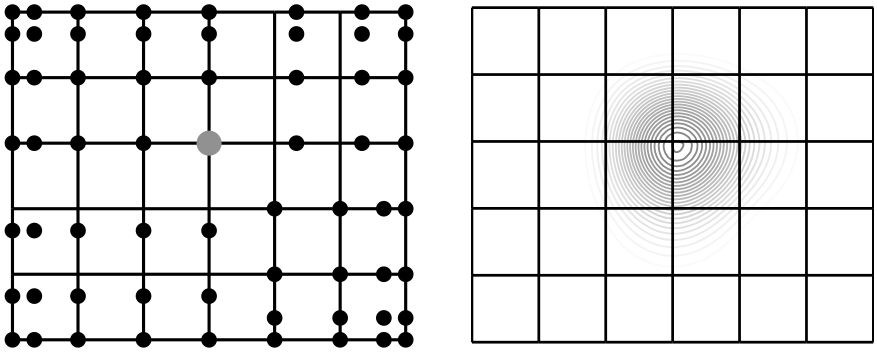


Figure 52: An example of a U-spline basis function that has a non-rectangular shape due to the close proximity of two continuity transitions near supersmooth interfaces. Notice the two perpendicular transition ribbons, starting at the vertices adjacent to the supersmooth interfaces, which touch at a common endpoint. This is an example of a basis function which is not possible with T-splines, which require basis functions to have a tensor product structure.



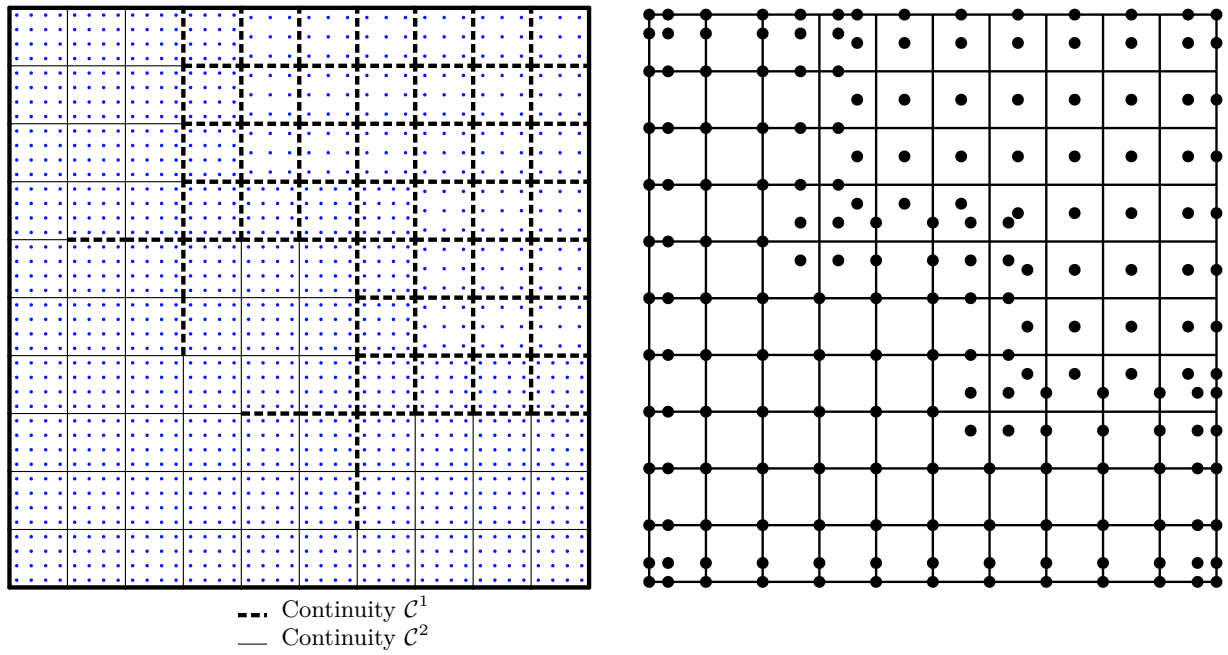
(a) The control points for a linear parameterization of the mesh in fig. 52. The control point associated with the basis function in fig. 52 is highlighted.

(b) A contour plot of the basis function whose nonzero Bernstein coefficients are shown in fig. 52.

Figure 53: A U-spline with local variation in smoothness.

12.2 Degree transitions

Figure 54 illustrates a transition from a degree 2, continuity $\mathcal{C}^1$ section of the mesh to a degree 3, continuity $\mathcal{C}^2$ section. The continuity in the vicinity of the degree transition must be carefully set to maintain the admissibility of the U-spline mesh.



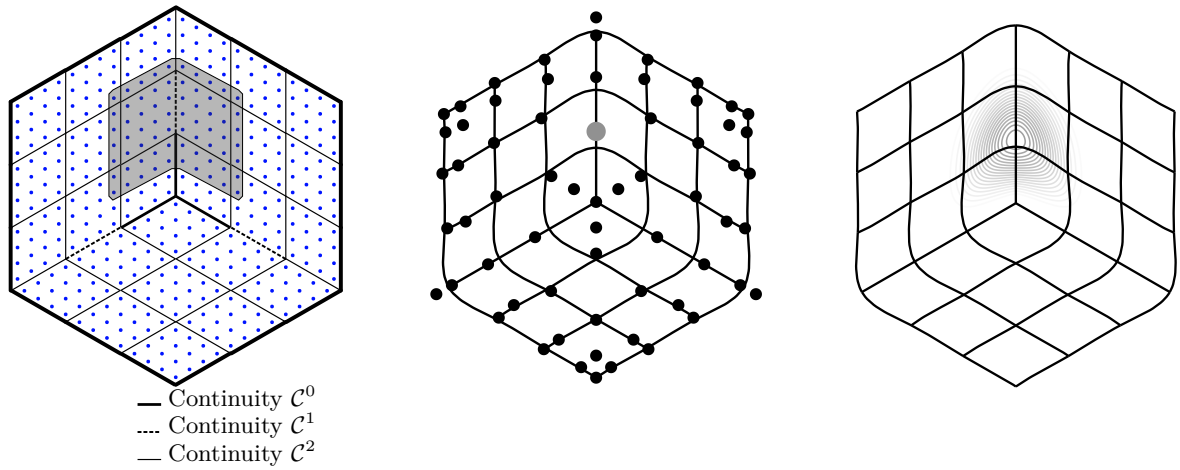
(a) A degree transition from $p = 2$, $\mathcal{C}^1$ (on the top-right) to $p = 3$, $\mathcal{C}^2$ (on the bottom-left). The creasing pattern here is required to ensure the U-spline mesh is admissible.

(b) The control points that form a linear parameterization for the U-spline mesh on the left.

Figure 54: A U-spline with local variation in degree.

12.3 Extraordinary vertices

A cubic U-spline mesh that contains an extraordinary vertex is shown in fig. 55a. Graded creasing on the edges emanating from the extraordinary vertex transition the continuity gradually from $\mathcal{C}^0$ to $\mathcal{C}^2$. The highlighted basis function overlaps these edges, transitioning from $\mathcal{C}^0$ to $\mathcal{C}^2$ smoothness within its support. The control points for a linear parameterization of the mesh are seen in fig. 55b, and a contour plot of the highlighted basis function is seen in fig. 55c. The coefficients for this basis function are listed in appendix E.4.

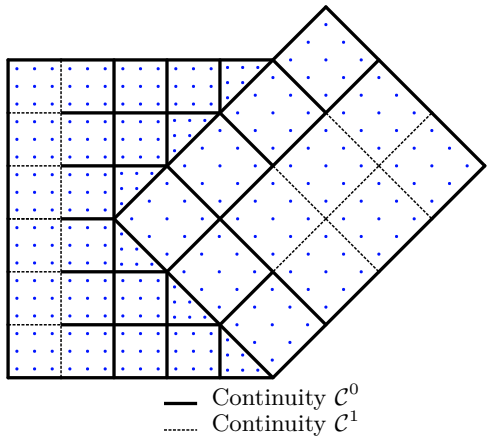


(a) (b) (c)

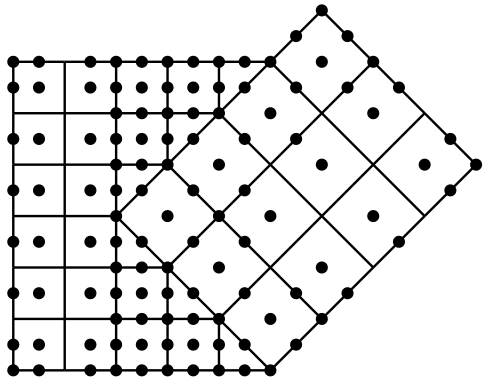
Figure 55: An example of a cubic U-spline basis function near an extraordinary vertex that transitions from $\mathcal{C}^0$ to $\mathcal{C}^1$, and then from $\mathcal{C}^1$ to $\mathcal{C}^2$. The coefficients for this example are listed in appendix E.4.

12.4 Triangles

In fig. 56 two tensor product regions meet diagonally. Triangles are used in the transition region. In fig. 57 triangles are utilized in one portion of the domain, which then interface with quadrilateral cells to transition to a $\mathcal{C}^2$ outer boundary.

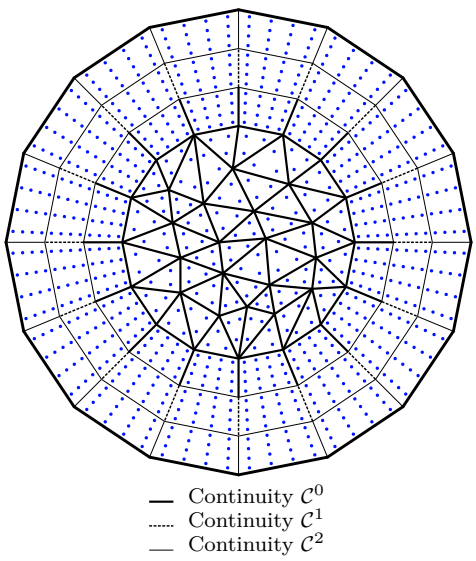


(a) A quadratic U-spline mesh where triangles allow two tensor product regions to meet diagonally. The edges near the triangles are creased to $\mathcal{C}^0$, but edges further out retain $\mathcal{C}^1$ continuity.

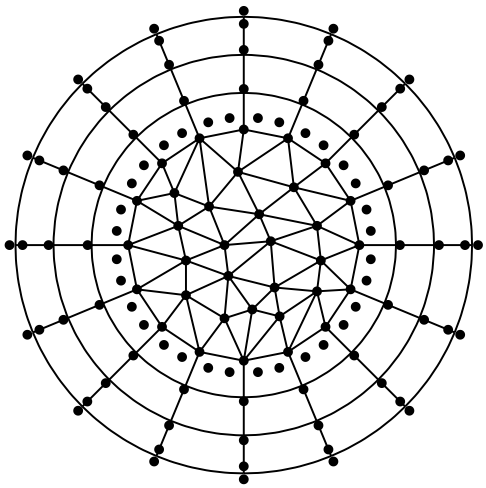


(b) The control points and geometry of a U-spline where two regular grids meet diagonally with the help of triangles at the interface.

Figure 56: A U-spline composed of quadrilateral and triangular cells.



(a) This U-spline mesh is cubic with $\mathcal{C}^2$ continuity on the outside edge, but is filled in the interior with linear triangles. A continuity transition was required between the $\mathcal{C}^0$ triangles and the $\mathcal{C}^2$ quadrilaterals.

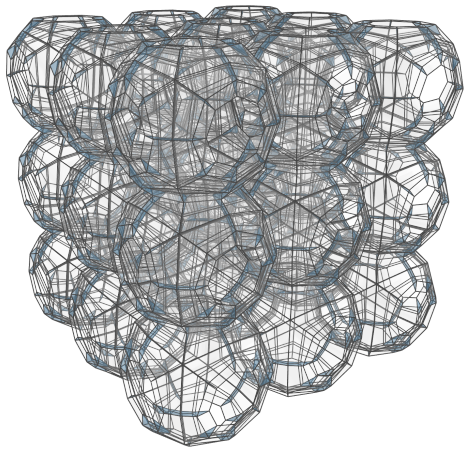


(b) The control points and geometry of the U-spline mesh in fig. 57a. Meshes of this type may prove useful to engineers who are only interested in smoothness on the surface, and opt for faster linear triangular cells on the interior.

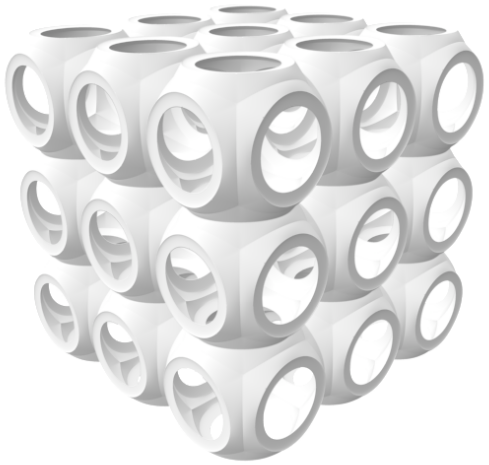
Figure 57: A U-spline with local variation in smoothness, degree, and cell type.

12.5 Unstructured volumetric U-splines

Figure 58a shows an example of a quadratic fully-unstructured volumetric U-spline mesh that forms a $3 \times 3 \times 3$ lattice structure with repeating unit cells sharing circular interfaces. This mesh has 3888 quadratic hexahedral elements. Most of the faces have $\mathcal{C}^0$ continuity (colorless faces) because they are adjacent to an extraordinary edge, while a few of the faces have $\mathcal{C}^1$ continuity (colored). The control points for this U-spline were chosen by projecting a linear mesh onto the U-spline basis. A rendered representation of the geometry defined by these control points is shown in fig. 58b.



(a) A quadratic fully-unstructured volumetric U-spline mesh which forms a $3 \times 3 \times 3$ lattice structure with unit cells shaped like spheres with holes.



(b) A rendered representation of a volumetric U-spline whose basis is defined by the U-spline mesh shown in fig. 58a.

Figure 58: An example of a fully-unstructured volumetric U-spline mesh. The U-spline mesh is on the left and the U-spline geometry is on the right.

Figure 59 shows an example of a quadratic fully-unstructured volumetric U-spline mesh of a segment of a piston. This mesh has 1539 quadratic hexahedral elements. All colorless faces have $\mathcal{C}^1$ continuity while the colored faces are $\mathcal{C}^0$. This example contains several extraordinary edges both interior and on boundaries. The continuity scheme set on the interfaces of this mesh uses the minimal creasing necessary to be admissible. Earlier spline methods (such as multi-patch NURBS) would have required many more faces to be creased. The control points for this U-spline were chosen by performing a projection of the linear mesh onto the U-spline basis. A rendered representation of the U-spline geometry is shown in fig. 60.

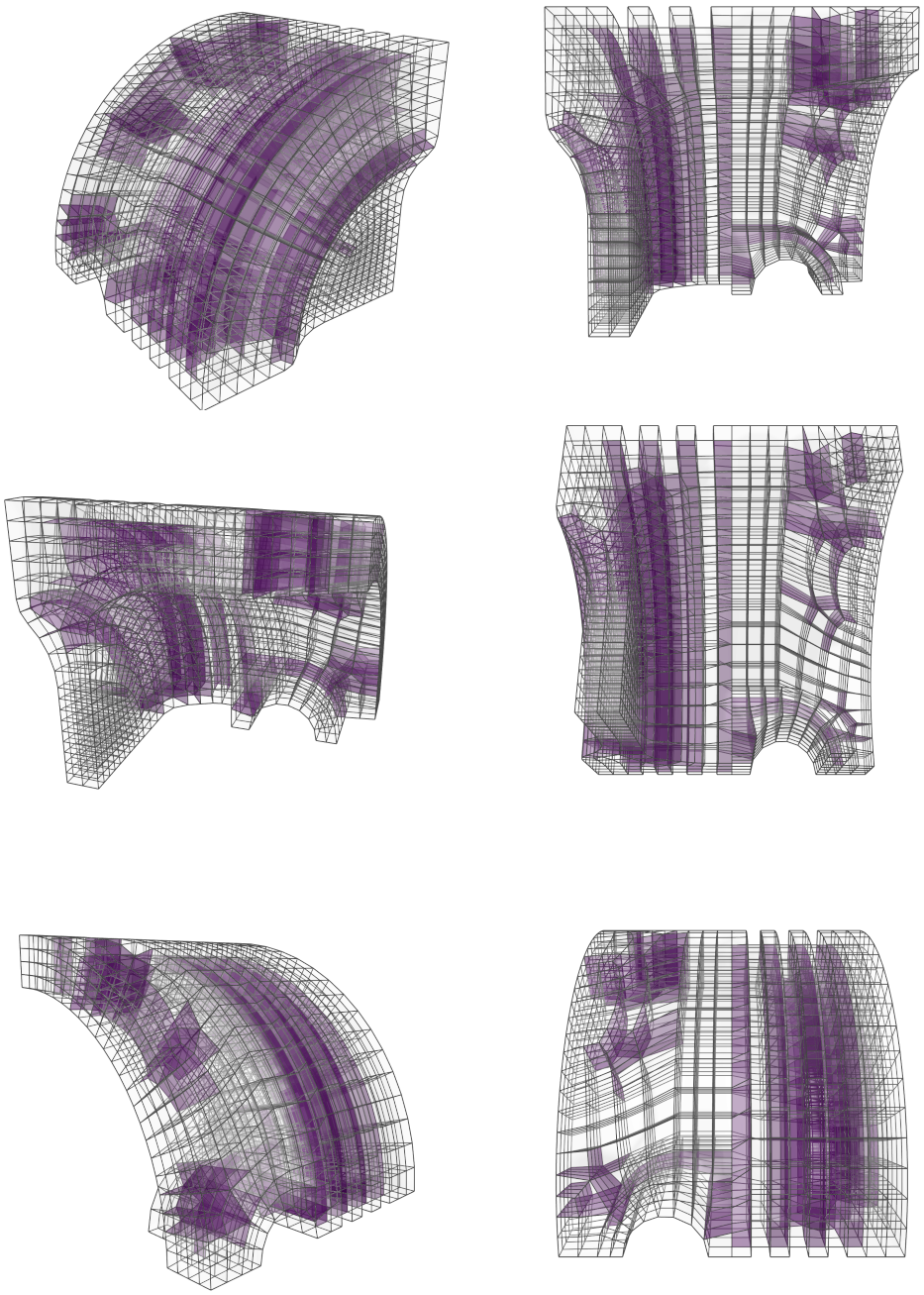


Figure 59: Several views of a fully-unstructured volumetric U-spline mesh of a segment of a piston.

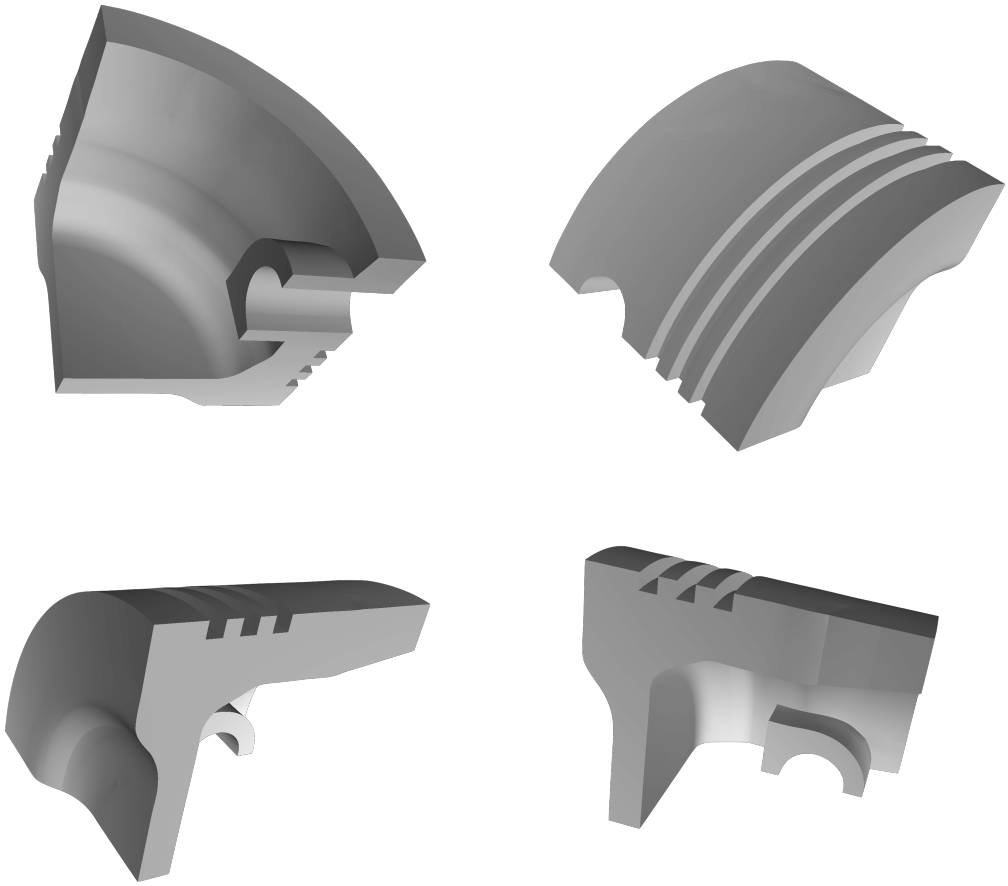


Figure 60: Several rendered views of the piston segment volumetric U-spline whose basis is defined by the mesh shown in fig. 59.

13 Conclusion

The ability to construct analysis-suitable smooth splines over unstructured meshes is a significant step forward for improving the capabilities of CAD/CAE systems. The U-splines technology introduced in this work is capable of smoothly and accurately representing complex CAD models, is compatible with prevailing industrial spline representations, such as NURBS and analysis-suitable T-splines, and supports the local variation of cell size (i.e., h-refinement), degree (i.e., p-refinement), and intercell continuity (i.e., ϑ-refinement) including support for extraordinary and mixed cell types. U-splines offers a locally-supported, complete, positive basis that forms a partition of unity and is locally linearly independent, making it ideal for use in simulation procedures such as FEA. Furthermore, the algorithm for constructing a U-spline basis is not hampered by dimensional limitations, which we have exploited to develop volumetric U-spline bases. It is our opinion that a smooth spline meshing technology for industrial-scale FEA problems should have all of these characteristics, and that U-splines is capable of fulfilling this purpose.

14 Acknowledgements

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under award numbers

- SBIR Phase II DE-SC0017051
- SBIR DE-SC0020906

- SBIR DE-SC0019945

Support was also provided by the Department of Defense, Navy, under contract numbers

- SBIR N68335-15-C-0245
- STTR N68335-16-C-0209

Support was also provided by the U.S. Army, STTR W911NF20P0002.
Funding was also provided by Honeywell Federal Manufacturing & Technologies, managing and operating partner of the Kansas City National Security Campus pursuant to Prime Contract DE-NA0002839 with the U.S. Department of Energy, under contract numbers

- N000254262
- N0000294261
- N000335868
- N000344045
- N000375957

Additional support was provided by a grant from the Utah Science Technology and Research Initiative Technology Acceleration Program, grant numbers 171117 and 181472.

This work was supported by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

The authors would like to acknowledge Matt Sederberg for his support in editing and improving this work.

A A gentle introduction to U-splines

We recognize that for those new to this material, many of the concepts and definitions described in this work may be challenging to parse. Accordingly, in this section we introduce some of the more technical ideas through simple examples and practice problems, and their accompanying solutions. We recommend the reader work through these exercises to gain greater insight into the principles behind U-splines.

A.1 Building intuition: Constraints

The following exercises introduce basic concepts behind continuity constraints, described in detail in section 5 and appendix B.

Exercise 1

Consider two d-cells a and b on a one-dimensional mesh, separated by interface I as shown in fig. 61. Each cell is assigned a parameterization with parent coordinate $\xi \in [0, 1]$ running from left to right. The length of the parametric domain on cell a is ℓ^{a}, and the length of the parametric domain on cell b is ℓ^{b}. The mapping from the parent to the parametric domain on each cell is

$$\begin{aligned} s^{\mathsf{a}} &= \ell^{\mathsf{a}} \xi^{\mathsf{a}}, \\ s^{\mathsf{b}} &= \ell^{\mathsf{b}} \xi^{\mathsf{b}}. \end{aligned} \tag{138}$$

The degree assigned to cell a is p^{a}, and the degree assigned to cell b is p^{b}. Derive the continuity constraint on the nth derivative across interface I in homogeneous Bernstein form.

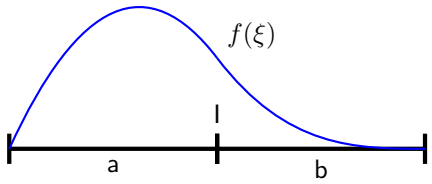


Figure 61: Two cells a and b separated by interface I on a one-dimensional mesh. An example function $f(\xi)$ spans the domains of both cells.

Answer to Exercise 1

The piecewise function $f(\xi)$ (see fig. 61) may be written in Bernstein form as

$$f(\xi) = \begin{cases} \sum\limits_{\mathsf{i}^{\mathsf{a}} \in \mathsf{ID}(\mathsf{a})} B_{\mathsf{i}^{\mathsf{a}}}^{p^{\mathsf{a}}}(\xi^{\mathsf{a}}) c_{\mathsf{i}^{\mathsf{a}}} & \text{for } \xi^{\mathsf{a}} \in \overline{\Omega}^{\mathsf{a}}, \\ \sum\limits_{\mathsf{i}^{\mathsf{b}} \in \mathsf{ID}(\mathsf{b})} B_{\mathsf{i}^{\mathsf{b}}}^{p^{\mathsf{b}}}(\xi^{\mathsf{b}}) c_{\mathsf{i}^{\mathsf{b}}} & \text{for } \xi^{\mathsf{b}} \in \overline{\Omega}^{\mathsf{b}}. \end{cases} \tag{139}$$

The continuity constraint on the nth derivative across the interface is given by:

$$\left(\frac{1}{\ell^{\mathsf{a}}}\right)^n \frac{d^n f(\xi^{\mathsf{a}})}{d{\xi^{\mathsf{a}}}^n}\bigg|_{\xi^{\mathsf{a}}=1} = \left(\frac{1}{\ell^{\mathsf{b}}}\right)^n \frac{d^n f(\xi^{\mathsf{b}})}{d{\xi^{\mathsf{b}}}^n}\bigg|_{\xi^{\mathsf{b}}=0} \tag{140}$$

which can be written in homogeneous Bernstein form, with respect to the parent coordinate system of each d-cell, as

$$\left(\frac{1}{\ell^{\mathsf{a}}}\right)^n \sum_{\mathsf{i}^{\mathsf{a}} \in \mathsf{ID}(\mathsf{a})} \frac{d^n B_{\mathsf{i}^{\mathsf{a}}}^{p^{\mathsf{a}}}(1)}{d{\xi^{\mathsf{a}}}^n} c_{\mathsf{i}^{\mathsf{a}}} - \left(\frac{1}{\ell^{\mathsf{b}}}\right)^n \sum_{\mathsf{i}^{\mathsf{b}} \in \mathsf{ID}(\mathsf{b})} \frac{d^n B_{\mathsf{i}^{\mathsf{b}}}^{p^{\mathsf{b}}}(0)}{d{\xi^{\mathsf{b}}}^n} c_{\mathsf{i}^{\mathsf{b}}} = 0 \quad \forall\, n \in [0, \vartheta^{\mathsf{I}}]. \tag{141}$$

Exercise 2

A four-element Bézier mesh with a bilinear $\mathcal{C}^0$ basis is shown in fig. 62. Use the provided function indexing to determine the constraint matrices for all four internal interfaces and the global constraint matrix. The interface I_0 is the bottom interface and the remaining interfaces are ordered in a counterclockwise fashion. What is the rank of the global constraint matrix?

10	11	14	15
8	9	12	13
2	3	6	7
0	1	4	5

Figure 62: A four-element Bézier mesh with a bilinear $\mathcal{C}^0$ basis. The Bernstein functions are labeled with the indexing used in the exercise.

Answer to Exercise 2

The interface constraint matrices are:

$$\boldsymbol{R}(\mathsf{I}_0) = \begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{142}$$

$$\boldsymbol{R}(\mathsf{I}_1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}, \tag{143}$$

$$\boldsymbol{R}(\mathsf{I}_2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix}, \tag{144}$$

$$\boldsymbol{R}(\mathsf{I}_3) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{145}$$

and the global constraint matrix is

$$\boldsymbol{R}(\mathsf{B}) = \begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{146}$$

which has rank seven. The rank of the associated nullspace is nine. In general, to find a minimal basis for the nullspace of a high-rank constraint system, we will identify subsystems that have rank one nullspaces (see exercises 7, 9 and 15). Observe that while the matrix has rank seven, it has eight rows which means that the constraints contain a linear dependence—a situation we often find on higher-dimensional meshes. Linear optimization approaches may be used to overcome this complication (see section 10.2).

Exercise 3

Determine the constraint matrix $\boldsymbol{R}(\mathsf{I})$ for the Bézier mesh shown in fig. 63.

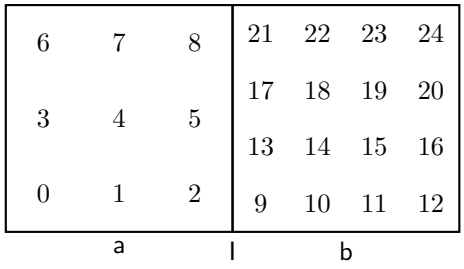


Figure 63: A Bézier mesh with two cells with degrees $\mathsf{p} = (2, 2)$ on the left and $\mathsf{p} = (3, 3)$ on the right, separated by a $\mathcal{C}^1$ interface I. The Bernstein functions are labeled with the indexing used in the exercise.

Answer to Exercise 3

The indices corresponding to the nonzero Bernstein coefficients in each constraint are shown in fig. 64 and the associated Bernstein coefficient values are shown in eq. (147).

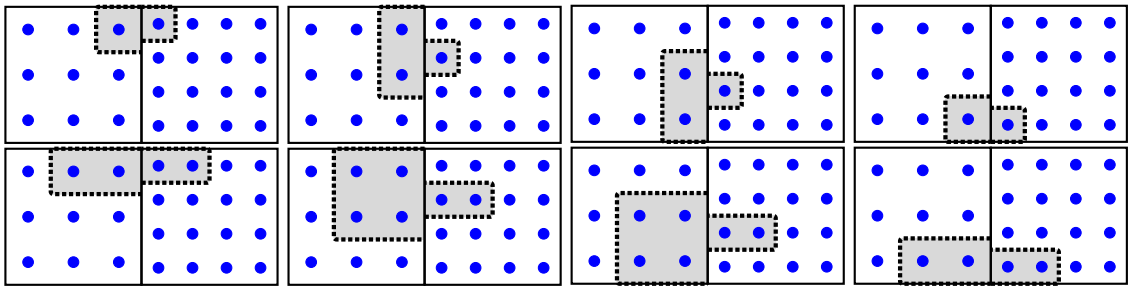


Figure 64: Sets of indices, highlighted in grey, corresponding to the Bernstein coefficients involved in the constraint equations for a $\mathcal{C}^1$ interface between two cells with degrees $\mathsf{p} = (2, 2)$ on the left and $\mathsf{p} = (3, 3)$ on the right. The top four index sets correspond to the four $\mathcal{C}^0$ constraint equations and the bottom four index sets correspond to the four $\mathcal{C}^1$ constraint equations. The values are given in eq. (147).

$$\boldsymbol{R}(\mathsf{I}) = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{2}{3} & 0 & 0 & -\frac{1}{3} & 0 & 0 & 0 & 0 \\
0 & 0 & -\frac{1}{3} & 0 & 0 & -\frac{2}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{4}{3} & -\frac{4}{3} & 0 & \frac{2}{3} & -\frac{2}{3} & 0 & 0 & 0 & 0 \\
0 & \frac{2}{3} & -\frac{2}{3} & 0 & \frac{4}{3} & -\frac{4}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 3 & 0 & 0
\end{bmatrix} \tag{147}$$

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 3 & 0 & 0 \\
0 & 0 & 0 & 0 & -3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\
-3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

A.2 Building intuition: Splines

The following exercises introduce basic concepts behind splines, spline bases, and the associated nullspace problem. These concepts were described in detail in section 6.

Exercise 4

Consider the two element Bézier mesh consisting of linear cells and a $\mathcal{C}^0$ constraint assigned to the interface I between them as shown in fig. 65. Solve for the basis vectors of the nullspace.

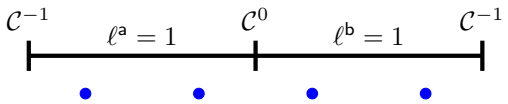


Figure 65: A two element linear Bézier mesh with a single $\mathcal{C}^0$ interface. The exercise is to solve for the basis vectors of the nullspace of this Bézier mesh.

Answer to Exercise 4

We start by defining the corresponding global constraint matrix

$$\boldsymbol{R}(\mathbf{B}) = \begin{bmatrix} 0 & 1 & -1 & 0 \end{bmatrix}. \tag{148}$$

A basis for the nullspace of this matrix is

$$\mathbf{UV}(\mathbf{B}) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{149}$$

Each coefficient in each vector corresponds to one of the Bernstein basis functions in the two element mesh.

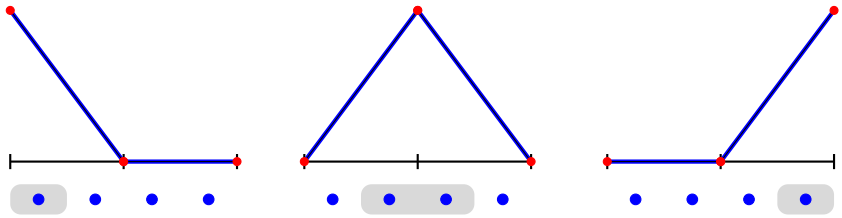


Figure 66: The nonzero coefficients of the basis vectors that span the nullspace of the Bézier mesh shown in fig. 65. The red dots represent the control points of the Bézier curve on each element.

Exercise 5

Consider the two element Bézier mesh shown in fig. 67. Compute the constraint matrix $\boldsymbol{R}(\mathbf{B})$ and verify that one possible basis for the nullspace of $\boldsymbol{R}(\mathbf{B})$ is given by the basis vectors

$$\mathbf{UV}(\mathbf{B}) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \frac{1}{2} \\ \frac{1}{4} \\ \frac{1}{4} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{2} \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{150}$$

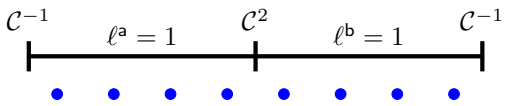


Figure 67: A two element cubic Bézier mesh with a single $\mathcal{C}^2$ interface. The exercise is to compute the constraint matrix of the nullspace of this Bézier mesh, and verify the basis given in eq. (150).

Answer to Exercise 5

The constraint matrix across the interface I can be shown to be

$$\boldsymbol{R}(\mathbf{B}) = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -3 & 3 & 3 & -3 & 0 & 0 \\ 0 & 6 & -12 & 6 & -6 & 12 & -6 & 0 \end{bmatrix}. \tag{151}$$

The basis vectors which represent the sparsest possible positive basis for the nullspace of $\boldsymbol{R}(\mathbf{B})$ are shown in fig. 68. This Bézier mesh generates a spline space which is equivalent to a B-spline with knot vector $\Xi = [0\ 0\ 0\ 0\ 1\ 2\ 2\ 2\ 2]$ [8, 54].

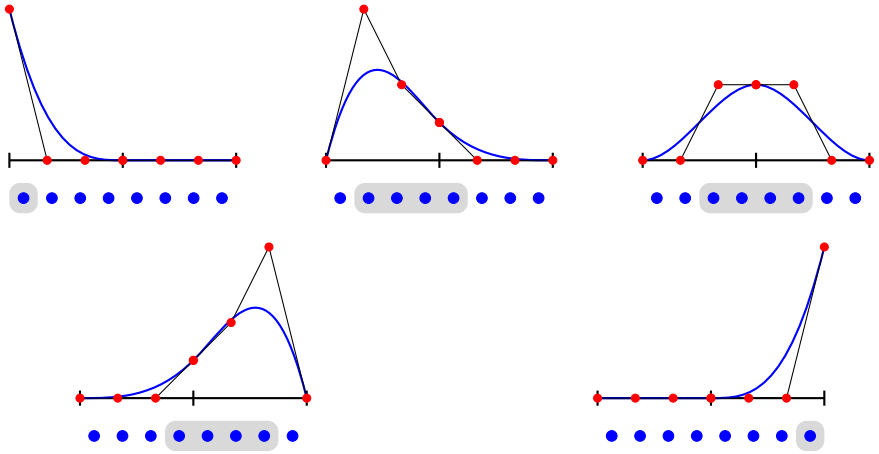


Figure 68: The nonzero coefficients of the basis vectors that span the nullspace of the Bézier mesh shown in fig. 67. The red dots represent the control points of the Bézier curve on each element.

Exercise 6

Compute the global extraction operator $\boldsymbol{C}$ for the knot vector $\Xi = [0\ 0\ 0\ 0\ 1\ 2\ 2\ 2\ 2]$ [8, 54].

Answer to Exercise 6

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{152}$$

Observe that the basis vectors computed in eq. (150) have the same coefficient values as the rows of the global extraction matrix shown in eq. (152). This demonstrates the connection between the spline basis and basis vectors for the nullspace.

A.3 Building intuition: Basis vectors

The following exercises introduce basic concepts behind basis vectors for k-cell nullspaces and the related rank one nullspace problem. These concepts were described in detail in section 8.

Exercise 7

Observe that the three interior basis vectors with multiple nonzero coefficients, shown in eq. (150), all have four nonzero entries and that these same coefficients can be obtained by solving the three reduced

problems

$$\boldsymbol{R}(\mathsf{I})_1 \boldsymbol{c}[\mathbf{B}]_1 = \mathbf{0}, \tag{153}$$
$$\boldsymbol{R}(\mathsf{I})_2 \boldsymbol{c}[\mathbf{B}]_2 = \mathbf{0}, \tag{154}$$
$$\boldsymbol{R}(\mathsf{I})_3 \boldsymbol{c}[\mathbf{B}]_3 = \mathbf{0}, \tag{155}$$

where the reduced smoothness constraint matrices are

$$\boldsymbol{R}(\mathsf{I})_1 = \begin{bmatrix} 0 & 0 & 1 & -1 \\ 0 & -3 & 3 & 3 \\ 6 & -12 & 6 & -6 \end{bmatrix}, \tag{156}$$
$$\boldsymbol{R}(\mathsf{I})_2 = \begin{bmatrix} 0 & 1 & -1 & 0 \\ -3 & 3 & 3 & -3 \\ -12 & 6 & -6 & 12 \end{bmatrix}, \tag{157}$$
$$\boldsymbol{R}(\mathsf{I})_3 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 3 & 3 & -3 & 0 \\ 6 & -6 & 12 & -6 \end{bmatrix}. \tag{158}$$

Each of these matrices has rank three which means that it has a rank one nullspace. These problems do not have unique solutions. One way to obtain a positive solution to these problems is to augment the constraint matrix so that one of the values is constrained to be one. Carry this procedure out for each of the reduced matrices and solve for the resulting vectors.

Answer to Exercise 7

The augmented matrix system for $\boldsymbol{R}(\mathsf{I})_1$ is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & -3 & 3 & 3 \\ 6 & -12 & 6 & -6 \end{bmatrix} \boldsymbol{c}[\mathbf{B}]_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{159}$$

The augmented matrix system for $\boldsymbol{R}(\mathsf{I})_2$ is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ -3 & 3 & 3 & -3 \\ -12 & 6 & -6 & 12 \end{bmatrix} \boldsymbol{c}[\mathbf{B}]_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{160}$$

The augmented matrix system for $\boldsymbol{R}(\mathsf{I})_1$ is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 3 & 3 & -3 & 0 \\ 6 & -6 & 12 & -6 \end{bmatrix} \boldsymbol{c}[\mathbf{B}]_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{161}$$

The solutions to these three augmented problems are

$$\boldsymbol{c}[\mathbf{B}]_1 = \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}, \boldsymbol{c}[\mathbf{B}]_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \boldsymbol{c}[\mathbf{B}]_3 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 4 \end{bmatrix}. \tag{162}$$

This solution, however, does not coincide with the nonzero values for the three central vectors in eq. (150). Because any scaling of the basis vectors is also a solution, we can define a new basis

$\boldsymbol{c}[\tilde{\mathbf{B}}]_1 = \boldsymbol{c}[\mathbf{B}]_1$, $\boldsymbol{c}[\tilde{\mathbf{B}}]_2 = \frac{1}{2}\boldsymbol{c}[\mathbf{B}]_2$, and $\boldsymbol{c}[\tilde{\mathbf{B}}]_3 = \frac{1}{4}\boldsymbol{c}[\mathbf{B}]_3$:

$$\boldsymbol{c}[\tilde{\mathbf{B}}]_1 = \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}, \boldsymbol{c}[\tilde{\mathbf{B}}]_2 = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \boldsymbol{c}[\tilde{\mathbf{B}}]_3 = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{2} \\ 1 \end{bmatrix}. \tag{163}$$

The values of this new basis now coincide with the nonzero values of the middle three basis vectors given in eq. (150).

Exercise 8

Why is the augmentation procedure, where one additional equation is added that has exactly one nonzero entry, sufficient to solve a rank one nullspace problem?

Answer to Exercise 8

Each of the reduced constraint matrices has rank three but there are four undetermined coefficients. This means that each of the problems is a rank one nullspace problem; in other words, the nullspace is one-dimensional. Any rank one nullspace problem can be converted into a full rank linear system by augmenting the system with one additional equation (as long as the added equation is linearly independent from the original equations). The additional equation raises the system rank to four and so we can obtain a unique solution. Any solution obtained will also be a solution to the original system; additionally, any nonzero scaling of a solution to the augmented system will also be a solution of the original reduced system.

Exercise 9

Describe the geometry of a rank one nullspace problem where the dimension of the constraint space $\mathcal{C}$ is m and the dimension of the Bernstein space $\mathcal{B}$ is n. Graphically convey these ideas for the rank one nullspace of the basis vector on the interface of the linear mesh given in exercise 4 in appendix A.2.

Answer to Exercise 9

The rank of the nullspace of the full constraint matrix is $n - m$, but the rank of the nullspace of each reduced constraint matrix is always one. This rank one nullspace is orthogonal to the constraint space and contained in $\mathcal{B}$. Since this nullspace is a linear subspace of $\mathcal{B}$ of dimension one it is a line that passes through the origin of $\mathcal{B}$. The orientation of the line is such that it is orthogonal to the constraint space. The requirement that the basis vector, which spans the nullspace, must be sparse means that the chosen basis vector, which is not unique, is as close to orthogonal to $\mathcal{B}$ as possible. The requirement that all entries be positive means that the chosen basis vector is in the non-negative orthant of $\mathbb{R}^n$.

The Bernstein space of the one-dimensional Bézier mesh in fig. 65 has dimension $n = 4$ and the constraint space has dimension $m = 1$, resulting in a nullspace of dimension 3. The basis vector on the $\mathcal{C}^0$ interface has the reduced constraint matrix

$$\boldsymbol{R}(\mathbf{B}) = \begin{bmatrix} 1 & -1 \end{bmatrix}, \tag{164}$$

which spans a line, seen as the red line in in fig. 69. This line is the constraint space. The nullspace problem associated with this matrix can be written as $x - y = 0$, the solution of which is any vector $\begin{bmatrix} x & y \end{bmatrix}^T \in \mathbb{R}^2$ which passes through the origin and is orthogonal to the red line (the constraint space). Thus, a positive normalized basis vector spanning this nullspace is $\begin{bmatrix} 1 & 1 \end{bmatrix}^T$, seen as a blue vector in fig. 69, and any scaling of which is a solution to this nullspace problem.

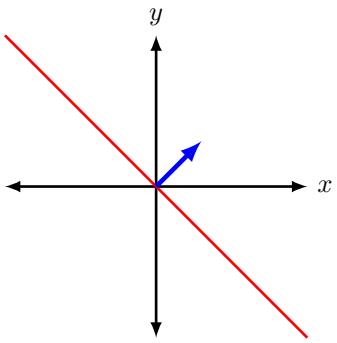


Figure 69: The constraint space on the $\mathcal{C}^0$ interface of fig. 65 in exercise 4 is represented by the red line. Any vector that is orthogonal to this line is a member of the nullspace.

Exercise 10

Write down the $\mathcal{C}^0$ constraint matrix for the system of cells shown in fig. 70 and draw the nonzero entries for each constraint.

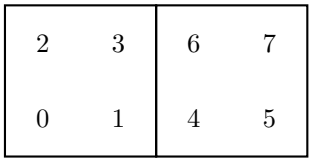


Figure 70: A two-element Bézier mesh with a bilinear basis and a $\mathcal{C}^0$ interface. The Bernstein functions are labeled with the indexing used in the exercise.

Answer to Exercise 10

The coefficients corresponding to Greville points that are adjacent across the interface must be equal. This knowledge leads to the constraint matrix:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix}. \tag{165}$$

The indices associated with nonzero coefficients in the constraint system are shown outlined in gray boxes with dashed outlines. The two boxes represent the two rows in the matrix.

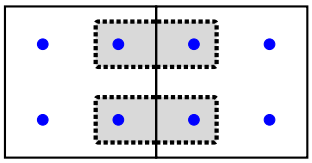


Figure 71: Coefficients corresponding to the nonzero entries in the two rows of the constraint matrix. Each grey block surrounds the nonzero entries of one row.

The sparsest basis vectors for the constraint matrix are obtained by setting a single value to one and solving all coupled constraints.

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{166}$$

All constraints rows have only two nonzero entries and so there are two vectors with two values (two-value constraint rows are equality constraints). A representation of the basis vectors is shown in fig. 72. The markers corresponding to indices of the nonzero entries are surrounded with gray boxes; each gray box corresponds to one basis vector.

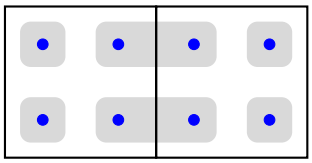


Figure 72: Each gray box marks the nonzero entries in one basis vector.

Exercise 11

Determine the constraint matrix and the index sets for the basis vectors for the quadratic-linear two cell example with a $\mathcal{C}^1$ interface shown in fig. 73.

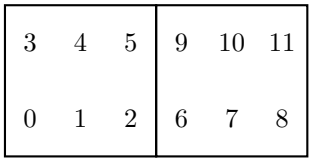


Figure 73: A two-element Bézier mesh with a quadratic-linear basis and a $\mathcal{C}^1$ interface. The Bernstein functions are labeled with the indexing used in the exercise.

Answer to Exercise 11

The constraint matrix on the $\mathcal{C}^1$ interface is

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & -2 & 2 & 0 \end{bmatrix}. \tag{167}$$

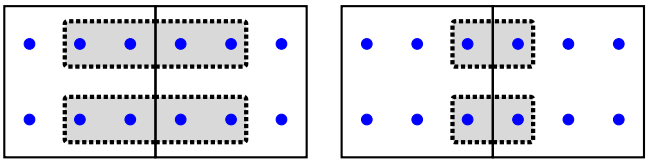


Figure 74: Coefficients corresponding to the nonzero entries in the four rows of the constraint matrix. Each grey block surrounds the nonzero entries of one row: the $\mathcal{C}^1$ constraints on the left, and the $\mathcal{C}^0$ constraints on the right.

The basis vectors associated with this constraint matrix are

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{2} \\ 0 \\ 0 \\ 0 \\ \frac{1}{2} \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2} \\ 0 \\ 0 \\ 0 \\ \frac{1}{2} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \frac{1}{2} \\ 0 \\ 0 \\ 0 \\ \frac{1}{2} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ \frac{1}{2} \\ 0 \\ 0 \\ 0 \\ \frac{1}{2} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{168}$$

The markers corresponding to indices of the nonzero entries are surrounded with gray boxes in fig. 75. Each gray box corresponds to one basis vector.

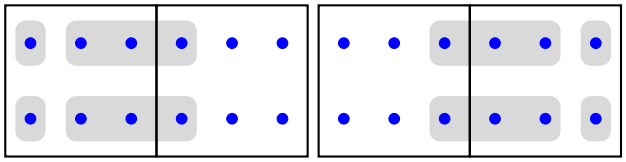


Figure 75: Each gray box marks the nonzero entries in one basis vector.

Exercise 12

Determine the basis vectors with more than one nonzero entry for each interface constraint matrix in eqs. (142) to (145) corresponding to the interfaces in fig. 62, and draw a picture highlighting the nonzero entries for each vector.

Answer to Exercise 12

The required basis vectors for $\boldsymbol{R}(\mathsf{I}_0)$ are

$$\boldsymbol{v}(a)^{\mathsf{I}_0} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{169}$$

$$\boldsymbol{v}(b)^{\mathsf{I}_0} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{170}$$

The required basis vectors for $\boldsymbol{R}(\mathsf{I}_1)$ are

$$\boldsymbol{v}(a)^{\mathsf{I}_1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T, \tag{171}$$

$$\boldsymbol{v}(b)^{\mathsf{I}_1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T. \tag{172}$$

The required basis vectors for $\boldsymbol{R}(\mathsf{I}_2)$ are

$$\boldsymbol{v}(a)^{\mathsf{I}_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T, \tag{173}$$

$$\boldsymbol{v}(b)^{\mathsf{I}_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}^T. \tag{174}$$

The required basis vectors for $\boldsymbol{R}(\mathsf{I}_3)$ are

$$\boldsymbol{v}(a)^{\mathsf{I}_3} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{175}$$

$$\boldsymbol{v}(b)^{\mathsf{l}_3} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{176}$$

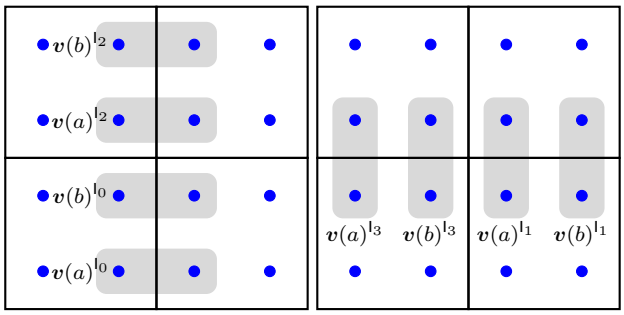


Figure 76: The nonzero coefficients of the individual edge basis vectors are surrounded by gray boxes.

A.4 Building intuition: The U-spline mesh

A U-spline mesh is a Bézier mesh which conforms to a set of admissibility conditions, as described in section 9. These admissibility conditions rely on a concept called a *ribbon*. The following exercise introduces the basic ideas behind the construction of a ribbon.

Exercise 13

Construct a ribbon of maximum coupling length on the mesh in fig. 77, originating at the red Bernstein index near the vertex labeled with a small hollow square and proceeding to the right. Determine whether the ribbon is truncated.

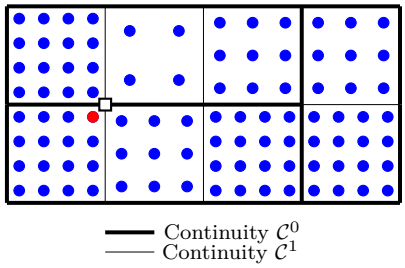


Figure 77: An example two-dimensional Bézier mesh with mixed degree and mixed continuity. The exercise is to construct a ribbon of maximum coupling length, beginning at the vertex adjacent to the red Bernstein index and extending to the right.

Answer to Exercise 13

First, we label the origin vertex o, head interface I^h, and all potential ribbon tail interfaces I_i, extending to the right of o as seen in fig. 78. Then, a one-dimensional Bézier mesh is constructed that corresponds to the minimum parallel degree adjacent to each potential ribbon interface (including the head), and the maximum perpendicular continuity on each $(d-2)$-cell in the ribbon skeleton, seen in fig. 80. The ribbon is then constructed as described in algorithm 2. This procedure is analogous to chaining a sequence of one-dimensional vertex basis vectors end-to-end, starting at the initial Bernstein index, and overlapping only on their boundary as seen in fig. 80. The interfaces in the ribbon tail correspond to the elements in the one-dimensional Bézier mesh which have at least one Bernstein index in a basis vector, excluding the head (see fig. 79). In fig. 81 we observe that this ribbon is truncated. The condition for truncation described in algorithm 2 is true for a ribbon which would have been longer had the maximum perpendicular continuity on the final $(d-2)$-cell in the ribbon skeleton been greater than or equal to $\mathcal{C}^{p-1}$.

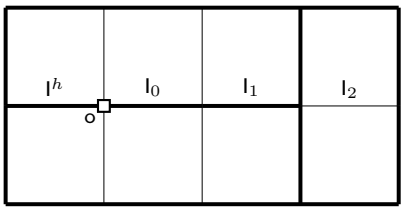


Figure 78: The ribbon origin $\circ$, ribbon head I^h, and potential ribbon tail interfaces I_i are labeled.

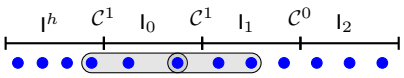


Figure 80: The minimum parallel degrees and maximum perpendicular continuities adjacent to the potential interfaces of the ribbon are used to form a one-dimensional Bézier mesh. The coupling length (and thus the length of the ribbon) is measured as described in algorithm 2.

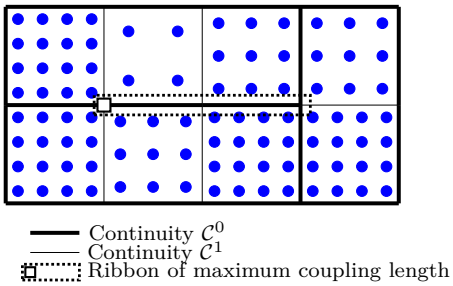


Figure 79: The completed ribbon of maximum coupling length. The interfaces of the tail include $\boldsymbol{t} = [\mathsf{I}_0, \mathsf{I}_1]$.

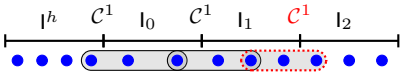


Figure 81: By temporarily setting the final vertex on the one-dimensional mesh to continuity $\mathcal{C}^{p-1}$, we observe that this ribbon is truncated.

A.5 Building intuition: The U-spline basis

The following exercises introduce basic concepts that are utilized in the algorithm for constructing a U-spline basis, described in detail in section 10.

Exercise 14

The constraint systems of two interfaces I_1 and I_2 are said to interact if the product of the matrices of absolute values has nonzero entries. Let $A_{ij} = |[\boldsymbol{R}(\mathsf{I}_1)]_{ij}|$ and $B_{ij} = |[\boldsymbol{R}(\mathsf{I}_2)]_{ij}|$ then the constraints associated with I_1 and I_2 interact if there exists at least one entry in the matrix

$$D_{ij} = A_{ik}B_{jk} \tag{177}$$

that is greater than zero. Determine the constraint matrices for the two interfaces $\boldsymbol{R}(\mathsf{I}_1)$ and $\boldsymbol{R}(\mathsf{I}_2)$ and $\boldsymbol{R}(\mathbf{U})$ for the U-spline mesh shown in fig. 82.

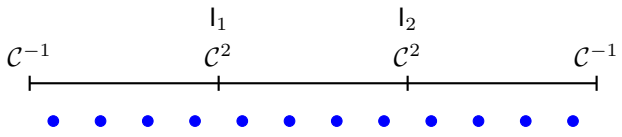


Figure 82: Three cells with cubic bases on a one-dimensional Bézier mesh, separated by $\mathcal{C}^2$ interfaces. The exercise is to compute the constraint matrices for the two interfaces on this mesh.

Answer to Exercise 14

For convenience we express the constraint matrices in terms of the global indexing of the Bernstein bases on each element. The constraint matrices are

$$\boldsymbol{R}(\mathsf{I}_1) = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 3 & 3 & -3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & -12 & 6 & -6 & 12 & -6 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{178}$$

$$\boldsymbol{R}(\mathsf{I}_2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3 & 3 & 3 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & -12 & 6 & -6 & 12 & -6 & 0 \end{bmatrix} \tag{179}$$

$$\begin{aligned} \boldsymbol{R}(\mathbf{U}) &= \begin{bmatrix} \boldsymbol{R}(\mathsf{I}_1) \\ \boldsymbol{R}(\mathsf{I}_2) \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 3 & 3 & -3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & -12 & 6 & -6 & 12 & -6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3 & 3 & 3 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & -12 & 6 & -6 & 12 & -6 & 0 \end{bmatrix}. \end{aligned} \tag{180}$$

We can see that the interaction matrix defined in eq. (177) has nonzero entries and so the constraint systems interact. We now seek to determine the number of nonzero coefficients required to construct a basis vector for the coupled system (see exercise 15).

Exercise 15

We showed that the number of nonzero contiguous Bernstein coefficients required for a basis vector on an interface in the univariate case can be determined from the continuity of the interface, specifically $\vartheta^{\mathsf{I}} + 2$. Restricting the constraint system to the coefficients that are known to be nonzero results in a rank one system. Consider all the contiguous index sets of size 4 from interface I_1 and I_2 in fig. 82. Consider all pairings of these sets where one comes from I_1 and the other comes from I_2 such that the combined index set is contiguous. Construct the matrices corresponding to these sets of nonzero coefficients and determine the rank of the nullspace for each one.

Answer to Exercise 15

The index sets for I_1 are $\{1,2,3,4\}$, $\{2,3,4,5\}$, and $\{3,4,5,6\}$. The index sets for I_2 are $\{5,6,7,8\}$, $\{6,7,8,9\}$, and $\{7,8,9,10\}$. The paired contiguous index sets $a = \{1,2,3,4,5,6,7,8\}$, $b = \{2,3,4,5,6,7,8,9\}$, $c = \{3,4,5,6,7,8,9,10\}$ have corresponding matrices

$$\boldsymbol{R}(a) = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -3 & 3 & 3 & -3 & 0 & 0 & 0 \\ 6 & -12 & 6 & -6 & 12 & -6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -3 & 3 & 3 \\ 0 & 0 & 0 & 0 & 6 & -12 & 6 & -6 \end{bmatrix} \quad \boldsymbol{R}(b) = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -3 & 3 & 3 & -3 & 0 & 0 & 0 & 0 \\ -12 & 6 & -6 & 12 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -3 & 3 & 3 & -3 \\ 0 & 0 & 0 & 6 & -12 & 6 & -6 & 12 \end{bmatrix} \tag{181}$$

$$\boldsymbol{R}(c) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 3 & -3 & 0 & 0 & 0 & 0 & 0 \\ 6 & -6 & 12 & -6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -3 & 3 & 3 & -3 & 0 \\ 0 & 0 & 6 & -12 & 6 & -6 & 12 & -6 \end{bmatrix} \tag{182}$$

which each have a rank 2 nullspace. The paired contiguous index sets $d = \{2, 3, 4, 5, 6, 7, 8\}$ and $e = \{3, 4, 5, 6, 7, 8, 9\}$ have corresponding matrices

$$\boldsymbol{R}(d) = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ -3 & 3 & 3 & -3 & 0 & 0 & 0 \\ -12 & 6 & -6 & 12 & -6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -3 & 3 & 3 \\ 0 & 0 & 0 & 6 & -12 & 6 & -6 \end{bmatrix} \quad \boldsymbol{R}(e) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 3 & -3 & 0 & 0 & 0 & 0 \\ 6 & -6 & 12 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -3 & 3 & 3 & -3 \\ 0 & 0 & 6 & -12 & 6 & -6 & 12 \end{bmatrix} \tag{183}$$

which each have a rank 1 nullspace. Finally, the paired contiguous index set $f = \{3, 4, 5, 6, 7, 8\}$ with corresponding matrix

$$\boldsymbol{R}(f) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 3 & 3 & -3 & 0 & 0 & 0 \\ 6 & -6 & 12 & -6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -3 & 3 & 3 \\ 0 & 0 & 6 & -12 & 6 & -6 \end{bmatrix} \tag{184}$$

has a rank 0 nullspace. The six matrices shown in this exercise hint at an important result. The index sets corresponding to basis vectors for adjacent systems can be used to construct index sets for the coupled system in certain cases. The index sets $d = \{2, 3, 4, 5, 6, 7, 8\}$ and $e = \{3, 4, 5, 6, 7, 8, 9\}$ generate systems with rank 1 nullspaces while all others do not. In both cases the index sets for the interface basis vector from each system overlap at a single index.

This turns out to be general. The index set for a contiguous basis vector can be generated by choosing one basis vector from each interface constraint system such that the index overlap between basis vectors from adjacent systems is a single index. We can express this as a flood that is seeded with a basis vector from a single interface system and continues adding basis vectors from interface constraint systems that share a single index overlap with previously included basis vectors until no more basis vectors satisfy the overlap requirement. This is a useful result in the univariate case and it can be shown to agree with previously published work on the construction of mixed-degree splines [57, 65, 58, 64]. This result also motivates the algorithm described in section 10 for constructing contiguous basis vectors in higher-dimensions, where the minimal overlap of neighboring vertex basis vectors is used to construct the support of a U-spline basis function.

B Interface continuity constraints in two dimensions

We consider constructing constraints on the interface between two cells on a two-dimensional U-spline mesh. We consider three cases: the interface between two quadrilateral cells, between a quadrilateral and triangle cell, and between two triangle cells. Without loss of generality, in each case we express these constraints with respect to the interface-aligned parameterizations indicated by the axes in fig. 83, fig. 84, and fig. 85. We recognize that in practice, arbitrary relative rotations of the parameterizations on the adjacent cells must be accounted for, but for simplicity of exposition we assume the adjacent cells have the aligned coordinate systems as indicated.

We also assume the existence of a degree-elevation operator $\boldsymbol{D}$ that maps the coefficients $\boldsymbol{c}$ of a Bernstein

polynomial of degree p to a new vector of coefficients $\bar{\boldsymbol{c}}$ of a Bernstein polynomial of degree $q > p$, thus representing the original function with a new basis of degree q. Thus,

$$\bar{\boldsymbol{c}} = \boldsymbol{D}^{p,q}\boldsymbol{c} \tag{185}$$

where the nonzero entries of the matrix operator are given by

$$D_{ij}^{p,q} = \frac{\binom{q-p}{i-j}\binom{p}{j}}{\binom{q}{i}} \quad \begin{matrix} i \in [0,q] \\ j \in [\max(0, i-q+p), \min(p,i)]. \end{matrix} \tag{186}$$

B.1 Quadrilateral-quadrilateral interface

Consider two quadrilateral cells a and b on a two-dimensional mesh, separated by interface I as shown in fig. 83. Each cell has been assigned a box-like parameterization with parent coordinates $\xi_i \in [0,1], i \in \{0,1\}, \boldsymbol{\xi} \in \overline{\Omega}$. These cells have their parameterizations oriented such that the parameters ${\xi_0}^{\mathsf{a}}$ on cell a and the parameter ${\xi_0}^{\mathsf{b}}$ on cell b each point parallel the interface (although in opposite directions). Let $\eta \in [0,1]$ parameterize the shared interface I, and be defined as $\eta = {\xi_0}^{\mathsf{a}} = (1 - {\xi_0}^{\mathsf{b}})$. The lengths of the parametric domain on cell a are $\boldsymbol{\ell}^{\mathsf{a}} = [{\ell_0}^{\mathsf{a}}, {\ell_1}^{\mathsf{a}}]$, and the lengths of the parametric domain on cell b are $\boldsymbol{\ell}^{\mathsf{b}} = \left[{\ell_0}^{\mathsf{b}}, {\ell_1}^{\mathsf{b}}\right]$. The mapping from the parent to the parametric domain in each parametric direction on each cell is

$$\begin{aligned} {s_0}^{\mathsf{a}} &= {\ell_0}^{\mathsf{a}}{\xi_0}^{\mathsf{a}} \\ {s_1}^{\mathsf{a}} &= {\ell_1}^{\mathsf{a}}{\xi_1}^{\mathsf{a}} \\ {s_0}^{\mathsf{b}} &= {\ell_0}^{\mathsf{b}}{\xi_0}^{\mathsf{b}} \\ {s_1}^{\mathsf{b}} &= {\ell_1}^{\mathsf{b}}{\xi_1}^{\mathsf{b}} \end{aligned} \tag{187}$$

$s_i \in [0, \ell_i], i \in \{0,1\}, \boldsymbol{s} \in \hat{\Omega}$. The degree assigned to cell a is $\mathsf{p}^{\mathsf{a}} = ({p_0}^{\mathsf{a}}, {p_1}^{\mathsf{a}})$, and the degree assigned to cell b is $\mathsf{p}^{\mathsf{b}} = \left({p_0}^{\mathsf{b}}, {p_1}^{\mathsf{b}}\right)$.

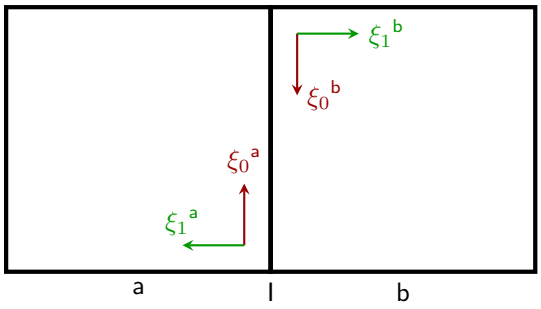


Figure 83: Two cells separated by interface I on a two-dimensional mesh. We consider building the continuity constraint on the derivative of order n on this interface.

A piecewise polynomial with support over a and b can be written in Bernstein form as

$$f(\boldsymbol{\xi}) = \begin{cases} \sum_{i=0}^{{p_0}^{\mathsf{a}}} \sum_{j=0}^{{p_1}^{\mathsf{a}}} B_i^{{p_0}^{\mathsf{a}}}({\xi_0}^{\mathsf{a}}) B_j^{{p_1}^{\mathsf{a}}}({\xi_1}^{\mathsf{a}}) c_{(i,j)^{\mathsf{a}}} & \text{for } \boldsymbol{\xi}^{\mathsf{a}} \in \overline{\Omega}^{\mathsf{a}} \\ \sum_{i=0}^{{p_0}^{\mathsf{b}}} \sum_{j=0}^{{p_1}^{\mathsf{b}}} B_i^{{p_0}^{\mathsf{b}}}({\xi_0}^{\mathsf{b}}) B_j^{{p_1}^{\mathsf{b}}}({\xi_1}^{\mathsf{b}}) c_{(i,j)^{\mathsf{b}}} & \text{for } \boldsymbol{\xi}^{\mathsf{b}} \in \overline{\Omega}^{\mathsf{b}} \end{cases}. \tag{188}$$

We can write the continuity constraint on the derivative of order n across the interface as

$$\left(\frac{1}{{\ell_1}^{\mathsf{a}}}\right)^n \sum_{i=0}^{{p_0}^{\mathsf{a}}} \sum_{j={p_1}^{\mathsf{a}}-n}^{{p_1}^{\mathsf{a}}} B_i^{{p_0}^{\mathsf{a}}}(\eta) \frac{d^n B_j^{{p_1}^{\mathsf{a}}}(0)}{d{\xi_1}^{\mathsf{a}\,n}} c_{(i,j)^{\mathsf{a}}} = \left(\frac{-1}{{\ell_1}^{\mathsf{b}}}\right)^n \sum_{i=0}^{{p_0}^{\mathsf{b}}} \sum_{j=0}^{n} B_i^{{p_0}^{\mathsf{b}}}(\eta) \frac{d^n B_j^{{p_1}^{\mathsf{b}}}(0)}{d{\xi_1}^{\mathsf{b}\,n}} c_{(i,j)^{\mathsf{b}}} \quad \forall\, \eta \in \overline{\Omega}^{\mathsf{I}} \tag{189}$$

and if ${p_0}^{\mathsf{a}} = {p_0}^{\mathsf{b}} = p^{\parallel}$ we can enforce the above equality constraint by instead enforcing

$$\left(\frac{1}{{\ell_1}^{\mathsf{a}}}\right)^n \sum_{j={p_1}^{\mathsf{a}}-n}^{{p_1}^{\mathsf{a}}} \frac{d^n B_j^{{p_0}^{\mathsf{a}}}(0)}{d{\xi_1}^{\mathsf{a}^n}} c_{(i,j)^{\mathsf{a}}} = \left(\frac{-1}{{\ell_1}^{\mathsf{b}}}\right)^n \sum_{j=0}^{n} \frac{d^n B_j^{{p_0}^{\mathsf{b}}}(0)}{d{\xi_1}^{\mathsf{b}^n}} c_{(p^{\parallel}-i,j)^{\mathsf{b}}} \quad \forall\, i \in [0, p^{\parallel}]. \tag{190}$$

In the case that ${p_0}^{\mathsf{a}} \neq {p_0}^{\mathsf{b}}$, we define $p^{\parallel}_{\max}(\mathsf{I}) = \max({p_0}^{\mathsf{a}}, {p_0}^{\mathsf{b}})$. The degree-elevation operator $\boldsymbol{D}$ is used here to map the respective Bernstein coefficents to a degree $p^{\parallel}_{\max}(\mathsf{I})$ Bernstein space on the shared interface. Substituting in and rearranging yields the constraints in homogeneous form

$$\left(\frac{1}{{\ell_1}^{\mathsf{a}}}\right)^n \sum_{\substack{i\in[0,{p_0}^{\mathsf{a}}] \\ j\in[{p_1}^{\mathsf{a}}-n,{p_1}^{\mathsf{a}}]}} \frac{d^n B_j^{{p_1}^{\mathsf{a}}}(0)}{d{\xi_1}^{\mathsf{a}^n}} D_{k,i}^{{p_0}^{\mathsf{a}},p^{\parallel}_{\max}(\mathsf{I})} c_{(i,j)^{\mathsf{a}}} - \left(\frac{-1}{{\ell_1}^{\mathsf{b}}}\right)^n \sum_{\substack{i\in[0,{p_0}^{\mathsf{b}}] \\ j\in[0,n]}} \frac{d^n B_j^{{p_1}^{\mathsf{b}}}(0)}{d{\xi_1}^{\mathsf{b}^n}} D^{{p_0}^{\mathsf{b}},p^{\parallel}_{\max}(\mathsf{I})}_{p^{\parallel}_{\max}(\mathsf{I})-k,i} c_{(i,j)^{\mathsf{b}}} = 0 \quad \forall \begin{matrix} k \in [0, p^{\parallel}_{\max}(\mathsf{I})] \\ n \in [0, \vartheta^{\mathsf{I}}]. \end{matrix} \tag{191}$$

An example of the nonzero constraint equation coefficients on an interface between two quadrilaterals of degree $\mathsf{p} = (2,2)$ and $\mathsf{p} = (3,3)$ may be seen in fig. 64, with the full constraint matrix listed in eq. (147).

B.2 Quadrilateral-triangle interface

Consider a quadrilateral cell a and a triangle cell b on a two-dimensional mesh, separated by interface I as shown in fig. 84. The quadrilateral cell has been assigned a box-like parameterization with parent coordinates ${\xi_i}^{\mathsf{a}} \in [0,1], i \in \{0,1\}$, and the triangle cell has been assigned a barycentric parameterization with parent coordinates ${\xi_i}^{\mathsf{b}} \in [0,1], i \in \{0,1,2\}$. These cells have their parameterizations oriented such that the parameter ${\xi_0}^{\mathsf{a}}$ on the quadrilateral and the parameter ${\xi_1}^{\mathsf{b}}$ on the triangle each point parallel the interface (although in opposite directions). Let $\eta \in [0,1]$ parameterize the shared interface I, and be defined as $\eta = {\xi_0}^{\mathsf{a}} = (1 - {\xi_1}^{\mathsf{b}})$. The degree assigned to cell a is $\mathsf{p}^{\mathsf{a}} = ({p_0}^{\mathsf{a}}, {p_1}^{\mathsf{a}})$, and the degree assigned to cell b is p^{b}. All barycentric index tuples in the triangle cell b are expressed with two indices i and j corresponding to barycentric coordinates ${\xi_1}^{\mathsf{b}}$ and ${\xi_2}^{\mathsf{b}}$. The remaining index corresponding to ${\xi_0}^{\mathsf{b}}$ is omitted, but is implicitly defined to be equal to $p^{\mathsf{b}} - i - j$.

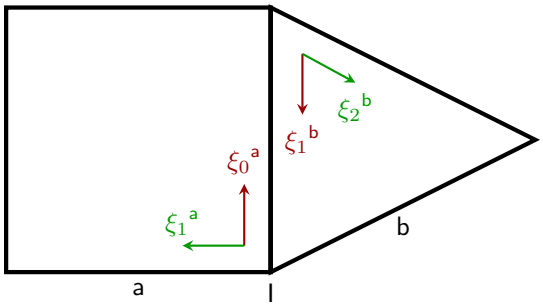


Figure 84: A quadrilateral cell and a triangle cell separated by interface I on a two-dimensional mesh. We consider building the continuity constraint on the derivative of order 0 on this interface.

A piecewise polynomial with support over a and b can be written in Bernstein form as

$$f(\boldsymbol{\xi}) = \begin{cases} \sum_{j=0}^{{p_1}^{\mathsf{a}}} \sum_{i=0}^{{p_0}^{\mathsf{a}}} B_i^{{p_0}^{\mathsf{a}}}({\xi_0}^{\mathsf{a}}) B_j^{{p_1}^{\mathsf{a}}}({\xi_1}^{\mathsf{a}}) c_{(i,j)^{\mathsf{a}}} & \text{for } \boldsymbol{\xi}^{\mathsf{a}} \in \overline{\Omega}^{\mathsf{a}} \\ \sum_{j=0}^{p^{\mathsf{b}}} \sum_{i=0}^{p^{\mathsf{b}}-j} B_{i,j}^{p^{\mathsf{b}}}({\xi_0}^{\mathsf{b}}, {\xi_1}^{\mathsf{b}}, {\xi_2}^{\mathsf{b}}) c_{(i,j)^{\mathsf{b}}} & \text{for } \boldsymbol{\xi}^{\mathsf{b}} \in \overline{\Omega}^{\mathsf{b}} \end{cases}. \tag{192}$$

For an interface that is adjacent to a triangle, we are only required to consider the constraint on the derivative of order 0. We can write the continuity constraint on the derivative of order 0 across the interface as

$$\sum_{i=0}^{{p_0}^{\mathsf{a}}} B_i^{{p_0}^{\mathsf{a}}}(\eta) B_0^{{p_1}^{\mathsf{a}}}(0) c_{(i,0)^{\mathsf{a}}} = \sum_{i=0}^{p^{\mathsf{b}}} B_{i,0}^{p^{\mathsf{b}}}(\eta, 1-\eta, 0) c_{(i,0)^{\mathsf{b}}} \quad \forall\, \eta \in \overline{\Omega}^{\mathsf{I}} \tag{193}$$

and if ${p_0}^{\mathsf{a}} = p^{\mathsf{b}} = p^{\parallel}$ we can enforce the above equality constraint by instead enforcing

$$c_{(i,0)^{\mathsf{a}}} = c_{\left(p^{\parallel}-i,0\right)^{\mathsf{b}}} \quad \forall\, i \in [0, p^{\parallel}]. \tag{194}$$

In the case that ${p_0}^{\mathsf{a}} \neq p^{\mathsf{b}}$, we define $p^{\parallel}_{\max}(\mathsf{I}) = \max({p_0}^{\mathsf{a}}, p^{\mathsf{b}})$. The degree-elevation operator $\boldsymbol{D}$ is used here to map the respective Bernstein coefficients to a degree $p^{\parallel}_{\max}(\mathsf{I})$ Bernstein space on the shared interface. Substituting in and rearranging yields the constraints in homogeneous form

$$\sum_{i \in [0, {p_0}^{\mathsf{a}}]} D_{k,i}^{{p_0}^{\mathsf{a}}, p^{\parallel}_{\max}(\mathsf{I})} c_{(i,0)^{\mathsf{a}}} - \sum_{i \in [0, p^{\mathsf{b}}]} D^{p^{\mathsf{b}}, p^{\parallel}_{\max}(\mathsf{I})}_{p^{\parallel}_{\max}(\mathsf{I})-k,i} c_{(i,0)^{\mathsf{b}}} = 0 \quad \forall\, k \in [0, p^{\parallel}_{\max}(\mathsf{I})]\ . \tag{195}$$

B.3 Triangle-triangle interface

Consider two triangle cells a and b on a two-dimensional mesh, separated by interface I as shown in fig. 85. Each triangle cell has been assigned a barycentric parameterization with parent coordinates ${\xi_i}^{\mathsf{a}} \in [0,1], i \in \{0,1,2\}$ and ${\xi_i}^{\mathsf{b}} \in [0,1], i \in \{0,1,2\}$. These cells have their parameterizations oriented such that the parameters ${\xi_1}^{\mathsf{a}}$ on cell a and the parameter ${\xi_1}^{\mathsf{b}}$ on cell b each point parallel the interface (although in opposite directions). Let $\eta \in [0,1]$ parameterize the shared interface I, and be defined as $\eta = {\xi_0}^{\mathsf{a}} = (1 - {\xi_1}^{\mathsf{b}})$. The degree assigned to cell a is p^{a}, and the degree assigned to cell b is p^{b}. All barycentric index tuples on the two triangle cells are expressed with two indices i and j corresponding to barycentric coordinates ξ_1 and ξ_2. The remaining index corresponding to ξ_0 is omitted, but is implicitly defined to be equal to $p - i - j$ (where p is the degree assigned to the respective triangle cell).

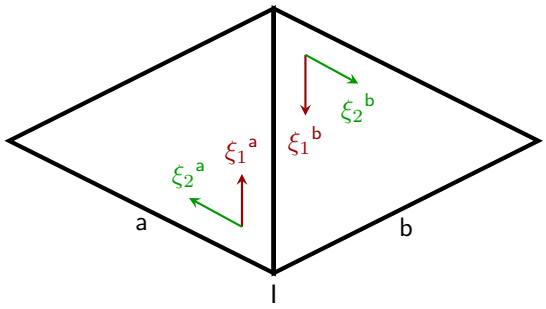


Figure 85: Two triangular cells separated by interface I on a two-dimensional mesh. We consider building the continuity constraint on the derivative of order 0 on this interface.

A piecewise polynomial with support over a and b can be written in Bernstein form as

$$f(\boldsymbol{\xi}) = \begin{cases} \sum_{j=0}^{p^{\mathsf{a}}} \sum_{i=0}^{p^{\mathsf{a}}-j} B_{i,j}^{p^{\mathsf{a}}}({\xi_0}^{\mathsf{a}}, {\xi_1}^{\mathsf{a}}, {\xi_2}^{\mathsf{a}}) c_{(i,j)^{\mathsf{a}}} & \text{for } \boldsymbol{\xi}^{\mathsf{a}} \in \overline{\Omega}^{\mathsf{a}} \\ \sum_{j=0}^{p^{\mathsf{b}}} \sum_{i=0}^{p^{\mathsf{b}}-j} B_{i,j}^{p^{\mathsf{b}}}({\xi_0}^{\mathsf{b}}, {\xi_1}^{\mathsf{b}}, {\xi_2}^{\mathsf{b}}) c_{(i,j)^{\mathsf{b}}} & \text{for } \boldsymbol{\xi}^{\mathsf{b}} \in \overline{\Omega}^{\mathsf{b}}. \end{cases} \tag{196}$$

For an interface that is adjacent to a triangle, we are only required to consider the constraint on the derivative of order 0. We can write the continuity constraint on the derivative of order 0 across the interface as

$$\sum_{i=0}^{p^{\mathsf{a}}} B_{p^{\mathsf{a}}-i,0}^{p^{\mathsf{a}}}(1-\eta, \eta, 0) c_{(p^{\mathsf{a}}-i,0)^{\mathsf{a}}} = \sum_{i=0}^{p^{\mathsf{b}}} B_{i,0}^{p^{\mathsf{b}}}(\eta, 1-\eta, 0) c_{(i,0)^{\mathsf{b}}} \quad \forall\, \eta \in \overline{\Omega}^{\mathsf{I}} \tag{197}$$

and if $p^{\mathsf{a}} = p^{\mathsf{b}} = p^{\parallel}$ we can enforce the above equality constraint by instead enforcing

$$c_{(i,0)^{\mathsf{a}}} = c_{\left(p^{\parallel}-i,0\right)^{\mathsf{b}}} \quad \forall\, i \in [0, p^{\parallel}]. \tag{198}$$

In the case that $p^{\mathsf{a}} \neq p^{\mathsf{b}}$, we define $p^{\parallel}_{\max}(\mathsf{I}) = \max(p^{\mathsf{a}}, p^{\mathsf{b}})$. The degree-elevation operator $\boldsymbol{D}$ is used here to map the respective Bernstein coefficients to a degree $p^{\parallel}_{\max}(\mathsf{I})$ Bernstein space on the shared interface. Substituting in and rearranging yields the constraints in homogeneous form

$$\sum_{i\in[0,p^{\mathsf{a}}]} D^{p^{\mathsf{a}},p^{\parallel}_{\max}(\mathsf{I})}_{k,i} c_{(p^{\mathsf{a}}-i,0)^{\mathsf{a}}} - \sum_{i\in[0,p^{\mathsf{b}}]} D^{p^{\mathsf{b}},p^{\parallel}_{\max}(\mathsf{I})}_{p^{\parallel}_{\max}(\mathsf{I})-k,i} c_{(i,0)^{\mathsf{b}}} = 0 \quad \forall\, k \in [0, p^{\parallel}_{\max}(\mathsf{I})]\,. \tag{199}$$

C Basis vectors in arbitrary dimensions

We now describe how to buid basis vectors on a k-cell from a d-dimensional Bézier mesh, $0 \leq k \leq d$. The description is recursive, and we begin with the base case: the basis vectors on a d-cell are the Bernstein functions which span the Bernstein space assigned to the element. Then, the basis vectors on a k-cell a^k, $0 \leq k < d$ are constructed as follows.

C.1 Composite k-cell basis vectors

Composite k-cell basis vectors are formed from multiple adjacent $(k+1)$-cell basis vectors. Each composite k-cell basis vector is associated with a choice of inclusion distances $\mathsf{INC}^{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}$ (section 8.3.2) and alignment index $\mathsf{i} \in \mathsf{ID}(\mathsf{a}^k)$ (section 8.3.3). An example of selecting a choice of inclusion distances is shown in fig. 23.

We begin by placing indexed submesh domains denoted Ω^{ab} over each set of elements adjacent to each $(k+1)$-cell adjacent to a^k (with their origins set to $\mathsf{v} \in \mathsf{ADJ}^0(\mathsf{a}^k)$), and then partitioning the mapped index sets of the basis vectors associated with each $(k+1)$-cell b^{k+1} into equivalence classes with respect to the parallel equivalence relation on the $(k+1)$-cell $\mathbf{BG}(\mathsf{b}^{k+1})/\varpi^{\parallel}_{\mathsf{b}^{k+1}}$. We then identify the equivalence classes for which the minimum projection onto the $(k+1)$-cell is less than or equal to $\mathsf{INC}^{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}_{\mathsf{b}^{k+1}}$:

$$\mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}(\mathsf{b}^{k+1}) = \Big\{\mathbf{EBG}(\mathsf{b}^{k+1}) \in \mathbf{BG}(\mathsf{b}^{k+1})/\varpi^{\parallel}_{\mathsf{b}^{k+1}} \;:\; \min_{\mathsf{g}\in\mathsf{G}} \left(\left[\pi^{\parallel}_{\mathsf{b}^{k+1}}(\mathsf{g})\right]_{s_r}\right) \leq \mathsf{INC}^{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}_{\mathsf{b}^{k+1}}, \mathsf{G} \in \mathbf{EBG}(\mathsf{b}^{k+1})\Big\} \tag{200}$$

where $s_r = \boldsymbol{s}^{\parallel}_{\Omega^{\mathsf{ab}}}(\mathsf{b}^{k+1}) \setminus \boldsymbol{s}^{\parallel}_{\Omega^{\mathsf{ab}}}(\mathsf{a}^k)$ is the parameter coordinate parallel to b^{k+1} and perpendicular to a^k. An example of these equivalence classes is shown in fig. 24.

Let $\mathsf{c}^{k+1}_{\perp,j} \in \mathbf{PC}(\mathsf{a}^k, \mathsf{b}^{k+1}), j \in \{1, \ldots, \left|\mathbf{PC}(\mathsf{a}^k, \mathsf{b}^{k+1})\right|\}$. ($\mathbf{PC}$ is defined in section 3.1.1.) We form the sets containing all indices whose associated cell is adjacent to b^{k+1} and whose associated submesh Greville point is a part of elements of $\mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}(\mathsf{c}^{k+1}_{\perp,j})$

$$\mathsf{ID}^{\perp}_{\mathsf{c}^{k+1}_{\perp,j}} = \left\{\mathsf{i} \in \mathsf{ID}(\mathsf{E}) \;:\; \mathsf{E} \in \mathsf{ADJ}^d(\mathsf{b}^{k+1}), \mathsf{g}(\mathsf{i}) \in \mathsf{G} \in \mathbf{EBG} \in \mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}(\mathsf{c}^{k+1}_{\perp,j})\right\}. \tag{201}$$

We form the set of Greville points that are fixed points of the parallel projectors onto $\mathsf{c}^{k+1}_{\perp,j}$:

$$\mathsf{G}^{\perp} = \bigcup_{j\in\{1,\ldots,|\mathbf{PC}(\mathsf{a}^k,\mathsf{b}^{k+1})|\}} \left\{\mathsf{g}(\mathsf{i}) \;:\; \mathsf{g}(\mathsf{i}) = \pi^{\perp}_{\mathsf{b}^{k+1}}(\mathsf{g}(\mathsf{i})), \mathsf{i} \in \mathsf{ID}^{\perp}_{\mathsf{c}^{k+1}_{\perp,j}}\right\}. \tag{202}$$

We take all basis vectors whose projections onto the $(k+1)$-cells perpendicular to b^{k+1} lie in $\mathsf{G}^{\perp}$:

$$\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}(\mathsf{b}^{k+1}) = \left\{\mathsf{G} \in \mathbf{EBG} \in \mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}(\mathsf{b}^{k+1}) \;:\; \forall_{\mathsf{g}\in\mathsf{G}}\; \pi^{\perp}_{\mathsf{b}^{k+1}}(\mathsf{g}) \subseteq \mathsf{G}^{\perp}\right\}. \tag{203}$$

In the case that $\mathbf{PC}(\mathsf{a}^k, \mathsf{b}^{k+1}) = \varnothing$ then $\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}(\mathsf{b}^{k+1}) = \mathbf{EBG}^{\parallel}_{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}(\mathsf{b}^{k+1})$. In fig. 25, we see an example of these basis vector subsets marked with dotted lines.

The set of indices associated with $\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}(\mathsf{b}^{k+1})$ is

$$\mathsf{ID}^{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}_{\mathsf{b}^{k+1}} = \bigcup_{\mathsf{G}\in\mathbf{BG}^{\perp}_{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}(\mathsf{b}^{k+1})} \left\{ \mathsf{i}\in\mathsf{ID}(\mathsf{E}) \,:\, \mathsf{E}\in\mathsf{ADJ}^{d}(\mathsf{b}^{k+1}), \mathsf{g}(\mathsf{i})\in\mathsf{G} \right\}. \tag{204}$$

The full set of indices associated with $\mathsf{inc}_j, j\in\{1,\ldots,d-k\}$ is

$$\mathsf{ID}^{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}_{\mathsf{a}^{k}} = \bigcup_{\mathsf{b}^{k+1}\in\mathsf{ADJ}^{k+1}(\mathsf{a}^k)} \mathsf{ID}^{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}_{\mathsf{b}^{k+1}}. \tag{205}$$

Given an alignment index $\mathsf{i}\in\mathsf{ID}(\mathsf{a}^k)$ and a choice of $\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}$, we can now define the index set for the composite k-cell basis vector as

$$\mathsf{ID}^{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k},\mathsf{i}}_{\mathsf{a}^{k}} = \mathsf{ID}(\mathbf{HBV}_{\mathsf{i}}(\mathsf{a}^k)) \cap \mathsf{ID}^{\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}}_{\mathsf{a}^{k}}. \tag{206}$$

We consider all possible alignment indices $\mathsf{i}\in\mathsf{ID}(\mathsf{a}^k)$ and all possible values of $\mathsf{inc}_1,\ldots,\mathsf{inc}_{d-k}$ on a^k to construct the set of all composite k-cell basis vectors. We use $\mathbf{BV}''(\mathsf{a}^k)$ to represent this set. The full set of indices contained in this set is

$$\mathsf{UID}_{\mathsf{a}^k} = \bigcup_{\boldsymbol{n}\in\mathbf{BV}''(\mathsf{a}^k)} \mathsf{ID}(\boldsymbol{n}). \tag{207}$$

An example of a set of composite vertex basis vectors is shown in fig. 26.

C.2 Simple k-cell basis vectors

Each *simple* k-cell basis vector is formed from a single $(k+1)$-cell basis vector on an adjacent $(k+1)$-cell, one for each $(k+1)$-cell basis vector whose index set is not subsumed by $\mathsf{UID}_{\mathsf{a}^k}$. We use $\mathbf{BV}'(\mathsf{a}^k)$ to represent this set:

$$\mathbf{BV}'(\mathsf{a}^k) = \bigcup_{\mathsf{b}^{k+1}\in\mathsf{ADJ}^{k+1}(\mathsf{a}^k)} \left\{ \boldsymbol{n}\in\mathbf{BV}(\mathsf{b}^{k+1}) \,:\, \mathsf{ID}(\boldsymbol{n}) \not\subseteq \mathsf{UID}_{\mathsf{a}^k} \right\}. \tag{208}$$

An example of a set of simple vertex basis vectors is shown in fig. 27.

C.3 The full set of k-cell basis vectors

The full set of k-cell basis vectors is found by combining the set of *composite* k-cell basis vectors with the set of *simple* k-cell basis vectors:

$$\mathbf{BV}(\mathsf{a}^k) = \mathbf{BV}''(\mathsf{a}^k) \cup \mathbf{BV}'(\mathsf{a}^k). \tag{209}$$

Once the index set for each k-cell basis vector has been obtained, the constraint system can be solved to obtain the coefficient values and construct the basis vector.

D Ribbon processing

A ribbon of maximum coupling length is constructed on a mesh beginning at an origin $(d-2)$-cell o adjacent to an initial Bernstein coefficient, and then adding interfaces I_j one by one beginning at the origin $(d-2)$-cell to form the tail $\boldsymbol{t}$. We use $|\boldsymbol{t}|$ to represent the number of interfaces present in the tail $\boldsymbol{t}$. As shown in figs. 86 and 87, to determine the length of the tail (or coupling distance) $|\boldsymbol{t}|$ we process a sequence of connected interfaces where the continuity assigned to each traversed $(d-2)$-cell is set to $\vartheta_j^{\max} = \max(\vartheta_j^0, \vartheta_j^1)$ and the degree of each traversed interface is set to $p_{\mathsf{I}_j}^{\min} = p^{\parallel,\perp}_{\min}(\mathsf{I}_j, \mathsf{w}), \mathsf{w}\in\mathsf{ADJ}^{d-2}(\mathsf{I}_j)\cap\mathsf{skel}(\mathsf{r})$.

Conceptually, this process extracts a one-dimensional U-spline mesh (where the vertices and edges of the one-dimensional mesh correspond to the $(d-2)$-cells and interfaces of the parent mesh), and determines how far the specified Bernstein coefficient couples with the neighboring coefficients assigned to the edges in the one-dimensional mesh given the continuity constraints assigned to each vertex. Algorithm 2 describes the procedure used to determine the edges in the ribbon tail leveraging the notation introduced in figs. 86 and 87.

The input parameters of the procedure include the interface at the head of the ribbon I^h, the origin vertex o, and $\mathsf{i}_{\mathsf{I}^h}$ is the index of the initial Bernstein coefficient in I^h. Several examples of ribbons of maximum coupling length are shown in fig. 88.

A ribbon of length $|\boldsymbol{t}|$ is said to be truncated if $\vartheta^{\max}_{|\boldsymbol{t}|} < p^{\min}_{[\boldsymbol{t}]_{|\boldsymbol{t}|-1}} - 1$ and the value of the Bernstein index i computed for the final interface $[\boldsymbol{t}]_{|\boldsymbol{t}|-1}$ is greater than zero, as shown in algorithm 2. The bottom two examples in fig. 88 show truncated ribbons.

See appendix A.4 for an intuitive example of constructing a ribbon of maximum coupling length.

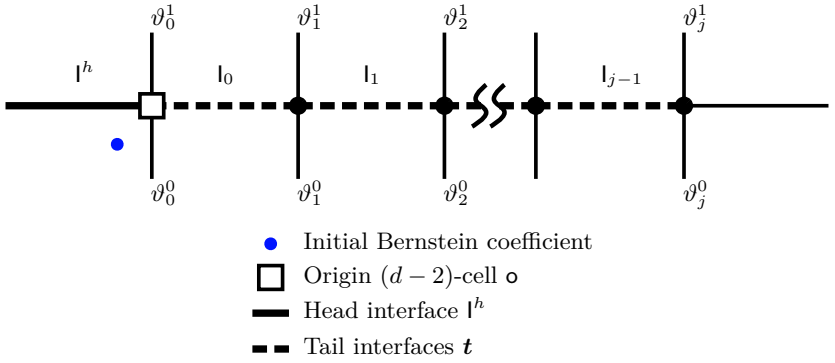


Figure 86: The features of a ribbon of maximum coupling length on a two-dimensional U-spline mesh. The perpendicular interfaces on the jth vertex are assigned continuities ϑ^0_j and ϑ^1_j.

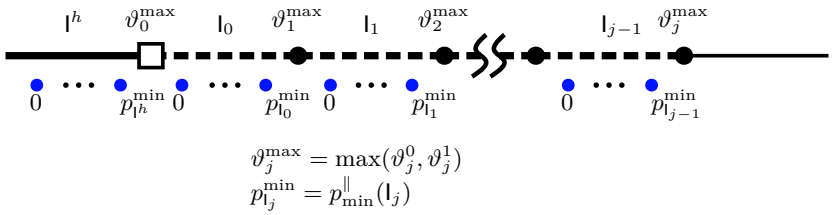


Figure 87: A ribbon that has been collapsed into a one-dimensional U-spline mesh. The continuity assigned to each vertex is the maximum of the perpendicular interfaces in the two-dimensional mesh. The degree of each one-dimensional cell is the minimum of the parallel degrees of the adjoining cells from the two-dimensional mesh, in the direction parallel to the ribbon. The Bernstein coefficients are labeled as they are indexed in algorithm 2.

Algorithm 2 Build a ribbon r of maximum coupling length, starting at head interface I^h, origin $(d-2)$-cell o, and initial Bernstein index $\mathsf{i}_{\mathsf{I}^h}$ (see fig. 86 and fig. 87 for notation).

procedure BUILDRIBBONOFMAXIMUMCOUPLINGLENGTH(I^h, o, $\mathsf{i}_{\mathsf{I}^h}$) ▷ $0 \le \mathsf{i}_{\mathsf{I}^h} \le p^{\min}_{\mathsf{I}^h}$
  $\boldsymbol{t} \leftarrow [\,]$ ▷ The array that will contain the interfaces in the tail.
  $j \leftarrow 0$ ▷ Counter for the length of the tail.
  $\mathsf{trunc} \leftarrow \mathsf{False}$ ▷ If the ribbon is found to be truncated, this will be set to true later.
  $\mathsf{i} \leftarrow \mathsf{i}_{\mathsf{I}^h}$ ▷ The Bernstein index counter used to determine the length of the ribbon.
  $\mathsf{I}_{-1} \leftarrow \mathsf{I}^h$ ▷ The index used in the loop to reference the head interface.
  $\mathsf{w}_0 \leftarrow \mathsf{o}$ ▷ The index used in the loop to reference the origin $(d-2)$-cell.
  // Loop until the maximum coupling length is reached.
  loop
    $\mathsf{i}_{\mathsf{prev}} \leftarrow \mathsf{i}$ ▷ Save the Bernstein index on interface I_{j-1} for later reference.
    // Get the max smoothness on the interfaces perpendicular to the ribbon on w_j.
    $\vartheta^{\max}_j \leftarrow \max(\vartheta^0_j, \vartheta^1_j)$ ▷ In other words, $\vartheta^{\max}_j \leftarrow \max_{\mathsf{I}' \in \mathsf{ADJ}^{d-1}(\mathsf{w}_j) \setminus \{\mathsf{I}_{j-1}, \mathsf{I}_j\}}(\vartheta^{\mathsf{I}'})$.
    $p^{\min}_{j-1} \leftarrow p^{\parallel}_{\min}(\mathsf{I}_{j-1})$ ▷ Get the min parallel degree on the previous interface.
    $\mathsf{i} \leftarrow \mathsf{i} + \vartheta^{\max}_j - p^{\min}_{j-1}$ ▷ We compute the Bernstein index for interface I_j.
    if $\mathsf{i} \ge 0$ **then**
      $[\boldsymbol{t}]_j \leftarrow \mathsf{I}_j$ ▷ If the Bernstein index for I_j is valid, then we add I_j to the tail.
      $j \leftarrow j + 1$ ▷ Increment the number of interfaces in the tail.
    else
      // Here, $\mathsf{i}_{\mathsf{prev}}$ refers to the value of i on $[\boldsymbol{t}]_{|\boldsymbol{t}|-1}$, the final interface of the tail $\boldsymbol{t}$.
      if $\mathsf{i}_{\mathsf{prev}} > 0$ **then**
        // Since $\mathsf{i}_{\mathsf{prev}} > 0$, the only way i on interface I_j to be less than 0 is for the
        // final interface to have reduced continuity.
        $\mathsf{trunc} \leftarrow \mathsf{True}$
      end if
      break ▷ Break out of the loop.
    end if
  end loop
  $\mathsf{r} \leftarrow \{\mathsf{I}^h, \mathsf{o}, \boldsymbol{t}\}$ ▷ Assemble the ribbon.
  return r, trunc ▷ Return the result.
end procedure

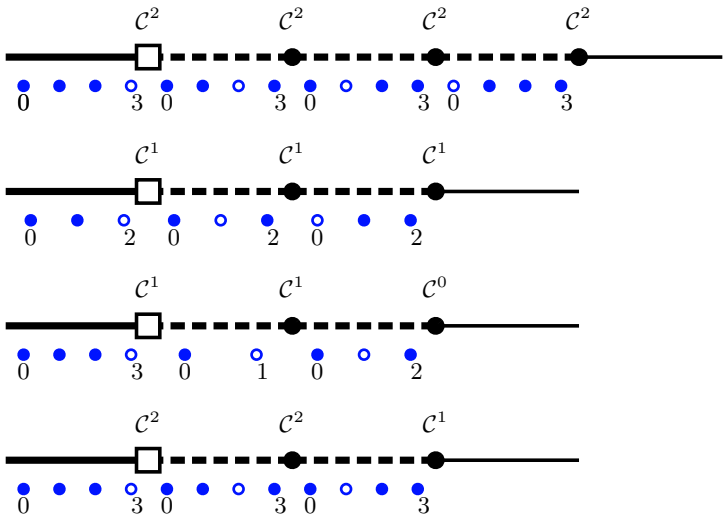


Figure 88: Several examples of the construction of ribbons of maximum coupling length. The Bernstein index marked by a hollow circle in each cell is the index referenced by $\mathsf{i}_{\mathsf{prev}}$ in each loop of algorithm 2. Note that the bottom two ribbons are truncated.

E U-spline test cases with Bézier extraction coefficients

E.1 U-spline extraction coefficients near a supersmooth interface

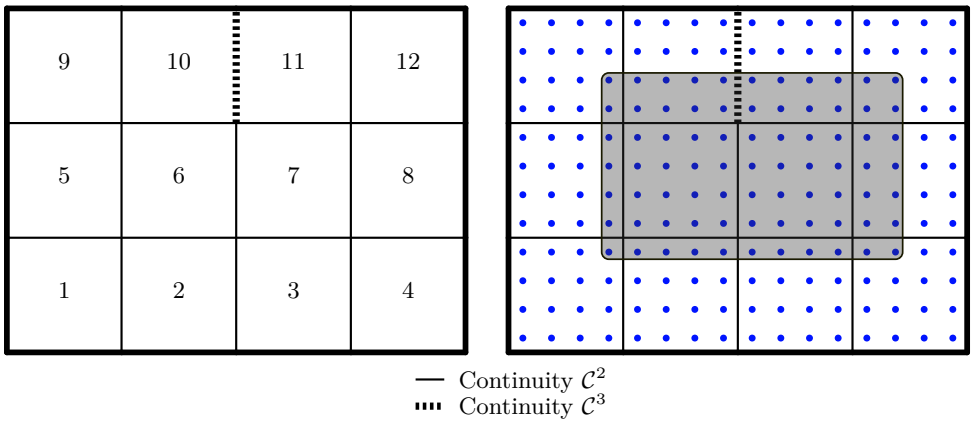


Figure 89: A four-by-three cubic U-spline mesh with $\mathcal{C}^2$ continuity on all interior edges except for one which has $\mathcal{C}^3$ continuity, forming a supersmooth interface. The Bernstein coefficients of the highlighted basis function are listed in table 2.

$(3,3)^1$	$(0,3)^2$	$(1,3)^2$	$(2,3)^2$	$(3,3)^2$	$(0,3)^3$	$(1,3)^3$	$(2,3)^3$	$(3,3)^3$	$(0,3)^4$
$\frac{3}{88}$	$\frac{3}{88}$	$\frac{3}{44}$	$\frac{3}{22}$	$\frac{2}{11}$	$\frac{2}{11}$	$\frac{5}{22}$	$\frac{1}{4}$	$\frac{17}{88}$	$\frac{17}{88}$
$(1,3)^4$	$(3,0)^5$	$(3,1)^5$	$(3,2)^5$	$(3,3)^5$	$(0,0)^6$	$(1,0)^6$	$(2,0)^6$	$(3,0)^6$	$(0,1)^6$
$\frac{3}{22}$	$\frac{3}{88}$	$\frac{3}{44}$	$\frac{3}{22}$	$\frac{21}{176}$	$\frac{3}{88}$	$\frac{3}{44}$	$\frac{3}{22}$	$\frac{2}{11}$	$\frac{3}{44}$
$(1,1)^6$	$(2,1)^6$	$(3,1)^6$	$(0,2)^6$	$(1,2)^6$	$(2,2)^6$	$(3,2)^6$	$(0,3)^6$	$(1,3)^6$	$(2,3)^6$
$\frac{3}{22}$	$\frac{3}{11}$	$\frac{4}{11}$	$\frac{3}{22}$	$\frac{3}{11}$	$\frac{6}{11}$	$\frac{8}{11}$	$\frac{21}{176}$	$\frac{21}{88}$	$\frac{21}{44}$
$(3,3)^6$	$(0,0)^7$	$(1,0)^7$	$(2,0)^7$	$(3,0)^7$	$(0,1)^7$	$(1,1)^7$	$(2,1)^7$	$(3,1)^7$	$(0,2)^7$
$\frac{7}{11}$	$\frac{2}{11}$	$\frac{5}{22}$	$\frac{1}{4}$	$\frac{17}{88}$	$\frac{4}{11}$	$\frac{5}{11}$	$\frac{1}{2}$	$\frac{17}{44}$	$\frac{8}{11}$
$(1,2)^7$	$(2,2)^7$	$(3,2)^7$	$(0,3)^7$	$(1,3)^7$	$(2,3)^7$	$(3,3)^7$	$(0,0)^8$	$(1,0)^8$	$(0,1)^8$
$\frac{10}{11}$	1	$\frac{17}{22}$	$\frac{7}{11}$	$\frac{35}{44}$	$\frac{7}{8}$	$\frac{119}{176}$	$\frac{17}{88}$	$\frac{3}{22}$	$\frac{17}{44}$
$(1,1)^8$	$(0,2)^8$	$(1,2)^8$	$(0,3)^8$	$(1,3)^8$	$(3,0)^9$	$(3,1)^9$	$(0,0)^{10}$	$(1,0)^{10}$	$(2,0)^{10}$
$\frac{3}{11}$	$\frac{17}{22}$	$\frac{6}{11}$	$\frac{119}{176}$	$\frac{21}{44}$	$\frac{21}{176}$	$\frac{9}{88}$	$\frac{21}{176}$	$\frac{21}{88}$	$\frac{21}{44}$
$(3,0)^{10}$	$(0,1)^{10}$	$(1,1)^{10}$	$(2,1)^{10}$	$(3,1)^{10}$	$(0,0)^{11}$	$(1,0)^{11}$	$(2,0)^{11}$	$(3,0)^{11}$	$(0,1)^{11}$
$\frac{7}{11}$	$\frac{9}{88}$	$\frac{9}{44}$	$\frac{9}{22}$	$\frac{6}{11}$	$\frac{7}{11}$	$\frac{35}{44}$	$\frac{7}{8}$	$\frac{119}{176}$	$\frac{6}{11}$
$(1,1)^{11}$	$(2,1)^{11}$	$(3,1)^{11}$	$(0,0)^{12}$	$(1,0)^{12}$	$(0,1)^{12}$	$(1,1)^{12}$			
$\frac{15}{22}$	$\frac{3}{4}$	$\frac{51}{88}$	$\frac{119}{176}$	$\frac{21}{44}$	$\frac{51}{88}$	$\frac{9}{22}$			

Table 2: The values of the nonzero Bernstein coefficients of the U-spline basis function highlighted in fig. 89.

E.2 U-spline extraction coefficients with non-rectangular support

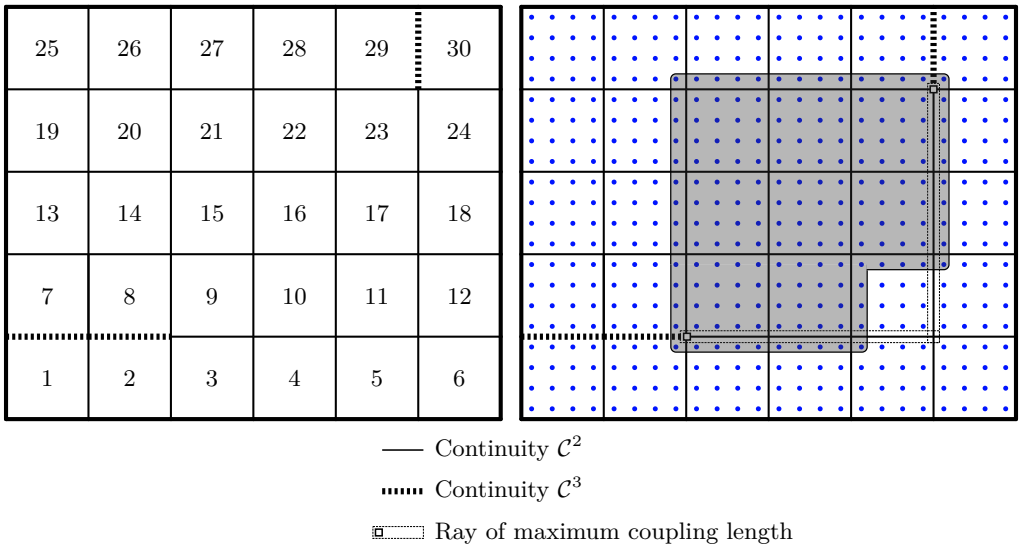
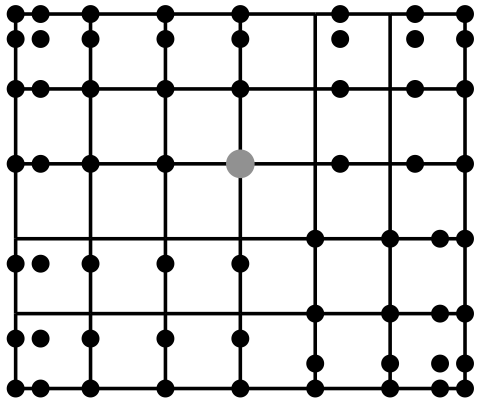
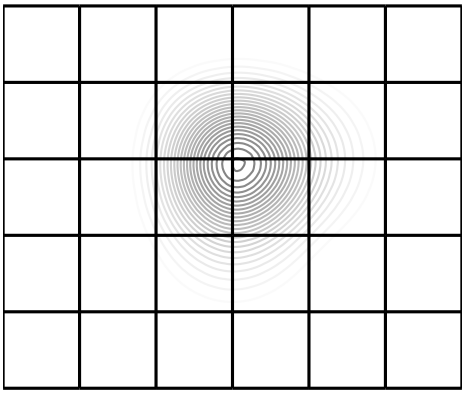


Figure 90: An example of a U-spline basis function that has a non-rectangular shape due to the close proximity of two continuity transitions near supersmooth interfaces. Notice the two perpendicular transition ribbons, starting at the vertices adjacent to the supersmooth interfaces, which touch at a common endpoint. This is an example of a basis function which is not possible with T-splines, which require basis functions to have a tensor product structure. The Bernstein coefficients of the highlighted basis function are listed in table 3.



(a) The control points for a linear parameterization of the mesh in fig. 90. The highlighted control point corresponds to the highlighted basis function.



(b) Contour plot of the basis function whose nonzero Bernstein coefficients are highlighted in fig. 90.

Figure 91: Control points and basis function contours for the U-spline in fig. 90.

$(3,3)^2$	$(0,3)^3$	$(1,3)^3$	$(2,3)^3$	$(3,3)^3$	$(0,3)^4$	$(1,3)^4$	$(2,3)^4$	$(3,3)^4$	$(0,3)^5$
$\frac{1}{80}$	$\frac{1}{80}$	$\frac{1}{40}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{80}$	$\frac{1}{80}$
$(3,0)^8$	$(3,1)^8$	$(3,2)^8$	$(3,3)^8$	$(0,0)^9$	$(1,0)^9$	$(2,0)^9$	$(3,0)^9$	$(0,1)^9$	$(1,1)^9$
$\frac{1}{80}$	$\frac{1}{40}$	$\frac{1}{20}$	$\frac{1}{10}$	$\frac{1}{80}$	$\frac{1}{40}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{20}$
$(2,1)^9$	$(3,1)^9$	$(0,2)^9$	$(1,2)^9$	$(2,2)^9$	$(3,2)^9$	$(0,3)^9$	$(1,3)^9$	$(2,3)^9$	$(3,3)^9$
$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{20}$	$\frac{1}{10}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{33}{80}$
$(0,0)^{10}$	$(1,0)^{10}$	$(2,0)^{10}$	$(3,0)^{10}$	$(0,1)^{10}$	$(1,1)^{10}$	$(2,1)^{10}$	$(3,1)^{10}$	$(0,2)^{10}$	$(1,2)^{10}$
$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{80}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{5}$	$\frac{1}{5}$
$(2,2)^{10}$	$(3,2)^{10}$	$(0,3)^{10}$	$(1,3)^{10}$	$(2,3)^{10}$	$(3,3)^{10}$	$(0,0)^{11}$	$(0,1)^{11}$	$(0,2)^{11}$	$(0,3)^{11}$
$\frac{1}{10}$	$\frac{1}{20}$	$\frac{33}{80}$	$\frac{17}{40}$	$\frac{1}{4}$	$\frac{3}{20}$	$\frac{1}{80}$	$\frac{1}{40}$	$\frac{1}{20}$	$\frac{3}{20}$
$(1,3)^{11}$	$(2,3)^{11}$	$(3,3)^{11}$	$(0,3)^{12}$	$(3,0)^{14}$	$(3,1)^{14}$	$(3,2)^{14}$	$(3,3)^{14}$	$(0,0)^{15}$	$(1,0)^{15}$
$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{80}$	$\frac{1}{80}$	$\frac{1}{10}$	$\frac{3}{20}$	$\frac{9}{40}$	$\frac{17}{80}$	$\frac{1}{10}$	$\frac{1}{5}$
$(2,0)^{15}$	$(3,0)^{15}$	$(0,1)^{15}$	$(1,1)^{15}$	$(2,1)^{15}$	$(3,1)^{15}$	$(0,2)^{15}$	$(1,2)^{15}$	$(2,2)^{15}$	$(3,2)^{15}$
$\frac{2}{5}$	$\frac{33}{80}$	$\frac{3}{20}$	$\frac{3}{10}$	$\frac{3}{5}$	$\frac{5}{8}$	$\frac{9}{40}$	$\frac{9}{20}$	$\frac{9}{10}$	$\frac{19}{20}$
$(0,3)^{15}$	$(1,3)^{15}$	$(2,3)^{15}$	$(3,3)^{15}$	$(0,0)^{16}$	$(1,0)^{16}$	$(2,0)^{16}$	$(3,0)^{16}$	$(0,1)^{16}$	$(1,1)^{16}$
$\frac{17}{80}$	$\frac{17}{40}$	$\frac{17}{20}$	$\frac{9}{10}$	$\frac{33}{80}$	$\frac{17}{40}$	$\frac{1}{4}$	$\frac{3}{20}$	$\frac{5}{8}$	$\frac{13}{20}$
$(2,1)^{16}$	$(3,1)^{16}$	$(0,2)^{16}$	$(1,2)^{16}$	$(2,2)^{16}$	$(3,2)^{16}$	$(0,3)^{16}$	$(1,3)^{16}$	$(2,3)^{16}$	$(3,3)^{16}$
$\frac{2}{5}$	$\frac{1}{4}$	$\frac{19}{20}$	1	$\frac{13}{20}$	$\frac{17}{40}$	$\frac{9}{10}$	$\frac{19}{20}$	$\frac{5}{8}$	$\frac{33}{80}$
$(0,0)^{17}$	$(1,0)^{17}$	$(2,0)^{17}$	$(3,0)^{17}$	$(0,1)^{17}$	$(1,1)^{17}$	$(2,1)^{17}$	$(3,1)^{17}$	$(0,2)^{17}$	$(1,2)^{17}$
$\frac{3}{20}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{80}$	$\frac{1}{4}$	$\frac{1}{10}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{17}{40}$	$\frac{1}{5}$
$(2,2)^{17}$	$(3,2)^{17}$	$(0,3)^{17}$	$(1,3)^{17}$	$(2,3)^{17}$	$(3,3)^{17}$	$(0,0)^{18}$	$(0,1)^{18}$	$(0,2)^{18}$	$(0,3)^{18}$
$\frac{1}{10}$	$\frac{1}{20}$	$\frac{33}{80}$	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{1}{20}$	$\frac{1}{80}$	$\frac{1}{40}$	$\frac{1}{20}$	$\frac{1}{20}$
$(3,0)^{20}$	$(3,1)^{20}$	$(3,2)^{20}$	$(3,3)^{20}$	$(0,0)^{21}$	$(1,0)^{21}$	$(2,0)^{21}$	$(3,0)^{21}$	$(0,1)^{21}$	$(1,1)^{21}$
$\frac{17}{80}$	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{1}{20}$	$\frac{17}{80}$	$\frac{17}{40}$	$\frac{17}{20}$	$\frac{9}{10}$	$\frac{1}{5}$	$\frac{2}{5}$
$(2,1)^{21}$	$(3,1)^{21}$	$(0,2)^{21}$	$(1,2)^{21}$	$(2,2)^{21}$	$(3,2)^{21}$	$(0,3)^{21}$	$(1,3)^{21}$	$(2,3)^{21}$	$(3,3)^{21}$
$\frac{4}{5}$	$\frac{17}{20}$	$\frac{1}{10}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{17}{40}$	$\frac{1}{20}$	$\frac{1}{10}$	$\frac{1}{5}$	$\frac{17}{80}$
$(0,0)^{22}$	$(1,0)^{22}$	$(2,0)^{22}$	$(3,0)^{22}$	$(0,1)^{22}$	$(1,1)^{22}$	$(2,1)^{22}$	$(3,1)^{22}$	$(0,2)^{22}$	$(1,2)^{22}$
$\frac{9}{10}$	$\frac{19}{20}$	$\frac{5}{8}$	$\frac{33}{80}$	$\frac{17}{20}$	$\frac{9}{10}$	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{17}{40}$	$\frac{9}{20}$
$(2,2)^{22}$	$(3,2)^{22}$	$(0,3)^{22}$	$(1,3)^{22}$	$(2,3)^{22}$	$(3,3)^{22}$	$(0,0)^{23}$	$(1,0)^{23}$	$(2,0)^{23}$	$(3,0)^{23}$
$\frac{3}{10}$	$\frac{1}{5}$	$\frac{17}{80}$	$\frac{9}{40}$	$\frac{3}{20}$	$\frac{1}{10}$	$\frac{33}{80}$	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{1}{20}$
$(0,1)^{23}$	$(1,1)^{23}$	$(2,1)^{23}$	$(3,1)^{23}$	$(0,2)^{23}$	$(1,2)^{23}$	$(2,2)^{23}$	$(3,2)^{23}$	$(0,3)^{23}$	$(1,3)^{23}$
$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{1}{20}$	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{10}$	$\frac{1}{20}$
$(2,3)^{23}$	$(3,3)^{23}$	$(0,0)^{24}$	$(0,1)^{24}$	$(0,2)^{24}$	$(0,3)^{24}$	$(3,0)^{26}$	$(0,0)^{27}$	$(1,0)^{27}$	$(2,0)^{27}$
$\frac{1}{40}$	$\frac{1}{80}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{80}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{10}$	$\frac{1}{5}$
$(3,0)^{27}$	$(0,0)^{28}$	$(1,0)^{28}$	$(2,0)^{28}$	$(3,0)^{28}$	$(0,0)^{29}$	$(1,0)^{29}$	$(2,0)^{29}$	$(3,0)^{29}$	$(0,0)^{30}$
$\frac{17}{80}$	$\frac{17}{80}$	$\frac{9}{40}$	$\frac{3}{20}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{80}$	$\frac{1}{80}$

Table 3: The values of the nonzero Bernstein coefficients of the U-spline basis function highlighted in fig. 90.

E.3 U-spline extraction coefficients on mesh equivalent to analysis-suitable T-spline with non-crossing edge extensions

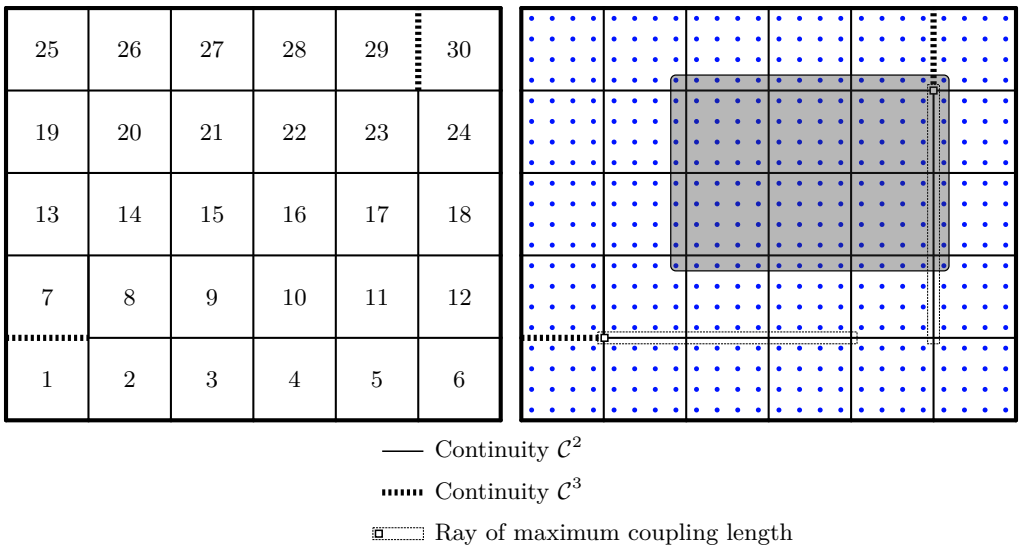
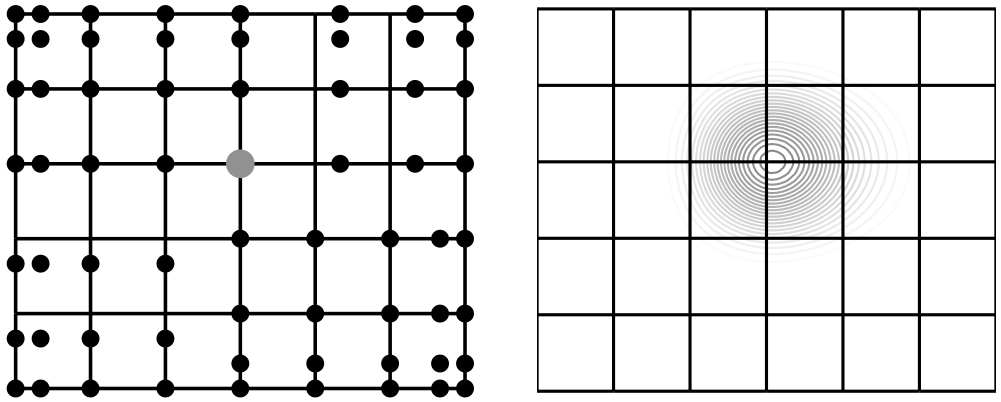


Figure 92: Basis function on a U-spline that is equivalent to an analysis-suitable T-spline [55, 35]. The T-mesh of the equivalent T-spline is shown in fig. 94. Note that the ribbons of maximum coupling length in this example are analogous to the non-crossing T-junction extensions which guarantee an analysis-suitable T-spline. The Bernstein coefficients of the highlighted basis function are listed in table 4.



(a) The control points for a linear parameterization of the mesh in fig. 92. The highlighted control point corresponds to the highlighted basis function.

(b) Contour plot of the basis function whose nonzero Bernstein coefficients are highlighted in fig. 92.

Figure 93: Control points and basis function contours for the U-spline in fig. 92.

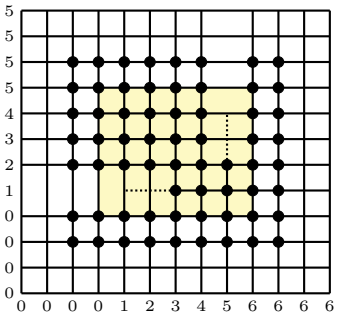


Figure 94: The T-mesh for a cubic T-spline which is equivalent to the U-spline mesh in fig. 92 [55]. The area highlighted in yellow corresponds to the Bézier elements of the U-spline mesh. The dotted lines next to the T-junctions are the lines of reduced continuity, where there is still $\mathcal{C}^2$ continuity [53]. It is notable to observe that the vertices associated with the T-junctions on the T-mesh are not the same as the vertices adjacent to the supersmooth interfaces on the U-spline mesh, due to the lines of reduced continuity.

$(3,3)^{8}$	$(0,3)^{9}$	$(1,3)^{9}$	$(2,3)^{9}$	$(3,3)^{9}$	$(0,3)^{10}$	$(1,3)^{10}$	$(2,3)^{10}$	$(3,3)^{10}$	$(0,3)^{11}$
$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{9}$	$\frac{2}{9}$	$\frac{17}{72}$	$\frac{17}{72}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{9}$	$\frac{1}{9}$
$(1,3)^{11}$	$(2,3)^{11}$	$(3,3)^{11}$	$(0,3)^{12}$	$(3,0)^{14}$	$(3,1)^{14}$	$(3,2)^{14}$	$(3,3)^{14}$	$(0,0)^{15}$	$(1,0)^{15}$
$\frac{1}{18}$	$\frac{1}{36}$	$\frac{1}{72}$	$\frac{1}{72}$	$\frac{1}{18}$	$\frac{1}{9}$	$\frac{2}{9}$	$\frac{2}{9}$	$\frac{1}{18}$	$\frac{1}{9}$
$(2,0)^{15}$	$(3,0)^{15}$	$(0,1)^{15}$	$(1,1)^{15}$	$(2,1)^{15}$	$(3,1)^{15}$	$(0,2)^{15}$	$(1,2)^{15}$	$(2,2)^{15}$	$(3,2)^{15}$
$\frac{2}{9}$	$\frac{17}{72}$	$\frac{1}{9}$	$\frac{2}{9}$	$\frac{4}{9}$	$\frac{17}{36}$	$\frac{2}{9}$	$\frac{4}{9}$	$\frac{8}{9}$	$\frac{17}{18}$
$(0,3)^{15}$	$(1,3)^{15}$	$(2,3)^{15}$	$(3,3)^{15}$	$(0,0)^{16}$	$(1,0)^{16}$	$(2,0)^{16}$	$(3,0)^{16}$	$(0,1)^{16}$	$(1,1)^{16}$
$\frac{2}{9}$	$\frac{4}{9}$	$\frac{8}{9}$	$\frac{17}{18}$	$\frac{17}{72}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{9}$	$\frac{17}{36}$	$\frac{1}{2}$
$(2,1)^{16}$	$(3,1)^{16}$	$(0,2)^{16}$	$(1,2)^{16}$	$(2,2)^{16}$	$(3,2)^{16}$	$(0,3)^{16}$	$(1,3)^{16}$	$(2,3)^{16}$	$(3,3)^{16}$
$\frac{1}{3}$	$\frac{2}{9}$	$\frac{17}{18}$	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{17}{18}$	1	$\frac{2}{3}$	$\frac{4}{9}$
$(0,0)^{17}$	$(1,0)^{17}$	$(2,0)^{17}$	$(3,0)^{17}$	$(0,1)^{17}$	$(1,1)^{17}$	$(2,1)^{17}$	$(3,1)^{17}$	$(0,2)^{17}$	$(1,2)^{17}$
$\frac{1}{9}$	$\frac{1}{18}$	$\frac{1}{36}$	$\frac{1}{72}$	$\frac{2}{9}$	$\frac{1}{9}$	$\frac{1}{18}$	$\frac{1}{36}$	$\frac{4}{9}$	$\frac{2}{9}$
$(2,2)^{17}$	$(3,2)^{17}$	$(0,3)^{17}$	$(1,3)^{17}$	$(2,3)^{17}$	$(3,3)^{17}$	$(0,0)^{18}$	$(0,1)^{18}$	$(0,2)^{18}$	$(0,3)^{18}$
$\frac{1}{9}$	$\frac{1}{18}$	$\frac{4}{9}$	$\frac{2}{9}$	$\frac{1}{9}$	$\frac{1}{18}$	$\frac{1}{72}$	$\frac{1}{36}$	$\frac{1}{18}$	$\frac{1}{18}$
$(3,0)^{20}$	$(3,1)^{20}$	$(3,2)^{20}$	$(3,3)^{20}$	$(0,0)^{21}$	$(1,0)^{21}$	$(2,0)^{21}$	$(3,0)^{21}$	$(0,1)^{21}$	$(1,1)^{21}$
$\frac{2}{9}$	$\frac{2}{9}$	$\frac{1}{9}$	$\frac{1}{18}$	$\frac{2}{9}$	$\frac{4}{9}$	$\frac{8}{9}$	$\frac{17}{18}$	$\frac{2}{9}$	$\frac{4}{9}$
$(2,1)^{21}$	$(3,1)^{21}$	$(0,2)^{21}$	$(1,2)^{21}$	$(2,2)^{21}$	$(3,2)^{21}$	$(0,3)^{21}$	$(1,3)^{21}$	$(2,3)^{21}$	$(3,3)^{21}$
$\frac{8}{9}$	$\frac{17}{18}$	$\frac{1}{9}$	$\frac{2}{9}$	$\frac{4}{9}$	$\frac{17}{36}$	$\frac{1}{18}$	$\frac{1}{9}$	$\frac{2}{9}$	$\frac{17}{72}$
$(0,0)^{22}$	$(1,0)^{22}$	$(2,0)^{22}$	$(3,0)^{22}$	$(0,1)^{22}$	$(1,1)^{22}$	$(2,1)^{22}$	$(3,1)^{22}$	$(0,2)^{22}$	$(1,2)^{22}$
$\frac{17}{18}$	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{17}{18}$	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{17}{36}$	$\frac{1}{2}$
$(2,2)^{22}$	$(3,2)^{22}$	$(0,3)^{22}$	$(1,3)^{22}$	$(2,3)^{22}$	$(3,3)^{22}$	$(0,0)^{23}$	$(1,0)^{23}$	$(2,0)^{23}$	$(3,0)^{23}$
$\frac{1}{3}$	$\frac{2}{9}$	$\frac{17}{72}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{9}$	$\frac{4}{9}$	$\frac{2}{9}$	$\frac{1}{9}$	$\frac{1}{18}$
$(0,1)^{23}$	$(1,1)^{23}$	$(2,1)^{23}$	$(3,1)^{23}$	$(0,2)^{23}$	$(1,2)^{23}$	$(2,2)^{23}$	$(3,2)^{23}$	$(0,3)^{23}$	$(1,3)^{23}$
$\frac{4}{9}$	$\frac{2}{9}$	$\frac{1}{9}$	$\frac{1}{18}$	$\frac{2}{9}$	$\frac{1}{9}$	$\frac{1}{18}$	$\frac{1}{36}$	$\frac{1}{9}$	$\frac{1}{18}$
$(2,3)^{23}$	$(3,3)^{23}$	$(0,0)^{24}$	$(0,1)^{24}$	$(0,2)^{24}$	$(0,3)^{24}$	$(3,0)^{26}$	$(0,0)^{27}$	$(1,0)^{27}$	$(2,0)^{27}$
$\frac{1}{36}$	$\frac{1}{72}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{36}$	$\frac{1}{72}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{9}$	$\frac{2}{9}$
$(3,0)^{27}$	$(0,0)^{28}$	$(1,0)^{28}$	$(2,0)^{28}$	$(3,0)^{28}$	$(0,0)^{29}$	$(1,0)^{29}$	$(2,0)^{29}$	$(3,0)^{29}$	$(0,0)^{30}$
$\frac{17}{72}$	$\frac{17}{72}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{18}$	$\frac{1}{36}$	$\frac{1}{72}$	$\frac{1}{72}$

Table 4: The values of the nonzero Bernstein coefficients of the U-spline basis function highlighted in fig. 92.

E.4 U-spline extraction coefficients near an extraordinary vertex

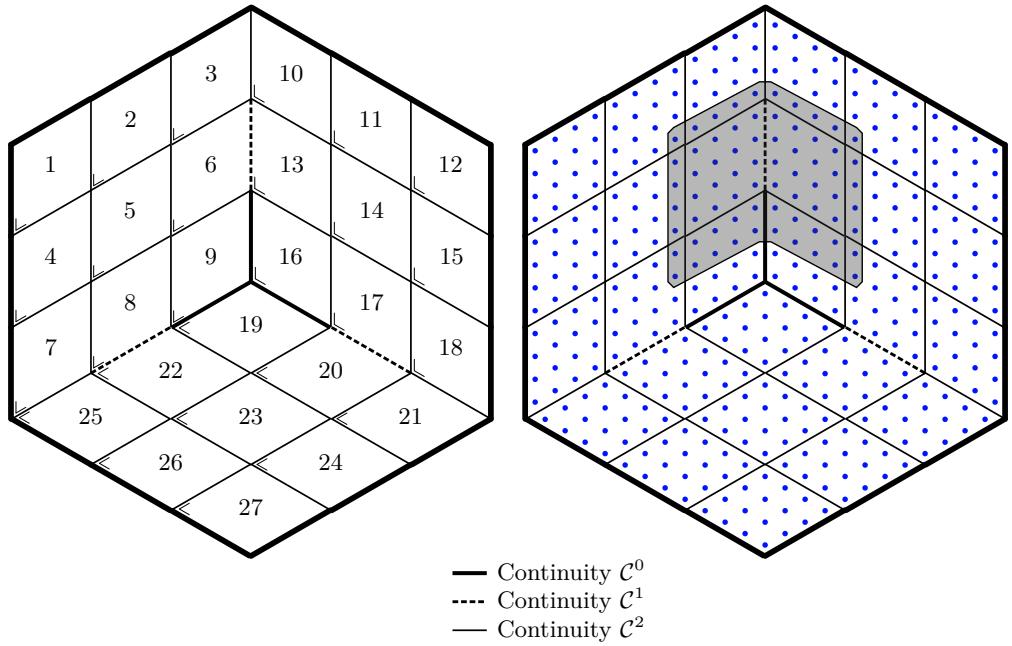
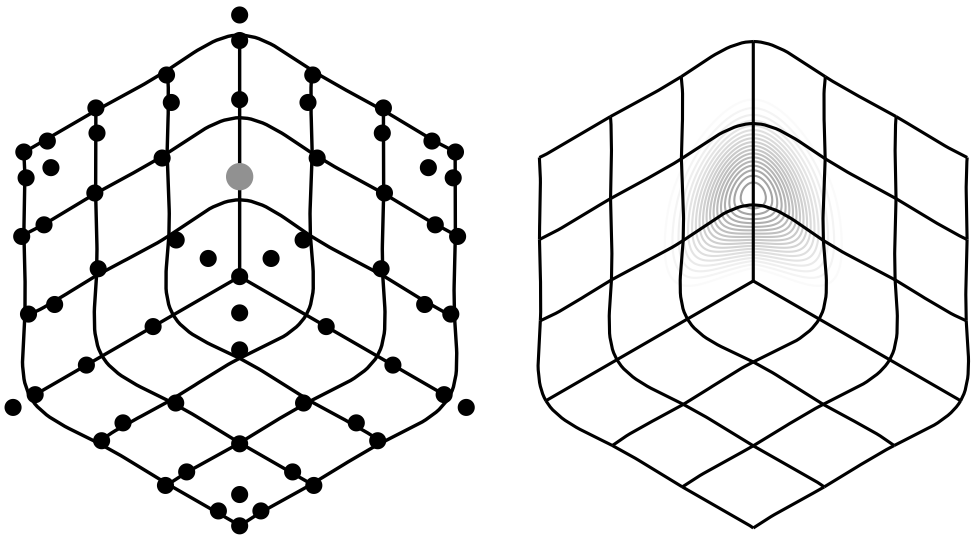


Figure 95: The support of one of the basis functions on a cubic mesh with a valence-3 extraordinary vertex. The edges about the extraordinary vertex are creased in a stair-step pattern, starting at $\mathcal{C}^0$ and incrementally increasing continuity until $\mathcal{C}^2$. The Bernstein coefficients of the highlighted basis function are listed in table 5.



(a) The control points for a linear parameterization of the mesh in fig. 95. The highlighted control point corresponds to the highlighted basis function.

(b) Contour plot of the basis function whose nonzero Bernstein coefficients are highlighted in fig. 95.

Figure 96: Control points and basis function contours for the U-spline in fig. 95.

$(3,0)^2$	$(0,0)^3$	$(1,0)^3$	$(2,0)^3$	$(3,0)^3$	$(3,0)^5$	$(3,1)^5$	$(3,2)^5$	$(3,3)^5$	$(0,0)^6$
$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{7}{32}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{7}{32}$
$(1,0)^6$	$(2,0)^6$	$(3,0)^6$	$(0,1)^6$	$(1,1)^6$	$(2,1)^6$	$(3,1)^6$	$(0,2)^6$	$(1,2)^6$	$(2,2)^6$
$\frac{7}{16}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	1	1	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$
$(3,2)^6$	$(0,3)^6$	$(1,3)^6$	$(2,3)^6$	$(3,3)^6$	$(3,2)^8$	$(3,3)^8$	$(0,2)^9$	$(1,2)^9$	$(2,2)^9$
$\frac{1}{2}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{16}$	$\frac{7}{32}$	$\frac{3}{16}$	$\frac{3}{8}$	$\frac{3}{4}$
$(3,2)^9$	$(0,3)^9$	$(1,3)^9$	$(2,3)^9$	$(3,3)^9$	$(0,0)^{10}$	$(1,0)^{10}$	$(2,0)^{10}$	$(3,0)^{10}$	$(0,0)^{11}$
$\frac{3}{4}$	$\frac{7}{32}$	$\frac{7}{16}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$
$(0,0)^{13}$	$(1,0)^{13}$	$(2,0)^{13}$	$(3,0)^{13}$	$(0,1)^{13}$	$(1,1)^{13}$	$(2,1)^{13}$	$(3,1)^{13}$	$(0,2)^{13}$	$(1,2)^{13}$
$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{16}$	$\frac{7}{32}$	1	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$
$(2,2)^{13}$	$(3,2)^{13}$	$(0,3)^{13}$	$(1,3)^{13}$	$(2,3)^{13}$	$(3,3)^{13}$	$(0,0)^{14}$	$(0,1)^{14}$	$(0,2)^{14}$	$(0,3)^{14}$
$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{7}{32}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
$(0,2)^{16}$	$(1,2)^{16}$	$(2,2)^{16}$	$(3,2)^{16}$	$(0,3)^{16}$	$(1,3)^{16}$	$(2,3)^{16}$	$(3,3)^{16}$	$(0,2)^{17}$	$(0,3)^{17}$
$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{8}$	$\frac{3}{16}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{16}$	$\frac{7}{32}$	$\frac{3}{16}$	$\frac{7}{32}$

Table 5: The values of the nonzero Bernstein coefficients of the U-spline basis function highlighted in fig. 95.

E.5 U-spline extraction coefficients near a triangle

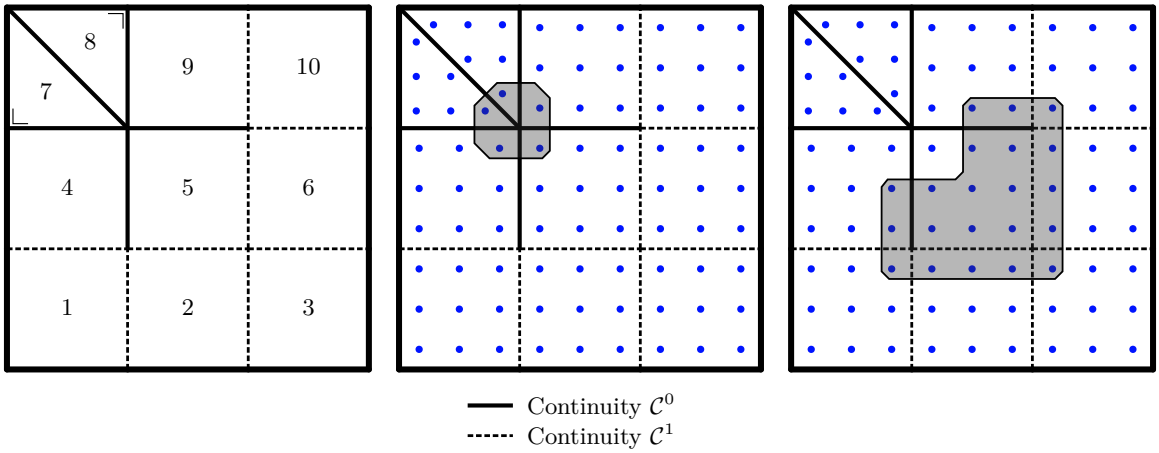
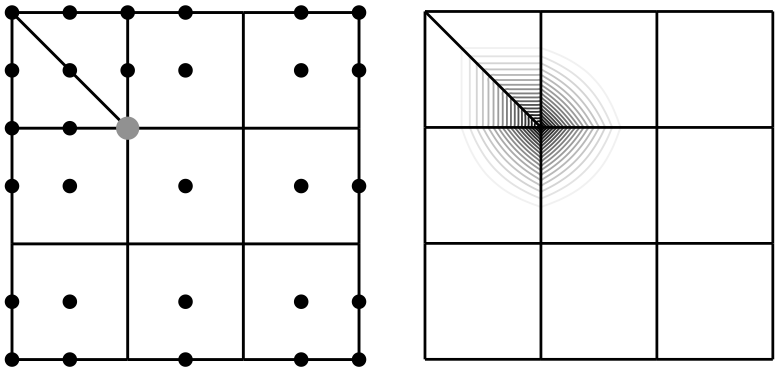


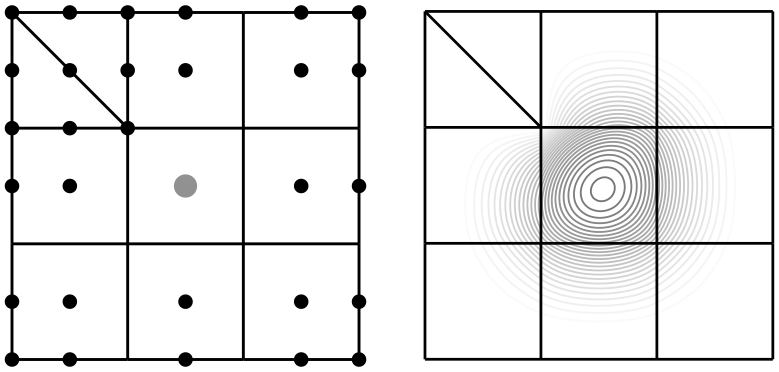
Figure 97: The support of basis functions on a mesh that includes triangles. The values of the control points of the highlighted functions are listed in table 6 and table 7.



(a) The control points for a linear parameterization of the mesh in fig. 97. The highlighted control point corresponds to the left highlighted basis function.

(b) Contour plot of the basis function whose nonzero Bernstein coefficients are highlighted on the left in fig. 97.

Figure 98: Control points and basis function contours for the U-spline basis function highlighted on the left in fig. 97.



(a) The control points for a linear parameterization of the mesh in fig. 97. The highlighted control point corresponds to the right highlighted basis function.

(b) Contour plot of the basis function whose nonzero Bernstein coefficients are highlighted on the right in fig. 97.

Figure 99: Control points and basis function contours for the U-spline basis function highlighted on the right in fig. 97.

$(2,2)^4$	$(0,2)^5$	$(2,0)^7$	$(0,2)^8$	$(0,0)^9$
1	1	1	1	1

Table 6: The values of the nonzero Bernstein coefficients of the U-spline basis function highlighted in fig. 97 on the left.

$(2,2)^1$	$(0,2)^2$	$(1,2)^2$	$(2,2)^2$	$(0,2)^3$	$(2,0)^4$	$(2,1)^4$	$(0,0)^5$	$(1,0)^5$	$(2,0)^5$
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
$(0,1)^5$	$(1,1)^5$	$(2,1)^5$	$(1,2)^5$	$(2,2)^5$	$(0,0)^6$	$(0,1)^6$	$(0,2)^6$	$(1,0)^9$	$(2,0)^9$
$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
$(0,0)^{10}$									
$\frac{1}{4}$									

Table 7: The values of the nonzero Bernstein coefficients of the U-spline basis function highlighted in fig. 97 on the right.

References

[1] M. Aigner et al. "Swept Volume Parameterization for Isogeometric Analysis". In: *Mathematics of Surfaces XIII*. Ed. by Edwin R. Hancock, Ralph R. Martin, and Malcolm A. Sabin. Lecture Notes in Computer Science 5654. Springer Berlin Heidelberg, 2009, pp. 19–44. ISBN: 978-3-642-03595-1 978-3-642-03596-8. URL: http://link.springer.com/chapter/10.1007/978-3-642-03596-8_2 (visited on 08/14/2015).

[2] Peter Alfeld. "Bivariate spline spaces and minimal determining sets". In: *Journal of Computational and Applied Mathematics* 119.1 (July 1, 2000), pp. 13–27. ISSN: 0377-0427. DOI: 10.1016/S0377-0427(00)00369-1. URL: http://www.sciencedirect.com/science/article/pii/S0377042700003691 (visited on 04/11/2018).

[3] Peter Alfeld and Larry L. Schumaker. "Smooth macro-elements based on Clough-Tocher triangle splits". In: *Numerische Mathematik* 90.4 (Feb. 1, 2002), pp. 597–616. ISSN: 0945-3245. DOI: 10.1007/s002110100304. URL: https://doi.org/10.1007/s002110100304 (visited on 10/30/2018).

[4] J. H. Argyris, I. Fried, and D. W. Scharpf. "The TUBA Family of Plate Elements for the Matrix Displacement Method". In: *The Aeronautical Journal* 72.692 (Aug. 1968), pp. 701–709. ISSN: 0001-9240, 2059-6464. DOI: 10.1017/S000192400008489X. (Visited on 10/30/2018).

[5] Gerard Awanou, Ming-jun Lai, and Paul Wenston. *The Multivariate Spline Method for Scattered Data Fitting . . .* 2005.

[6] Y. Bazilevs et al. "Isogeometric analysis using T-splines". In: *Computer Methods in Applied Mechanics and Engineering*. Computational Geometry and Analysis 199.5 (Jan. 1, 2010), pp. 229–263. ISSN: 0045-7825. DOI: 10.1016/j.cma.2009.02.036. URL: http://www.sciencedirect.com/science/article/pii/S0045782509000875 (visited on 02/04/2019).

[7] Michel Berkelaar, Kjell Eikland, and Peter Notebaert. *lp_solve*. Version 5.1.0.0. May 2004. URL: http://lpsolve.sourceforge.net/5.5/.

[8] Michael J. Borden et al. "Isogeometric finite element data structures based on Bézier extraction of NURBS". In: *International Journal for Numerical Methods in Engineering* 87.1 (2011), pp. 15–47. ISSN: 1097-0207. DOI: 10.1002/nme.2968. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2968 (visited on 02/04/2019).

[9] Marcel Campen and Denis Zorin. "Similarity Maps and Field-guided T-splines: A Perfect Couple". In: *ACM Trans. Graph.* 36.4 (July 2017), 91:1–91:16. ISSN: 0730-0301. DOI: 10.1145/3072959.3073647. URL: http://doi.acm.org/10.1145/3072959.3073647 (visited on 05/07/2018).

[10] Fehmi Cirak, Michael Ortiz, and Peter Schröder. "Subdivision surfaces: a new paradigm for thin-shell finite-element analysis". In: *International Journal for Numerical Methods in Engineering* 47.12 (Apr. 30, 2000), pp. 2039–2072. ISSN: 1097-0207. DOI: 10.1002/(SICI)1097-0207(20000430)47:12<2039::AID-NME872>3.0.CO;2-1. (Visited on 10/24/2018).

[11] Elaine Cohen, Tom Lyche, and Richard Riesenfeld. "A B-spline-like basis for the Powell-Sabin 12-split based on simplex splines". In: *Mathematics of Computation* 82.283 (2013), pp. 1667–1707. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/S0025-5718-2013-02664-6. URL: https://www.ams.org/home/page/ (visited on 10/30/2018).

[12] T. Coleman and A. Pothen. "The Null Space Problem I. Complexity". In: *SIAM Journal on Algebraic Discrete Methods* 7.4 (Oct. 1, 1986), pp. 527–537. ISSN: 0196-5212. DOI: 10.1137/0607059. URL: http://epubs.siam.org/doi/abs/10.1137/0607059 (visited on 09/16/2016).

[13] T. Coleman and A. Pothen. "The Null Space Problem II. Algorithms". In: *SIAM Journal on Algebraic Discrete Methods* 8.4 (Oct. 1, 1987), pp. 544–563. ISSN: 0196-5212. DOI: 10.1137/0608045. URL: http://epubs.siam.org/doi/abs/10.1137/0608045 (visited on 09/16/2016).

[14] J. A. Cottrell et al. "Isogeometric Analysis of Structural Vibrations". In: (2006). DOI: 10.1016/J.CMA.2005.09.027.

[15] L. Demkowicz et al. "Toward a universal h-p adaptive finite element strategy, part 1. Constrained approximation and data structure". In: *Computer Methods in Applied Mechanics and Engineering* 77.1 (Dec. 1, 1989), pp. 79–112. ISSN: 0045-7825. DOI: 10.1016/0045-7825(89)90129-1. URL: http://www.sciencedirect.com/science/article/pii/0045782589901291 (visited on 10/31/2018).

[16] Jiansong Deng et al. "Polynomial splines over hierarchical T-meshes". In: *Graphical Models* 70.4 (July 1, 2008), pp. 76–86. ISSN: 1524-0703. DOI: 10.1016/j.gmod.2008.03.001. URL: http://www.sciencedirect.com/science/article/pii/S1524070308000039 (visited on 10/29/2018).

[17] Tor Dokken, Tom Lyche, and Kjell Fredrik Pettersen. "Polynomial splines over locally refined box-partitions". In: *Computer Aided Geometric Design* 30.3 (Mar. 1, 2013), pp. 331–356. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2012.12.005. URL: http://www.sciencedirect.com/science/article/pii/S0167839613000113 (visited on 10/29/2018).

[18] J. M. Escobar et al. "A new approach to solid modeling with trivariate T-splines based on mesh optimization". In: *Computer Methods in Applied Mechanics and Engineering* 200.45 (Oct. 15, 2011), pp. 3210–3222. ISSN: 0045-7825. DOI: 10.1016/j.cma.2011.07.004. URL: http://www.sciencedirect.com/science/article/pii/S0045782511002386 (visited on 08/10/2015).

[19] J. M. Escobar et al. "The meccano method for isogeometric solid modeling and applications". In: *Engineering with Computers* 30.3 (Dec. 1, 2012), pp. 331–343. ISSN: 0177-0667, 1435-5663. DOI: 10.1007/s00366-012-0300-z. URL: http://link.springer.com/article/10.1007/s00366-012-0300-z (visited on 06/30/2016).

[20] Rida T. Farouki. "The Bernstein polynomial basis: A centennial retrospective". In: *Computer Aided Geometric Design* 29.6 (Aug. 2012), pp. 379–419. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2012.03.001. URL: http://www.sciencedirect.com/science/article/pii/S0167839612000192 (visited on 04/13/2015).

[21] Carlo Garoni et al. "Spectral analysis and spectral symbol of matrices in isogeometric Galerkin methods". In: *Mathematics of Computation* 86.305 (2017), pp. 1343–1373. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/mcom/3143. URL: https://www.ams.org/home/page/ (visited on 11/01/2018).

[22] Carlotta Giannelli, Bert Jüttler, and Hendrik Speleers. "THB-splines: The truncated basis for hierarchical splines". In: *Computer Aided Geometric Design*. Geometric Modeling and Processing 2012 29.7 (Oct. 1, 2012), pp. 485–498. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2012.03.025. URL: http://www.sciencedirect.com/science/article/pii/S0167839612000519 (visited on 07/17/2020).

[23] David Groisser and Jörg Peters. "Matched -constructions always yield -continuous isogeometric elements". In: *Computer Aided Geometric Design* 34 (Mar. 2015), pp. 67–72. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2015.02.002. URL: http://www.sciencedirect.com/science/article/pii/S0167839615000151 (visited on 05/20/2015).

[24] Klaus Hollig. *Finite Element Methods with B-splines*. Google-Books-ID: jSVeF8FgYD8C. SIAM, Jan. 1, 2003. 153 pp. ISBN: 978-0-89871-753-2.

[25] X. Hu, D. Han, and M. Lai. "Bivariate Splines of Various Degrees for Numerical Solution of Partial Differential Equations". In: *SIAM Journal on Scientific Computing* 29.3 (Jan. 1, 2007), pp. 1338–1354. ISSN: 1064-8275. DOI: 10.1137/060667207. URL: http://epubs.siam.org/doi/abs/10.1137/060667207 (visited on 08/21/2015).

[26] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". In: *Computer Methods in Applied Mechanics and Engineering* 194.39 (Oct. 1, 2005), pp. 4135–4195. ISSN: 0045-7825. DOI: 10.1016/j.cma.2004.10.008. URL: https://www.sciencedirect.com/science/article/pii/S0045782504005171 (visited on 06/14/2021).

[27] Thomas J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Courier Corporation, May 23, 2012. 706 pp. ISBN: 978-0-486-13502-1.

[28] Noah Jaxon and Xiaoping Qian. "Isogeometric analysis on triangulations". In: *Computer-Aided Design*. 2013 SIAM Conference on Geometric and Physical Modeling 46 (Jan. 2014), pp. 45–57. ISSN: 0010-4485. DOI: 10.1016/j.cad.2013.08.017. URL: http://www.sciencedirect.com/science/article/pii/S0010448513001577 (visited on 08/13/2015).

[29] Bert Jüttler. "The dual basis functions for the Bernstein polynomials". en. In: *Advances in Computational Mathematics* 8.4 (June 1998), pp. 345–352. ISSN: 1572-9044. DOI: 10.1023/A:1018912801267. URL: https://doi.org/10.1023/A:1018912801267 (visited on 02/23/2022).

[30] Mario Kapl, Giancarlo Sangalli, and Thomas Takacs. "Construction of analysis-suitable G1 planar multi-patch parameterizations". In: *Computer-Aided Design* 97 (Apr. 1, 2018), pp. 41–55. ISSN: 0010-4485. DOI: 10.1016/j.cad.2017.12.002. URL: http://www.sciencedirect.com/science/article/pii/S0010448517302439 (visited on 10/30/2018).

[31] Ming-Jun Lai and Larry L. Schumaker. *Spline Functions on Triangulations*. Google-Books-ID: 6hvqGg-bBmEoC. Cambridge University Press, Apr. 19, 2007. 609 pp. ISBN: 978-0-521-87592-9.

[32] X. Li and M. A. Scott. "Analysis-suitable T-splines: characterization, refineability, and approximation". In: *Mathematical Models and Methods in Applied Science* 24.6 (2014), pp. 1141–1164.

[33] Xin Li and Falai Chen. "On the instability in the dimension of splines spaces over T-meshes". In: *Computer Aided Geometric Design* 28.7 (Oct. 1, 2011), pp. 420–426. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2011.08.001. URL: http://www.sciencedirect.com/science/article/pii/S0167839611000896 (visited on 07/11/2017).

[34] Xin Li, Jiansong Deng, and Falai Chen. "Polynomial splines over general T-meshes". In: *The Visual Computer* 26.4 (Apr. 2010), pp. 277–286. ISSN: 0178-2789, 1432-2315. DOI: 10.1007/s00371-009-0410-9. URL: https://link.springer.com/article/10.1007/s00371-009-0410-9 (visited on 12/14/2017).

[35] Xin Li et al. "On linear independence of T-spline blending functions". In: *Computer Aided Geometric Design*. Geometric Constraints and Reasoning 29.1 (Jan. 2012), pp. 63–76. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2011.08.005. URL: http://www.sciencedirect.com/science/article/pii/S0167839611000938 (visited on 01/28/2016).

[36] Lei Liu. "Volumetric T-spline Construction for Isogeometric Analysis – Feature Preservation, Weighted Basis and Arbitrary Degree". en. thesis. Carnegie Mellon University, Sept. 2015. DOI: 10.1184/R1/6724256.v1. URL: https://figshare.com/articles/thesis/Volumetric_T-spline_Construction_for_Isogeometric_Analysis_Feature_Preservation_Weighted_Basis_and_Arbitrary_Degree/6724256/1 (visited on 01/04/2022).

[37] Lei Liu et al. "Volumetric T-spline construction using Boolean operations". In: *Engineering with Computers* 30.4 (Nov. 19, 2013), pp. 425–439. ISSN: 0177-0667, 1435-5663. DOI: 10.1007/s00366-013-0346-6. URL: http://link.springer.com/article/10.1007/s00366-013-0346-6 (visited on 08/14/2015).

[38] Tom Lyche and Georg Muntingh. "Simplex Spline Bases on the Powell-Sabin 12-Split: Part I". In: *Oberwolfach Reports* 12.2 (2015), pp. 1139–1200. ISSN: 1660-8933. DOI: 10.4171/OWR/2015/21. arXiv: 1505.01798. URL: http://arxiv.org/abs/1505.01798 (visited on 10/30/2018).

[39] Tobias Martin, Elaine Cohen, and Mike Kirby. "Volumetric Parameterization and Trivariate B-spline Fitting Using Harmonic Functions". In: *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling*. SPM '08. New York, NY, USA: ACM, 2008, pp. 269–280. ISBN: 978-1-60558-106-4. DOI: 10.1145/1364901.1364938. URL: http://doi.acm.org/10.1145/1364901.1364938 (visited on 08/10/2015).

[40] Marie-Laurence Mazure. "On a general new class of quasi Chebyshevian splines". In: *Numerical Algorithms* 58.3 (Nov. 1, 2011), pp. 399–438. ISSN: 1017-1398, 1572-9265. DOI: 10.1007/s11075-011-9461-x. URL: https://link.springer.com/article/10.1007/s11075-011-9461-x (visited on 05/03/2018).

[41] S. Morganti et al. "Patient-specific isogeometric structural analysis of aortic valve closure". In: *Computer Methods in Applied Mechanics and Engineering*. Isogeometric Analysis Special Issue 284 (Feb. 1, 2015), pp. 508–520. ISSN: 0045-7825. DOI: 10.1016/j.cma.2014.10.010. URL: http://www.sciencedirect.com/science/article/pii/S0045782514003806 (visited on 02/04/2019).

[42] Philipp Morgenstern. “Globally structured 3D Analysis-suitable T-splines: definition, linear independence and m-graded local refinement”. In: *SIAM Journal on Numerical Analysis* 54.4 (Jan. 2016). arXiv: 1505.05392, pp. 2163–2186. ISSN: 0036-1429, 1095-7170. DOI: 10.1137/15M102229X. URL: http://arxiv.org/abs/1505.05392 (visited on 01/04/2022).

[43] Marian Neamtu. “Delaunay configurations and multivariate splines: A generalization of a result of B. N. Delaunay”. In: *Transactions of the American Mathematical Society* 359.7 (2007), pp. 2993–3004. ISSN: 0002-9947, 1088-6850. DOI: 10.1090/S0002-9947-07-03976-1. URL: http://www.ams.org/tran/2007-359-07/S0002-9947-07-03976-1/ (visited on 09/10/2015).

[44] Thien Nguyen and Jörg Peters. “Refinable C1 spline elements for irregular quad layout”. In: *Computer Aided Geometric Design*. Geometric Modeling and Processing 2016 43 (Mar. 2016), pp. 123–130. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2016.02.009. URL: http://www.sciencedirect.com/science/article/pii/S0167839616300103 (visited on 12/22/2016).

[45] Francesca Pelosi et al. “Splines over regular triangulations in numerical simulation”. In: *Computer-Aided Design*. Isogeometric Design and Analysis 82 (Jan. 2017), pp. 100–111. ISSN: 0010-4485. DOI: 10.1016/j.cad.2016.08.002. URL: http://www.sciencedirect.com/science/article/pii/S0010448516300902 (visited on 01/19/2017).

[46] J. Peters and U. Reif. *Subdivision Surfaces*. Geometry and Computing. Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-09527-6. URL: https://books.google.com/books?id=ndaecQAACAAJ.

[47] Jörg Peters. “Splines and unsorted knot sequences”. In: *Computer Aided Geometric Design* 30.7 (Oct. 1, 2013), pp. 733–741. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2013.06.001. URL: http://www.sciencedirect.com/science/article/pii/S016783961300054X (visited on 11/05/2018).

[48] Hartmut Prautzsch, Wolfgang Boehm, and Marco Paluszny. *Bezier and B-Spline Techniques*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2002. ISBN: 978-3-540-43761-1.

[49] Ulrich Reif. “A Refineable Space of Smooth Spline Surfaces of Arbitrary Topological Genus”. In: *Journal of Approximation Theory* 90.2 (Aug. 1997), pp. 174–199. ISSN: 0021-9045. DOI: 10.1006/jath.1996.3079. URL: http://www.sciencedirect.com/science/article/pii/S0021904596930798 (visited on 05/11/2016).

[50] Larry Schumaker. *Spline Functions: Basic Theory*. Cambridge University Press, Aug. 16, 2007. 598 pp. ISBN: 978-0-521-70512-7.

[51] Larry L. Schumaker and Lujun Wang. “Spline spaces on TR-meshes with hanging vertices”. In: *Numerische Mathematik* 118.3 (July 1, 2011), pp. 531–548. ISSN: 0945-3245. DOI: 10.1007/s00211-010-0353-0. URL: https://doi.org/10.1007/s00211-010-0353-0 (visited on 10/22/2018).

[52] M. A. Scott, D. C. Thomas, and E. J. Evans. “Isogeometric spline forests”. In: *Computer Methods in Applied Mechanics and Engineering* 269 (Feb. 1, 2014), pp. 222–264. ISSN: 0045-7825. DOI: 10.1016/j.cma.2013.10.024. URL: http://www.sciencedirect.com/science/article/pii/S0045782513002764 (visited on 04/08/2015).

[53] M. A. Scott et al. “Local refinement of analysis-suitable T-splines”. In: *Computer Methods in Applied Mechanics and Engineering* 213-216 (Mar. 1, 2012), pp. 206–222. ISSN: 0045-7825. DOI: 10.1016/j.cma.2011.11.022. URL: http://www.sciencedirect.com/science/article/pii/S0045782511003689 (visited on 02/04/2019).

[54] Michael A. Scott et al. “Isogeometric finite element data structures based on Bézier extraction of T-splines”. In: *International Journal for Numerical Methods in Engineering* 88.2 (Oct. 14, 2011), pp. 126–156. ISSN: 1097-0207. DOI: 10.1002/nme.3167. URL: http://onlinelibrary.wiley.com/doi/10.1002/nme.3167/abstract.

[55] Thomas Sederberg et al. “T-splines and T-NURCCs”. In: *Faculty Publications* (July 27, 2003). URL: https://scholarsarchive.byu.edu/facpub/1057.

[56] Thomas W. Sederberg, Jianmin Zheng, and Xiaowen Song. “Knot intervals and multi-degree splines”. In: *Computer Aided Geometric Design* 20.7 (Oct. 1, 2003), pp. 455–468. ISSN: 0167-8396. DOI: 10.1016/S0167-8396(03)00096-7. URL: http://www.sciencedirect.com/science/article/pii/S0167839603000967 (visited on 05/16/2018).

[57] Wanqiang Shen and Guozhao Wang. "A basis of multi-degree splines". In: *Computer Aided Geometric Design* 27.1 (Jan. 1, 2010), pp. 23–35. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2009.08.005. URL: http://www.sciencedirect.com/science/article/pii/S0167839609000946 (visited on 05/16/2018).

[58] Hendrik Speleers. "Algorithm 999: Computation of Multi-Degree B-Splines". In: *ACM Transactions on Mathematical Software* 45.4 (Dec. 9, 2019), 43:1–43:15. ISSN: 0098-3500. DOI: 10.1145/3321514. URL: https://doi.org/10.1145/3321514 (visited on 07/09/2021).

[59] Hendrik Speleers. "Construction of Normalized B-Splines for a Family of Smooth Spline Spaces Over Powell–Sabin Triangulations". In: *Constructive Approximation* 37.1 (Jan. 24, 2012), pp. 41–72. ISSN: 0176-4276, 1432-0940. DOI: 10.1007/s00365-011-9151-x. URL: http://link.springer.com/article/10.1007/s00365-011-9151-x (visited on 06/29/2015).

[60] Hendrik Speleers, Carla Manni, and Francesca Pelosi. "From NURBS to NURPS geometries". In: *Computer Methods in Applied Mechanics and Engineering* 255 (Mar. 1, 2013), pp. 238–254. ISSN: 0045-7825. DOI: 10.1016/j.cma.2012.11.012. URL: http://www.sciencedirect.com/science/article/pii/S0045782512003507 (visited on 10/14/2016).

[61] Ivar Haugalokken Stangeby. "Simplex Splines on the Powell-Sabin 12-Split". In: (2018). URL: https://www.duo.uio.no/handle/10852/64070 (visited on 10/30/2018).

[62] Gilbert Strang. "Piecewise polynomials and the finite element method". In: *Bulletin of the American Mathematical Society* 79.6 (1973), pp. 1128–1137. ISSN: 0002-9904, 1936-881X. DOI: 10.1090/S0002-9904-1973-13351-8. URL: https://www.ams.org/home/page/ (visited on 10/24/2018).

[63] Deepesh Toshniwal, Hendrik Speleers, and Thomas J. R. Hughes. "Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: Geometric design and isogeometric analysis considerations". In: *Computer Methods in Applied Mechanics and Engineering*. Advances in Computational Mechanics and Scientific Computation—the Cutting Edge 327 (Dec. 1, 2017), pp. 411–458. ISSN: 0045-7825. DOI: 10.1016/j.cma.2017.06.008. URL: http://www.sciencedirect.com/science/article/pii/S0045782517305303 (visited on 10/31/2018).

[64] Deepesh Toshniwal et al. "Multi-degree B-splines: Algorithmic computation and properties". In: *Computer Aided Geometric Design* 76 (Jan. 1, 2020), p. 101792. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2019.101792. URL: https://www.sciencedirect.com/science/article/pii/S0167839619301013 (visited on 07/09/2021).

[65] Deepesh Toshniwal et al. "Multi-degree smooth polar splines: A framework for geometric modeling and isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering*. Special Issue on Isogeometric Analysis: Progress and Challenges 316 (Apr. 1, 2017), pp. 1005–1061. ISSN: 0045-7825. DOI: 10.1016/j.cma.2016.11.009. URL: http://www.sciencedirect.com/science/article/pii/S004578251631533X (visited on 10/31/2018).

[66] Wenyan Wang et al. "Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology". In: *Computer-Aided Design*. Solid and Physical Modeling 2012 45.2 (Feb. 2013), pp. 351–360. ISSN: 0010-4485. DOI: 10.1016/j.cad.2012.10.018. URL: http://www.sciencedirect.com/science/article/pii/S0010448512002230 (visited on 08/10/2015).

[67] Xilu Wang and Xiaoping Qian. "An optimization approach for constructing trivariate B-spline solids". In: *Computer-Aided Design*. 2013 SIAM Conference on Geometric and Physical Modeling 46 (Supplement C Jan. 1, 2014), pp. 179–191. ISSN: 0010-4485. DOI: 10.1016/j.cad.2013.08.030. URL: http://www.sciencedirect.com/science/article/pii/S001044851300170X.

[68] Songtao Xia and Xiaoping Qian. "Isogeometric analysis with Bézier tetrahedra". In: (2017). DOI: 10.1016/J.CMA.2016.09.045.

[69] Songtao Xia, Xilu Wang, and Xiaoping Qian. "Continuity and convergence in rational triangular Bézier spline based isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 297 (Dec. 1, 2015), pp. 292–324. ISSN: 0045-7825. DOI: 10.1016/j.cma.2015.09.001. URL: http://www.sciencedirect.com/science/article/pii/S0045782515002777 (visited on 01/27/2016).

[70] Gang Xu et al. “Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications”. In: *Computer-Aided Design*. Solid and Physical Modeling 2012 45.2 (Feb. 2013), pp. 395–404. ISSN: 0010-4485. DOI: 10.1016/j.cad.2012.10.022. URL: http://www.sciencedirect.com/science/article/pii/S0010448512002278 (visited on 06/30/2016).

[71] Gang Xu et al. “Parameterization of computational domain in isogeometric analysis: Methods and comparison”. In: *Computer Methods in Applied Mechanics and Engineering* 200.23 (June 1, 2011), pp. 2021–2031. ISSN: 0045-7825. DOI: 10.1016/j.cma.2011.03.005. URL: http://www.sciencedirect.com/science/article/pii/S0045782511001101.

[72] Yongjie Zhang, Wenyan Wang, and Thomas J. R. Hughes. “Solid T-spline construction from boundary representations for genus-zero geometry”. In: *Computer Methods in Applied Mechanics and Engineering*. Higher Order Finite Element and Isogeometric Methods 249 (Supplement C Dec. 1, 2012), pp. 185–197. ISSN: 0045-7825. DOI: 10.1016/j.cma.2012.01.014. URL: http://www.sciencedirect.com/science/article/pii/S0045782512000254.