



Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 316 (2017) 1005–1061

**Computer methods
in applied
mechanics and
engineering**

www.elsevier.com/locate/cma

Multi-degree smooth polar splines: A framework for geometric modeling and isogeometric analysis

Deepesh Toshniwal^{a,*}, Hendrik Speleers^b, René R. Hiemstra^a, Thomas J.R. Hughes^a

^a Institute for Computational Engineering and Sciences, University of Texas at Austin, USA

^b Department of Mathematics, University of Rome “Tor Vergata”, Italy

Available online 15 November 2016

Highlights

- A polar spline framework is proposed for geometric modeling and isogeometric analysis.
- We present a practical construction for univariate splines of non-uniform degree.
- This is extended to polar configurations using tensor-products with a collapsed edge.
- A C^k smooth polar spline basis is developed that forms a convex partition of unity.
- Optimal error convergence rates are observed for L^2 , H^1 and H^2 projection problems.

Abstract

We develop a multi-degree polar spline framework with applications to both geometric modeling and isogeometric analysis. First, multi-degree splines are introduced as piecewise non-uniform rational B-splines (NURBS) of non-uniform or variable polynomial degree, and a simple algorithm for their construction is presented. Then, an extension to two-dimensional polar configurations is provided by means of a tensor-product construction with a collapsed edge. Suitable combinations of these basis functions, encoded in a so-called isogeometric analysis suitable extraction operator, yield C^k smooth polar splines for any $k \geq 0$. We show that it is always possible to construct a set of smooth polar spline basis functions that form a convex partition of unity and possess locality. Explicit constructions for $k \in \{0, 1, 2\}$ are presented. Optimal approximation behavior is observed numerically, and examples of applications to free-form design, smooth hole-filling, and high-order partial differential equations demonstrate the applicability of the developed framework.

© 2016 Elsevier B.V. All rights reserved.

Keywords: Multi-degree splines; Polar splines; Free-form design; Smooth hole-filling; Isogeometric analysis

1. Introduction

Many geometric primitives such as cylinders and spheres are most conveniently parametrized in a polar or spherical coordinate system. With regard to geometric modeling and engineering analysis, this gives rise to so-called polar

* Corresponding author.

E-mail addresses: deepesh@ices.utexas.edu (D. Toshniwal), speleers@mat.uniroma2.it (H. Speleers), rene@ices.utexas.edu (R.R. Hiemstra), hughes@ices.utexas.edu (T.J.R. Hughes).

configurations where in 2D, for instance, a disk-like domain is obtained by degenerating an edge of a rectangular patch. Relevant applications of these configurations include geometric design [1–5], computational biomechanics [6–8], and solutions of high-order partial differential equations on surfaces [9], to name a few.

Current polar patch configurations have two main drawbacks that may lead to significant problems during the modeling and analysis stage. The first difficulty is the discontinuity at the origin of the polar construction, a consequence of the degeneracy introduced there. The second difficulty stems from the available boundary parameterizations of conics, as traditional low-degree (≤ 3) computer-aided design (CAD) descriptions are given in terms of globally C^0 non-uniform rational B-splines (NURBS). In analysis, the lack of smoothness makes simulation of phenomena governed by high-order partial differential equations cumbersome [10]. In particular, the spline spaces may not satisfy the regularity assumptions required by the numerical methods being used [11,12]. In geometric modeling, this means that small modifications to an initially smooth geometry may introduce undesirable kinks. This is demonstrated in Figs. 1(a) and 1(b) for the classical 6-point C^0 representation of a circle obtained by splicing together two cubic rational Bézier segments. It is a well-known fact that smoothness can be maintained by ensuring that the top (or bottom) row of control points in the figure stays collinear—a *smoothness constraint* (also called *continuity constraint*). While there are several possibilities of imposing this condition, we are strongly biased towards one: representation of the geometry using a minimal set of basis functions that are a priori smooth and have nice properties such as partition of unity, non-negativity, locality, etc. This is the approach we advocate in the present text, and is illustrated in Figs. 1(c) and 1(d) with a 4-point C^1 representation of a circle.

In this paper, we aim to develop polar spline technology that alleviates the aforementioned drawbacks by using solutions where the user only needs to work with a suitable minimum number of degrees of freedom. Therefore, in the spirit of [13], we first devise a novel approach towards building univariate splines as piecewise NURBS segments that are connected in a C^k fashion. The construction is simple and allows for non-uniformity in polynomial degree as well. Then, an extension to two-dimensional polar configurations is provided by means of a tensor-product construction with a collapsed edge. The resulting polar configuration is C^k everywhere except at the origin where it is discontinuous. Suitable combinations of these basis functions yield a set of globally C^k smooth basis functions for polar patches.

Fig. 2 shows a design-based application of our polar spline framework. In Fig. 2(a), a geometric object composed of 16 NURBS patches of different degrees is shown. There is a hole at the center of this object, with its boundary curve shown in Fig. 2(b). Using the multi-degree spline framework developed in Section 2, we can represent such curves with only a handful of degrees of freedom; the yellow disks in Fig. 2(b) are the control points used to reproduce the hole boundary. Subsequently, the polar spline framework laid out in Section 3 allows us to build smooth polar spline surfaces, such as in Fig. 2(c), that can be used to fill the hole in Fig. 2(a). In Fig. 2(d) we see how an arbitrary modification to the spline control net does not introduce non-smoothness in the associated surface. Section 4 presents additional examples of applications in design and analysis.

1.1. Geometry and analysis suitable extraction

The motif of extraction operators runs through this paper. The following brief exposition introduces the concept of extraction and elaborates on what its compatibility with geometric modeling and engineering analysis entails.

Bézier extraction was proposed in [14] to aid the formation and assembly process in isogeometric analysis (IGA). In particular, it allowed the use of standard finite element methodology within the isogeometric analysis paradigm. In essence, the process of Bézier extraction provides a map from the space of C^{-1} splines to that of C^k splines of the same polynomial degree. In this section we generalize the original idea of Bézier extraction to a linear transformation of any set of basis functions. The construction of these linear transformations is usually motivated by considerations of smoothness—the extraction operator maps onto a smoother subspace of the original. Now, assume that we are given a set of locally supported basis functions $\{B_j : j = 1, \dots, m\}$ forming a convex partition of unity. Then, we can construct another set of functions N_i defined using an extraction operator E , so that

$$N_i(u, v) = \sum_{j=1}^m E_{ij} B_j(u, v). \quad (1.1)$$

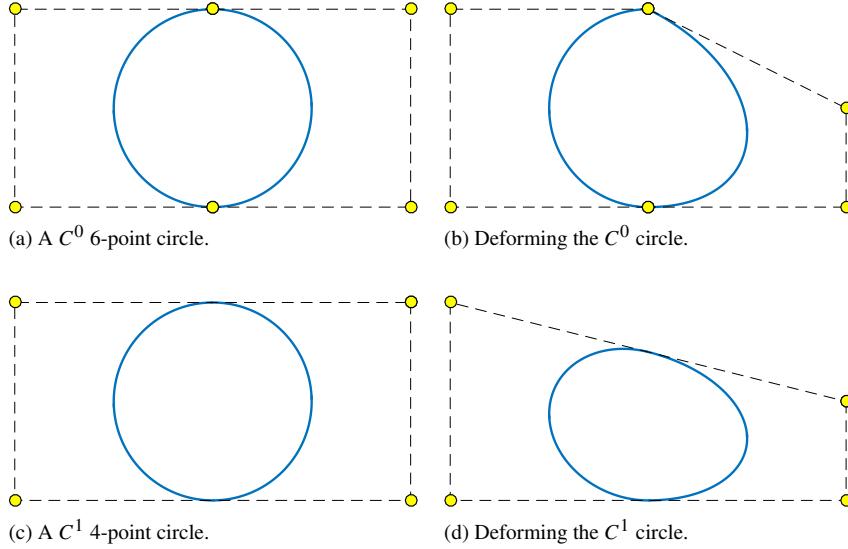


Fig. 1. The top-left figure shows the classical 6-point C^0 representation of a circle obtained by splicing together two cubic rational Bézier segments. The top-right figure illustrates the difficulty in working with such a representation: displacing the control points introduces a kink in the previously smooth geometry. The bottom-left figure shows a 4-point C^1 representation of a circle obtained using piecewise-NURBS basis functions. The bottom-right figure illustrates the advantage in working with such a representation.

Regardless of the smoothness endowed upon functions N_i via \mathbf{E} , we would like them to possess some additional properties – linear independence, non-negativity, locality and partition of unity – in order for them to be compatible with both design and analysis. This leads us to the concept of IGA-suitable extraction operators.¹

Definition 1.1 (IGA-Suitable Extraction). An extraction operator \mathbf{E} , acting on basis functions that form a convex partition of unity, is called an IGA-suitable extraction operator if

- (i) \mathbf{E} is a full-rank matrix,
- (ii) each column of \mathbf{E} sums to 1,
- (iii) each entry in \mathbf{E} is non-negative, and
- (iv) \mathbf{E} imparts locality to the functions N_i (through sparsity).

It is easy to see that the action of an IGA-suitable extraction operator on a convex partition of unity local basis gives rise to another local basis that also forms a convex partition of unity. Indeed, by summing over i in Eq. (1.1), we have

$$\sum_{i=1}^n N_i(u, v) = \sum_{i=1}^n \sum_{j=1}^m E_{ij} B_j(u, v) = \sum_{j=1}^m B_j(u, v) \sum_{i=1}^n E_{ij} = 1, \quad (1.2)$$

as the B_j form a partition of unity. Since \mathbf{E} has non-negative entries and is a full-rank matrix, non-negativity and linear independence of N_i follow from the non-negativity and linear independence of B_j .

Bézier extraction satisfies the above conditions and is, therefore, IGA-suitable. While it can be directly constructed from repeated application of the knot insertion algorithm [15], in the following we apply the concept of extraction in cases for which there is no known knot insertion algorithm. Section 2 illustrates this where we build a univariate space of smooth piecewise NURBS of non-uniform polynomial degree, while in Section 3 we work in a multidimensional setting pertaining to the construction of smooth polar basis functions.

¹ Usage of the spline basis in which the geometries are expressed for the purpose of analysis, and vice versa, is a way of creating efficient design-and-analysis cycles—the raison d'être of IGA. Hence, our terminology “IGA-suitable”.

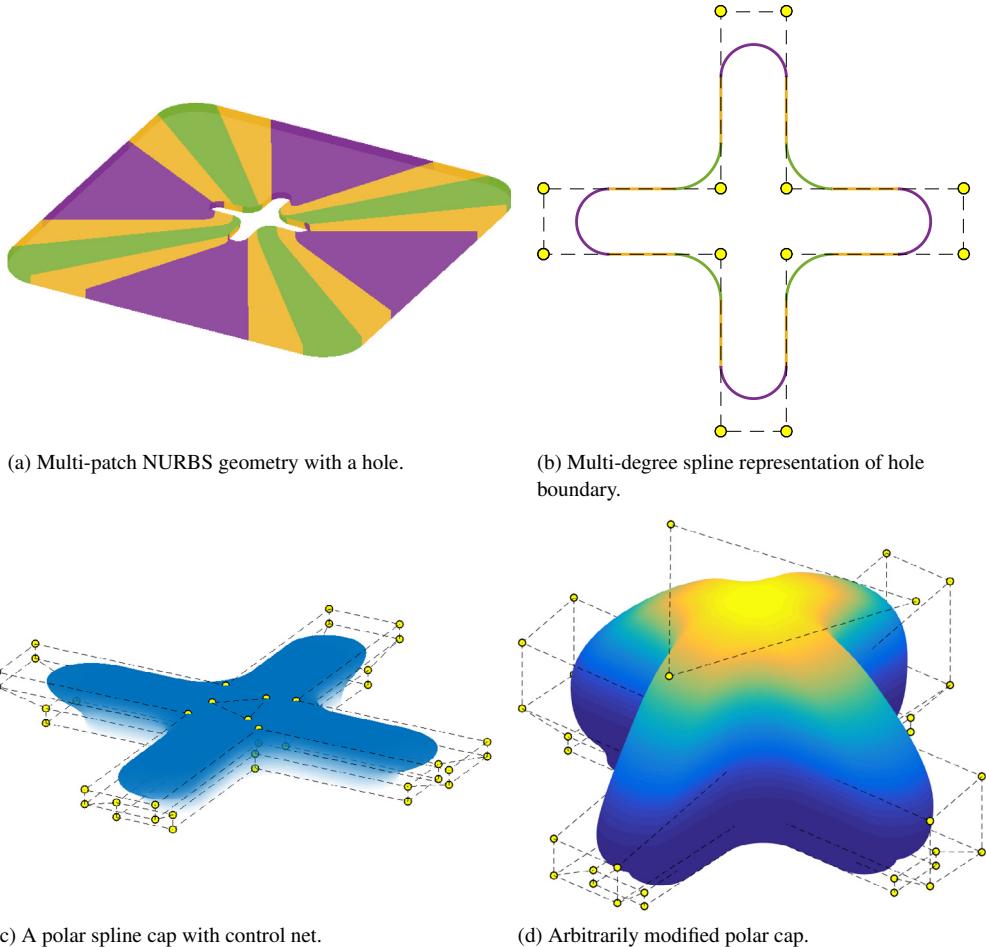


Fig. 2. The top-left figure shows a multi-patch NURBS geometry where the patches have been colored according to their degree: purple ($=3$), green ($=2$) and yellow ($=1$). The top-right figure shows the boundary of the hole represented using the multi-degree spline framework developed in Section 2. The hole at the center of the NURBS geometry can be filled with a multi-degree polar spline cap as developed in Section 3. An example of such a spline cap is shown in the bottom-left figure. All control points of the spline cap are associated with smooth multi-degree basis functions, and arbitrarily displacing them does not lead to any kinks in the modified geometry as shown in the bottom-right figure. The colors in the latter figure represent displacements applied to material points in the bottom-left figure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

1.2. Related literature

Polynomial and rational splines are a well-established tool in many application areas. They usually consist of a uniform degree over the whole domain, which allows them to be efficiently represented and manipulated in terms of the classical B-spline or NURBS basis functions [15–17]. However, having the flexibility of choosing a different degree over different intervals would substantially simplify the description of very complicated functions or curves, as it would require fewer degrees of freedom and lower degrees. *Multi-degree splines* were introduced in [18] for degrees $(1, 2, 3)$ and $(1, n)$. In a more general framework, basis functions for multi-degree splines were constructed in [19–21] by means of recursive integral relations involving global quantities that [22] observed as being difficult to compute. Instead, [22] presented a recursive geometric algorithm for computing multi-degree spline curves. In all of these papers, multi-degree splines are viewed as piecewise-polynomials. Our approach for constructing multi-degree splines is based on imposing continuity constraints between different NURBS segments. The null-space of these constraints yields the coefficients of an extraction operator that help us to create multi-degree basis functions. Imposition of geometric continuity between spline pieces has already been considered in the beta-spline framework; see, e.g., [23] for cubic beta-splines and [24,25] for higher order variants. While our continuity-constraint matrices are similar in

concept to the beta-constraint matrices, our approach towards computation of the spline basis is radically different. Unlike previous approaches requiring symbolic computations [25], we present simple and explicit algorithms for computation of the spline basis, thereby creating a framework rooted in practicality. Moreover, we work in a non-uniform degree setting that has hitherto, to the best of our knowledge, been missing from the beta-spline literature. Recently, there has been considerable interest in the smooth joining of B-spline and NURBS patches in multiple dimensions; see, e.g., [26,27]. This problem and our approach in one dimension are conceptually related, but the difficulties encountered in the multi-dimensional cases are not present in ours.

Smooth polar splines have featured prominently in research oriented towards CAD, mainly in the context of subdivision schemes. General quad-based subdivision techniques that are able to generate smooth surfaces over singularly parametrized points date back to at least the concepts of free-form splines [28] and TURBS [29]. Prescribing a polynomial or a k -jet to enforce smoothness is a well-known approach and was used in [30] to obtain globally C^2 Catmull–Clark subdivisions surfaces. For polar configurations, there also exist specific subdivision schemes producing C^1 surfaces [2,31] and C^2 surfaces [32,33]. The corresponding limit surfaces consist of an infinite sequence of surface rings. A more CAD-friendly finite construction was developed in [3]; this approach is the closest to ours and constructs “shape” basis functions for C^2 polar splines with bi-degree (6, 3). These basis functions correspond to unique Fourier frequencies in the polar expansion of a quadratic surface. In the special case of uniform bi-degree (6, 3) and C^2 smoothness, the polar spline basis functions we construct span the same space as the “shape” basis. However, the “shape” basis does not enjoy non-negativity and does not form a partition of unity, while in the present work we build B-spline-like basis functions that possess these properties by construction. Moreover, extension of the “shape” basis to higher smoothness leads to degrees of freedom that control non-intuitive shape parameters. Curvature continuous polar NURBS surfaces were discussed in [5], and [4] presented a construction of polar caps using periodic B-spline surfaces with G^n continuity for arbitrary n . A completely different approach to produce highly smooth surfaces is based on manifolds [34,35]. In our work, we only focus on (polar) holes with smooth boundaries; for dealing with polygonal holes, we refer to [36–40,26,27] and references therein. On the analysis side, a standard circular serendipity-type element for IGA was proposed in [13], and C^k smooth basis functions over singular parameterizations of triangular domains were constructed in [41]. A comparative study between several spline techniques for solving the Poisson equation on a disk can be found in [42].

1.3. Our contributions

In the multi-degree spline framework that we develop in Section 2:

- We approach multi-degree splines as piecewise-NURBS, a more general viewpoint than the piecewise-polynomial approach in [18–22].
- A simple algorithm presented in Section 2.3 computes an extraction operator that represents multi-degree basis functions as linear combinations of NURBS basis functions of non-uniform degrees. This point of view enables the use of existing efficient algorithms for NURBS evaluations and refinements in the context of multi-degree splines.
- We prove that Bézier extraction is a special case of our multi-degree spline extraction. Additionally, we prove several theoretical properties of the multi-degree basis, in particular, partition of unity and stability. We also investigate local h -, p -, k -refinements. For more details we refer to Sections 2.3 and 2.4.
- Theoretical results and examples confirm that the above approach leads to spline representations with increased flexibility thanks to greater control over the selection of weights for different spline segments, and allows a representation of geometries using fewer degrees of freedom and lower degree polynomials. In particular, low-degree parameterizations of smooth conics are shown in Figs. 1 and 17.

In the polar spline framework developed in Section 3:

- Over a singular (polar) parametrization of a disk-like domain, we present a general construction of C^k multi-degree polar spline basis functions that form a convex partition of unity.
- Explicit construction methodologies for C^0 , C^1 and C^2 polar splines are outlined in Section 3.3.
- Basis functions forming a convex partition of unity allow for surface manipulation through a control-net paradigm as demonstrated by examples of design-based applications; see, e.g., Figs. 2 and 22. Moreover, C^1 polar splines can be enriched by means of *weight points*, a highly intuitive design tool; see Fig. 19.

- The polar spline spaces are used to solve function approximation, Poisson and biharmonic problems, and optimal convergence behavior is observed. The fourth-order non-linear Cahn–Hilliard equation is solved on a disk as an example application; see Section 4.2.

Remark 1.1. This paper deals with both geometric modeling and engineering analysis considerations, and as such it spans the divide existing between these disciplines and communities. We believe this is the essence of isogeometric analysis. However, we have endeavored to make the paper readable and understandable to both communities, and to do that we have at times explicitly described certain things that are well known to one of the communities, but perhaps not the other. We beg the patience of readers who encounter certain descriptions that they feel have no need of mentioning.

2. Smooth boundary descriptions by multi-degree splines

Motivated by considerations detailed in the introduction, specifically with reference to Figs. 1(c) and 2(b), we are interested in creating a representation for smooth *piecewise*-NURBS curves of non-uniform degree. In addition to simplifying both design and analysis, this would allow low degree parametrically smooth representations of conics. We start by defining notation for B-splines and NURBS basis functions in Section 2.1. Section 2.2 introduces multi-degree splines as being composed of NURBS segments that join with specified smoothness. After illustrating their properties and generality, an algorithm is presented for constructing a set of spanning basis functions in Section 2.3. These are built as linear combinations of NURBS basis functions in the different segments. In other words, the algorithm computes an *extraction operator* for the multi-degree basis functions. We show that the Bézier extraction operator is a special case of our algorithm. In Section 2.4 we outline a practical refinement strategy for multi-degree curves, and we prove stability of the multi-degree spline basis. Section 2.5 concludes with tensor-product multi-degree spline spaces.

2.1. Background: B-splines and NURBS

In this section we define notation for B-spline/NURBS basis functions, and introduce some standard relations that can be found, e.g., in [15].

Given an interval $I := [x_1, x_2] \subset \mathbb{R}$, let us denote with U an *open knot vector* of degree $p \in \mathbb{N}$ and length $(n + p + 1) \in \mathbb{N}$. Assume that we are interested in B-splines of degree p defined on I that are at least C^k everywhere for some $0 \leq k \leq p - 1$. Then, U has the form shown below,

$$\begin{aligned} U &:= [u_1, u_2, \dots, u_{n+p+1}], \quad u_{i+1} \geq u_i, \\ u_1 = u_2 = \dots = u_{p+1} &= x_1 < u_{p+2}, \\ u_{n+1} = u_{n+2} = \dots = u_{n+p+1} &= x_2 > u_n, \end{aligned} \tag{2.1}$$

where the internal knots u_i , $i = p + 2, \dots, n$, have maximum multiplicities of $p - k$. The corresponding set of B-splines $\{b_{j,p} : j = 1, \dots, n\}$ are defined using the recursive relation,

$$b_{j,p}(x) := \frac{x - u_j}{u_{j+p} - u_j} b_{j,p-1}(x) + \frac{u_{j+p+1} - x}{u_{j+p+1} - u_{j+1}} b_{j+1,p-1}(x),$$

starting from

$$b_{j,0}(x) := \begin{cases} 1, & \text{if } u_j \leq x < u_{j+1}, \\ 0, & \text{otherwise,} \end{cases}$$

and under the usual convention that fractions with zero denominator have value zero. Each B-spline $b_{j,p}$ is non-negative on I and is locally supported on $[u_j, u_{j+p+1}]$. Moreover, the functions $b_{j,p}$ are linearly independent and form a partition of unity.

The B-spline space associated to U is denoted with $\mathcal{B}[U]$ and is defined as the span of $\{b_{j,p} : j = 1, \dots, n\}$. This is a space of piecewise polynomials of degree p with smoothness at least C^k (to be precise, with smoothness C^k at a knot with multiplicity $p - k$) and its dimension is n . We define the B-splines to be left continuous at the end point x_2 , so as to avoid asymmetry of the space over I .

Remark 2.1. The structure of U is such that p, k, n and I are embedded in it. Therefore, we will assume that once a knot vector is known, so are the degree, smoothness, and dimension of the corresponding spline space $\mathcal{B}[U]$.

We identify a B-spline function $f \in \mathcal{B}[U]$ with the vector of its coefficients $[f_1, \dots, f_n]$, i.e.,

$$f(x) = \sum_{j=1}^n f_j b_{j,p}(x).$$

The m th derivative of f (for $0 \leq m \leq p$) is given by,

$$\frac{d^m f}{dx^m}(x) = \sum_{j=1}^{n-m} f_{j,m} b_{j,p-m}(x),$$

where the lower degree B-splines $b_{j,p-m}$ are defined over the smaller knot vector $[u_{m+1}, u_{m+2}, \dots, u_{n+p-m+1}]$ and,

$$f_{j,m} := \begin{cases} f_j, & m = 0, \\ \alpha_{j,m} (f_{j+1,m-1} - f_{j,m-1}), & m > 0, \end{cases}$$

and $\alpha_{j,m} := \frac{p-m+1}{u_{j+p+1}-u_{j+m}}$. Only the first (last) $m+1$ basis functions contribute towards the m th derivative at the left (right) end point of I . Thus, given $[f_1, \dots, f_n]$, we can write the m th derivative of f at the left or right end point of I in terms of $[f_1, \dots, f_{m+1}]$ or $[f_{n-m}, \dots, f_n]$, respectively.

B-splines have minimal supports and this can be used to characterize them.

Lemma 2.1. Let $\{B_{j,p} : j = 1, \dots, n\}$ be a set of n non-zero spline functions belonging to $\mathcal{B}[U]$. If these splines,

- (i) have minimal supports, i.e., $B_{j,p}$ is supported on $[u_j, u_{j+p+1}]$, and,
- (ii) form a partition of unity,

then they are B-splines, i.e., $B_{j,p} = b_{j,p}$.

Given a positive function $w \in \mathcal{B}[U]$, we define the corresponding NURBS space as,

$$\mathcal{N}[U, w] := \left\{ \frac{f}{w} : f \in \mathcal{B}[U] \right\}.$$

Note that $\mathcal{B}[U] = \mathcal{N}[U, w]$ when $w \equiv c \in \mathbb{R}^+$. The function w is called a *weighting spline*. Let $w(x) = \sum_{j=1}^n w_j b_{j,p}(x)$, where we assume that $w_j > 0$ (sufficient condition for positivity of w). Then, any NURBS function $g \in \mathcal{N}[U, w]$ can be represented as,

$$g(x) = \sum_{j=1}^n g_j \hat{b}_{j,p}(x), \quad \hat{b}_{j,p}(x) := \frac{w_j b_{j,p}(x)}{\sum_{i=1}^n w_i b_{i,p}(x)},$$

where $\hat{b}_{j,p}$ are the NURBS basis functions that span $\mathcal{N}[U, w]$. The m th derivative of g is given by,

$$\frac{d^m g}{dx^m}(x) = \frac{1}{w(x)} \left(\frac{d^m \tilde{g}}{dx^m}(x) - \sum_{i=1}^m \binom{m}{i} \frac{d^i w}{dx^i}(x) \frac{d^{m-i} g}{dx^{m-i}}(x) \right),$$

where $\tilde{g}(x) := \sum_{j=1}^n g_j w_j b_{j,p}(x) \in \mathcal{B}[U]$. A NURBS curve in \mathbb{R}^d can be constructed as,

$$\mathbf{g}(x) = \sum_{j=1}^n \mathbf{g}_j \hat{b}_{j,p}(x),$$

where $\mathbf{g}_j \in \mathbb{R}^d$ are the control points assigned to each basis function. All coordinate functions of this curve belong to $\mathcal{N}[U, w]$ and therefore all the above relations hold for them.

We now present two elementary results that can be easily deduced from the properties of derivatives of B-splines/NURBS, and will be useful later on. Let $U_0 := U$ be a knot vector and, for $r \geq 1$, let U_r be the knot vector obtained by uniform knot refinement of U_{r-1} , i.e., the knot vector obtained after bisection of each non-empty knot interval. In the following, we define $r_0 := \lceil \log_2 p \rceil$. This is the initial number of refinements that must be applied to an arbitrary non-uniform knot vector U_0 in order to ensure uniformity in (at least) the first and last p knot spans. Of course, if U_0 was uniform to begin with, the following would hold for $r_0 = 0$.

Lemma 2.2. *Let $\mathcal{B}[U_r]$ be the B-spline space spanned by the B-spline basis $\{b_{j,p}^r : j = 1, \dots, n_r\}$ defined over the refined knot vector U_r starting from U_0 . For $m \in \{0, 1, \dots, p\}$ and $j \in \{1, 2, \dots, m+1\}$,*

$$\begin{aligned}\frac{d^m b_{j,p}^{r+r_0}}{dx^m}(x_1) &= 2^{rm} \frac{d^m b_{j,p}^{r_0}}{dx^m}(x_1), \\ \frac{d^m b_{n_r-j+1,p}^{r+r_0}}{dx^m}(x_2) &= 2^{rm} \frac{d^m b_{n_r-j+1,p}^{r_0}}{dx^m}(x_2).\end{aligned}$$

Lemma 2.3. *Let $w \in \mathcal{B}[U_0]$ be a weighting spline, and let $\mathcal{N}[U_r, w]$ be the NURBS space spanned by the NURBS basis $\{\hat{b}_{j,p}^r : j = 1, \dots, n_r\}$ defined over the refined knot vector U_r starting from U_0 . Then, for $m \in \{0, 1, \dots, p\}$ and $j \in \{1, 2, \dots, m+1\}$,*

$$\begin{aligned}\lim_{r \rightarrow \infty} \frac{1}{2^{rm}} \frac{d^m \hat{b}_{j,p}^{r+r_0}}{dx^m}(x_1) &= \frac{d^m b_{j,p}^{r_0}}{dx^m}(x_1), \\ \lim_{r \rightarrow \infty} \frac{1}{2^{rm}} \frac{d^m \hat{b}_{n_r-j+1,p}^{r+r_0}}{dx^m}(x_2) &= \frac{d^m b_{n_r-j+1,p}^{r_0}}{dx^m}(x_2).\end{aligned}$$

In order to illustrate the advantages of the multi-degree framework that we want to set up, we will compare multi-degree spline spaces with NURBS spaces such that both belong to the following configuration.

Configuration 2.1. *Spline functions defined over $I = [0, 3]$ such that,*

- they consist of 3 rational pieces, with the i th piece spanning $[i-1, i]$, and,
- they are C^2 over I .

Example 2.1 (NURBS Function). We want to construct a cubic spline function belonging to Configuration 2.1. Then, according to the classical approach, we can work with the following knot vector,

$$U = [0, 0, 0, 0, 1, 2, 3, 3, 3].$$

Using a weighting spline $w \in \mathcal{B}[U]$ with the following coefficients, for instance,

$$[w_i] = \left[1, \frac{2}{3}, 1, \frac{3}{4}, \frac{1}{3}, 1 \right],$$

we can build $\mathcal{N}[U, w]$. Then, every spline in $\mathcal{N}[U, w]$ belongs to Configuration 2.1.

An element of $\mathcal{N}[U, w]$ is shown in Fig. 3. The dashed curves are the B-spline functions, w and f , respectively, used for defining the NURBS function, $g = \frac{f}{w}$. We chose the following coefficients for defining f ,

$$[f_i] = \left[\frac{2}{3}, 0, \frac{1}{3}, \frac{1}{2}, \frac{1}{2}, \frac{1}{4} \right].$$

As can be seen, g has continuous first and second derivatives over I . It should also be noted that both f and w are in $\mathcal{B}[U]$ and thus have continuous first and second derivatives themselves.

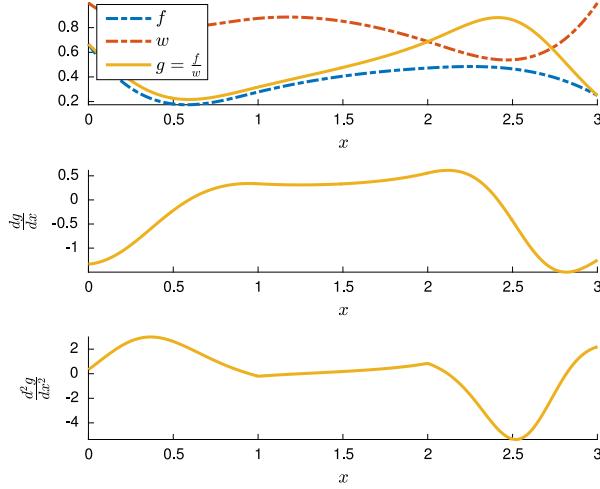


Fig. 3. A C^2 NURBS function defined over $I = [0, 3]$ using the knot vector given in Example 2.1. The NURBS function, g , is constructed as the ratio of two B-spline functions, f and w , where w acts as the weighting spline.

2.2. The multi-degree spline space $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$

As noted earlier, we are interested in creating a framework that allows us to work with smooth piecewise-NURBS curves for purposes of design and analysis. We define in this section multi-degree spline spaces. Their elements are piecewise-rational with any two adjacent pieces meeting with a certain order of smoothness.

Consider n_s open knot vectors $U^{(i)}$, $i = 1, \dots, n_s$, defined as in Eq. (2.1). We denote the left and right end points of $I^{(i)}$ associated to $U^{(i)}$ with $x_1^{(i)}$ and $x_2^{(i)}$, respectively. The vector $\mathbf{U} := (U^{(1)}, \dots, U^{(n_s)})$ is called an n_s -segment knot vector configuration. The multi-degree spline spaces will be constructed by considering NURBS spaces over the knot vectors $U^{(i)}$, which are glued together with certain smoothness requirements at the end points $x_2^{(i)}$ and $x_1^{(j_i)}$ for $i \in \{1, 2, \dots, n_s\}$ and $j_i := (i \bmod n_s) + 1$, thereby creating a periodically connected configuration. The equivalence class at the points $x_2^{(i)}$ and $x_1^{(j_i)}$ is called the i th segment join. In the following $\mathbb{N}_{-1} := \mathbb{N} \cup \{-1, 0\}$.

Definition 2.1 (Multi-Degree Spline Space). Consider an n_s -segment knot vector configuration \mathbf{U} . The NURBS space on the i th knot vector in \mathbf{U} is defined over the interval $I^{(i)} := [x_1^{(i)}, x_2^{(i)}] \subset \mathbb{R}$. For each segment, we define mappings,

$$\phi^{(i)} : I^{(i)} \rightarrow \hat{\Omega}^{(i)} : \phi^{(i)}(x) := x - x_1^{(i)} + \sum_{j=1}^{i-1} (x_2^{(j)} - x_1^{(j)}) + \xi_1^{(1)},$$

where $\hat{\Omega}^{(i)} := \phi^{(i)}(I^{(i)}) \subset \mathbb{R}$, and we construct the composed interval,

$$\hat{\Omega} := \hat{\Omega}^{(1)} \cup \dots \cup \hat{\Omega}^{(n_s)}.$$

Let $\mathbf{w} := (w^{(1)}, w^{(2)}, \dots, w^{(n_s)})$ be a vector consisting of n_s weighting splines $w^{(i)} \in \mathcal{B}[U^{(i)}]$. Given a multi-index $\boldsymbol{\kappa} := (\kappa^{(1)}, \kappa^{(2)}, \dots, \kappa^{(n_s)}) \in \mathbb{N}_{-1}^{n_s}$, we define the space of multi-degree splines as,

$$\begin{aligned} \mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}] := & \left\{ r : \hat{\Omega} \rightarrow \mathbb{R} : \forall i, r \circ \phi^{(i)} \in \mathcal{N}\left[U^{(i)}, w^{(i)}\right] \quad \& \dots \right. \\ & \left. \forall m \in \{0, \dots, \kappa^{(i)}\}, \frac{d^m(r \circ \phi^{(i)})}{dx^m}(x_2^{(i)}) = \frac{d^m(r \circ \phi^{(j_i)})}{dx^m}(x_1^{(j_i)}) \right\}. \end{aligned}$$

If all the weighting splines are constants, we denote the associated space of non-rational splines as,

$$\mathcal{S}[\mathbf{U}, \boldsymbol{\kappa}] := \mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}] \text{ when } \forall i, w^{(i)} \equiv c^{(i)} \in \mathbb{R}^+.$$

Moreover, we set $\hat{\Omega}^{(i)} =: [\xi_1^{(i)}, \xi_2^{(i)}]$. Note that $\xi_2^{(i)} = \xi_1^{(i+1)}$, $i \in \{1, \dots, n_s - 1\}$, and $\xi_1^{(1)} \in \mathbb{R}$ is an arbitrary origin for the composed interval $\hat{\Omega}$. For all practical purposes, hereafter we will assume $\kappa^{(i)} \leq \min(p^{(i)}, p^{(j)})$, where $p^{(i)}$ stands for the degree of the NURBS space defined on the i th segment.

Remark 2.2. From the periodic connectivity, [Definition 2.1](#) implies that $\kappa^{(n_s)}$ is the order of smoothness at the join of the n_s th segment with the 1st segment. If $\kappa^{(n_s)} \geq 0$, it means that the splines in $\mathcal{R}[U, \kappa, w]$ are periodic over $\hat{\Omega}$, and if $\kappa^{(n_s)} = -1$ then the splines in $\mathcal{R}[U, \kappa, w]$ are non-periodic over $\hat{\Omega}$.

Remark 2.3. If κ_1 and κ_2 are such that $\kappa_1^{(i)} \leq \kappa_2^{(i)}$ for all i , then $\mathcal{R}[U, \kappa_1, w] \supseteq \mathcal{R}[U, \kappa_2, w]$.

The elements of $\mathcal{R}[\cdot, \kappa, \cdot]$ are piecewise-NURBS functions such that the pieces meet with $C^{\kappa^{(i)}}$ continuity at the i th segment join. It is clear that classical NURBS spaces are a special case of the multi-degree spline spaces. The following simple result states this precisely.

Proposition 2.1. Consider $\hat{\Omega} \subset \mathbb{R}$, and let $\mathcal{N}[U, w]$ be a NURBS space defined on $\hat{\Omega}$ with degree p_0 . Then, by [Definition 2.1](#),

$$\exists U, \kappa, w : \mathcal{N}[U, w] = \mathcal{R}[U, \kappa, w], \quad (2.2)$$

where $\mathcal{R}[U, \kappa, w]$ is a uniform-degree ($= p_0$) spline space.

Proof. Eq. (2.2) follows directly from [Definition 2.1](#) by choosing a 1-segment configuration corresponding to U . \square

Example 2.2 (Uniform-Degree Spline). As in [Example 2.1](#), we are looking for a cubic spline function belonging to [Configuration 2.1](#). However, this time we consider the spline spaces from [Definition 2.1](#).

We choose to work with the following parameters for defining the space \mathcal{R} ,

- $n_s = 3$,
- knot vectors $\{U^{(i)}\}_{i=1}^{n_s}$,

$$U^{(i)} = [\underbrace{i-1, \dots, i-1}_{4 \text{ times}}, \underbrace{i, \dots, i}_{4 \text{ times}}],$$

- weights $\{w^{(i)}\}_{i=1}^{n_s}$,

$$\forall i, \quad [w_j^{(i)}] = \left[1, \frac{2}{3}, \frac{3}{4}, 1 \right],$$

- and continuity multi-index $\kappa = (2, 2, -1)$.

Then, all elements of $\mathcal{R}[U, \kappa, w]$ belong to [Configuration 2.1](#). Consider a function f defined over $\hat{\Omega}$ that is a cubic polynomial over each $\hat{\Omega}^{(i)}$ and is defined using the following coefficients,

$$[f_j^{(1)}] = \left[0, \frac{2}{9}, \frac{11}{36}, \frac{19}{51} \right], \quad [f_j^{(2)}] = \left[\frac{19}{51}, \frac{2}{9}, \frac{7}{36}, \frac{182}{1275} \right], \quad [f_j^{(3)}] = \left[\frac{182}{1275}, \frac{7}{900}, \frac{3}{80}, \frac{1}{10} \right].$$

The functions f and w (built using the weighting-spline coefficients) are shown in [Fig. 4](#) using dashed curves. Note that these are elements of $\mathcal{B}[U^{(i)}]$ over each $\hat{\Omega}^{(i)}$. The spline $r = \frac{f}{w}$ is therefore composed of elements of $\mathcal{N}[U^{(i)}, w^{(i)}]$. As can be seen from the bottom two plots in [Fig. 4](#), r has continuous first and second derivatives over $\hat{\Omega}$, while f and w are clearly non-smooth: the numerator and denominator do not need to be smooth in order for their ratio to be so! Contrast this with [Example 2.1](#) to see the additional flexibility offered by the notion of $\mathcal{R}[U, \kappa, w]$.

We now look for sufficient conditions for a multi-degree spline space to be contained in a single NURBS space. As will become clear, geometries can potentially be created using lower degree polynomials and a smaller number of degrees of freedom when the multi-degree framework is used instead of NURBS. Before outlining sufficient conditions, let us first explain what we mean by a *knot vector compatible with $\mathcal{S}[U, \kappa]$* .

Definition 2.2 (S Compatible Knot Vector). Consider a spline space $\mathcal{S}[U, \kappa]$. A knot vector U is said to be compatible with \mathcal{S} if,

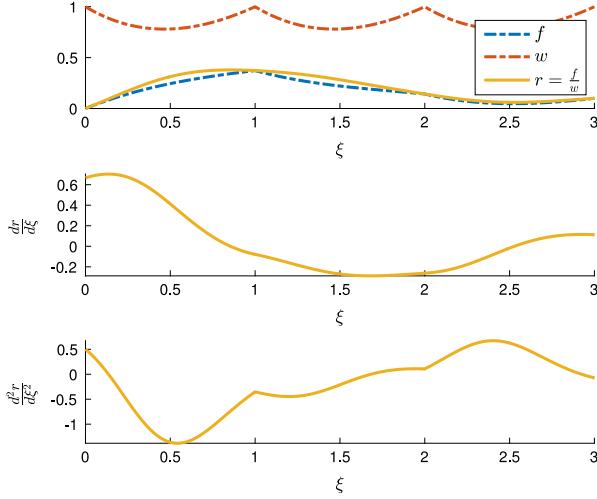


Fig. 4. The spline function r belonging to the space $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$ constructed in Example 2.2 has been plotted in yellow. It is constructed as the ratio of two piecewise B-spline functions, f and w . As can be seen, even though f and w are not smooth, r has continuous first and second derivatives everywhere. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- the degree of B-splines in $\mathcal{B}[\mathbf{U}]$ is $p_0 \geq \max_i p^{(i)}$,
- the elements of $\mathcal{B}[\mathbf{U}]$ are in the same continuity class as the elements of $\mathcal{S}[\mathbf{U}, \boldsymbol{\kappa}]$ everywhere on $\hat{\Omega}$.

Proposition 2.2. Consider a spline space $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$. Let \mathbf{U} be a knot vector compatible with $\mathcal{S}[\mathbf{U}, \boldsymbol{\kappa}]$ such that $\mathbf{w} \in \mathcal{B}[\mathbf{U}]$, where,

$$w|_{\hat{\Omega}^{(i)}} := v^{(i)} w^{(i)},$$

and $\{v^{(i)} : i = 1, \dots, n_s\}$ is a set of positive piecewise-polynomial functions. Then,

$$\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}] \subseteq \mathcal{N}[\mathbf{U}, \mathbf{w}].$$

Proof. Let r be an element of $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$, and assume that there exist n_s functions $v^{(i)}$ such that w as defined above is an element of $\mathcal{B}[\mathbf{U}]$. Let $f^{(i)} \in \mathcal{B}[U^{(i)}]$ be such that,

$$r|_{\hat{\Omega}^{(i)}} = \frac{f^{(i)}}{w^{(i)}} =: r^{(i)}.$$

This means,

$$r^{(i)} = \frac{v^{(i)} f^{(i)}}{v^{(i)} w^{(i)}}.$$

Since $r \in \mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$, for all i ,

- $r^{(i)}$ meets $r^{(j)}$ with $C^{\kappa^{(i)}}$ smoothness, and,
- $v^{(i)} w^{(i)}$ meets $v^{(j)} w^{(j)}$ with $C^{\kappa^{(i)}}$ smoothness.

Therefore, $v^{(i)} f^{(i)}$ must meet $v^{(j)} f^{(j)}$ with $C^{\kappa^{(i)}}$ smoothness. Let us define $f : \hat{\Omega} \rightarrow \mathbb{R}$ such that,

$$f|_{\hat{\Omega}^{(i)}} := v^{(i)} f^{(i)}.$$

Then, assuming that the degree of $v^{(i)}$ is $p_v^{(i)}$, we observe the following:

- $w \in \mathcal{B}[\mathbf{U}]$ and its degree is at most $\max_i (p^{(i)} + p_v^{(i)})$,
- f is a piecewise-polynomial function of degree at most $\max_i (p^{(i)} + p_v^{(i)})$, and,
- f is in the same continuity class as elements of $\mathcal{B}[\mathbf{U}]$ at every point in $\hat{\Omega}$.

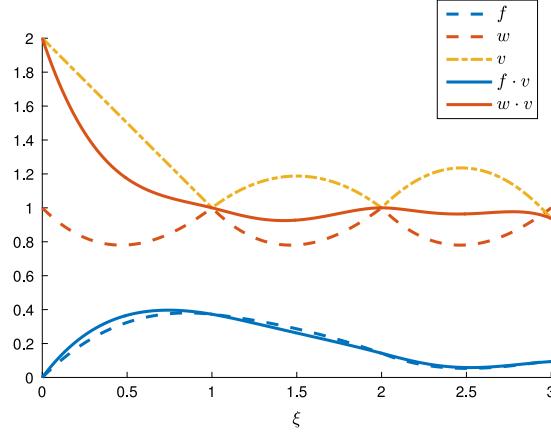


Fig. 5. The C^0 piecewise-cubic functions f and w shown in Fig. 4 can be transformed to C^2 quintic B-spline functions using an appropriate function v . See Example 2.3 for the numerical data used.

This means that $f \in \mathcal{B}[U]$ and we can express r as,

$$r = \frac{f}{w} \in \mathcal{N}[U, w]. \quad \square$$

Corollary 2.1. Consider a spline space $\mathcal{R}[U, \kappa, w]$ such that its elements are of uniform degree p_0 across all intervals, and let U be a knot vector compatible with $\mathcal{S}[U, \kappa]$ such that the splines in $\mathcal{B}[U]$ have degree at most p_0 , too. If there exist positive constants $v^{(i)} \in \mathbb{R}^+$, $i = 1, \dots, n_s$, such that $w \in \mathcal{B}[U]$, where,

$$w|_{\hat{\Omega}^{(i)}} := v^{(i)} w^{(i)},$$

then,

$$\mathcal{R}[U, \kappa, w] = \mathcal{N}[U, w].$$

Example 2.3 (Uniform-Degree Spline). Consider the spline function r from Example 2.2. The cubic weighting splines chosen there, shown in Fig. 4, are clearly only C^0 over $\hat{\Omega}$. However, we can find quadratic polynomials $v^{(i)}$ such that $w^{(i)} v^{(i)}$ form pieces of a quintic piecewise-polynomial that is C^2 over $\hat{\Omega}$.

Construct a piecewise function v using the quadratic polynomials $v^{(i)}$ defined over each $\hat{\Omega}^{(i)}$ with the following coefficients,

$$\begin{bmatrix} v_j^{(1)} \end{bmatrix} = \begin{bmatrix} 2, \frac{3}{2}, 1 \end{bmatrix}, \quad \begin{bmatrix} v_j^{(2)} \end{bmatrix} = \begin{bmatrix} 1, \frac{11}{8}, 1 \end{bmatrix}, \quad \begin{bmatrix} v_j^{(3)} \end{bmatrix} = \begin{bmatrix} 1, \frac{3}{2}, \frac{15}{16} \end{bmatrix}.$$

The functions f , w (from Example 2.2) and v are shown in Fig. 5, as well as the functions fv and wv (these are quintic piecewise-polynomials). As shown in Fig. 6, the functions fv and wv are actually C^2 over $\hat{\Omega}$. Then, choosing the knot vector compatible with $\mathcal{S}[U, \kappa]$ as,

$$U = [0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 3],$$

we have $fv, wv \in \mathcal{B}[U]$. Therefore, the spline function r built in Example 2.2 is a quintic NURBS function in $\mathcal{N}[U, wv]$.

Note that the dimension of $\mathcal{N}[U, wv]$ is 12 while, as shown later in Example 2.6, r can be represented using only 6 degrees of freedom and cubic piecewise-NURBS functions within the multi-degree framework from Definition 2.1.

The examples shown so far demonstrate some of the additional flexibility offered by the notion of multi-degree splines. In contrast to conventional NURBS spaces, these splines also allow for local degree elevation (see Section 2.4.3), and lead to convenient representations of geometries for the designer to work with. For instance, using these we can work with low degree parametrically smooth representations of conics as shown in Figs. 1 and 17.

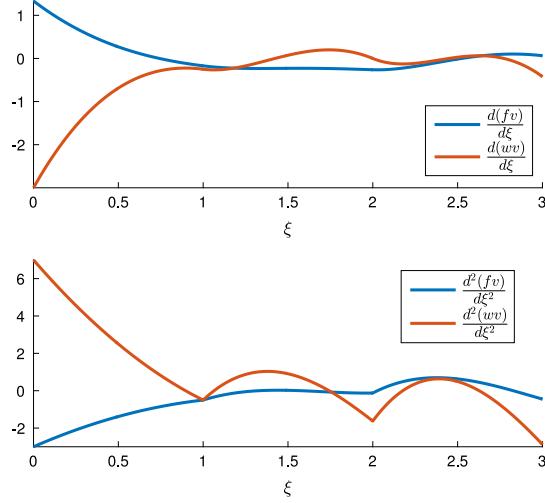


Fig. 6. The first and second derivatives of fv and wv from Fig. 5 have been plotted here, and they are clearly continuous over $\hat{\Omega}$.

2.3. A basis for $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$

In this section we present an algorithm to construct a basis spanning the multi-degree spline spaces. We do this by formulating the continuity constraints that the NURBS basis functions defined over adjacent segments on the real line would need to satisfy. The null-space of these constraints yields the coefficients of an extraction operator that help us to create the multi-degree basis functions as linear combinations of the NURBS basis functions.

Suppose we are looking to construct a basis for $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$. On the i th knot vector $\mathbf{U}^{(i)}$ in \mathbf{U} , we have $n^{(i)}$ unique NURBS basis functions $\hat{b}_{j,p^{(i)}}$ of degree $p^{(i)}$ that span $\mathcal{N}[\mathbf{U}^{(i)}, \mathbf{w}^{(i)}]$. We first map these basis functions onto adjacent segments on the real line using the maps $\phi^{(i)}$ in Definition 2.1. Let $\mu_i := \sum_{k=1}^i n^{(k)}$ and $\mu_0 := 0$. We define for $i \in \{1, \dots, n_s\}$, $j \in \{1, \dots, n^{(i)}\}$,

$$\hat{b}_{\mu_{i-1}+j}(\xi) := \begin{cases} \hat{b}_{j,p^{(i)}}(\phi^{(i)}(\xi)), & \text{if } \hat{\Omega}^{(i)} \ni \xi = \phi^{(i)}(\xi), \\ 0, & \text{if } \xi \notin \hat{\Omega}^{(i)}. \end{cases}$$

For the sake of simplicity, we dropped the reference to the degree in the notation. We arrange these mapped basis functions in a single (column) vector $\hat{\mathbf{b}}$ of length μ_{n_s} . Then, we want to construct a matrix \mathbf{H} of size $n \times \mu_{n_s}$ such that $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}] = \text{span}\{\hat{B}_1, \dots, \hat{B}_n\}$ where these functions are elements of the vector $\hat{\mathbf{B}}$ defined by,

$$\hat{\mathbf{B}} := \mathbf{H}\hat{\mathbf{b}}. \quad (2.3)$$

If \mathbf{H} is taken to be an identity matrix, then $\hat{\mathbf{B}} = \hat{\mathbf{b}}$ and we span the space $\mathcal{R}[\mathbf{U}, -\mathbf{1}, \mathbf{w}]$. In order to create a set of basis functions that span $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$, we build continuity constraints at all segment joins corresponding to $\boldsymbol{\kappa}$ and construct \mathbf{H} as (the transpose of) their null-space. Section 2.3.1 outlines the construction of continuity constraints while Section 2.3.2 presents a sparse null-space construction.

In the following we denote quantities associated to the i th and j_i th segments with superscripts L and R (left and right), respectively. Recall that j_i is defined as $(i \bmod n_s) + 1$. Quantities associated to their join will be denoted with superscript i .

2.3.1. Continuity constraints at the i th segment join

Let $\mathbf{K}^{(L)}$ be a matrix of size $(\kappa^{(i)} + 1) \times n^{(L)}$, whose m th row is given by,

$$\begin{bmatrix} 0 & \cdots & 0 & \frac{d^{m-1}}{d\xi^{m-1}} \hat{b}_{\mu_i-m+1}(\xi_2^{(i)}) & \cdots & \frac{d^{m-1}}{d\xi^{m-1}} \hat{b}_{\mu_i}(\xi_2^{(i)}) \end{bmatrix}, \quad (2.4)$$

Algorithm 1 Computation of \mathbf{H} (Section 2.3.1)

```

1:  $\kappa_0 \leftarrow \max_i \kappa^{(i)}$ 
2:  $\mathbf{H} \leftarrow$  identity matrix (size :  $\mu_{n_s} \times \mu_{n_s}$ )
3:  $\omega \leftarrow 0$ 
4: for  $m = 0 : \kappa_0$  do
5:   for  $i = 1 : n_s$  do
6:     if  $\kappa^{(i)} \geq m$  then
7:        $\bar{\mathbf{H}} \leftarrow$  sparse null-space of  $(m + 1)$ th row of  $\mathbf{K}^{(i)}$ 
8:        $\omega \leftarrow \omega + 1$ 
9:     else
10:       $\bar{\mathbf{H}} \leftarrow$  identity matrix (size :  $(\mu_{n_s} - \omega) \times (\mu_{n_s} - \omega)$ )
11:       $\mathbf{H} \leftarrow \bar{\mathbf{H}}^T \bar{\mathbf{H}}$ 
12:      for  $j = 1 : n_s$  do
13:         $\mathbf{K}^{(j)} \leftarrow \mathbf{K}^{(j)} \bar{\mathbf{H}}$ 
14: return  $\mathbf{H}$ 

```

Algorithm 2 Null-space of $\bar{\mathbf{c}}$ (Section 2.3.2)

```

1:  $\bar{\mathbf{H}}_1 \leftarrow \mathbf{0}$  (size:  $q \times (q - 1)$ )
2:  $\bar{\mathbf{H}}_1(1, 1) \leftarrow 1$ 
3: for  $i = 1 : q - 2$  do
4:    $\bar{\mathbf{H}}_1(i + 1, i) \leftarrow -\frac{\bar{c}_i \bar{\mathbf{H}}_1(i, i)}{\bar{c}_{i+1}}$ 
5:    $\bar{\mathbf{H}}_1(i + 1, i + 1) \leftarrow 1 - \bar{\mathbf{H}}_1(i + 1, i)$ 
6:  $\bar{\mathbf{H}}_1(q, q - 1) = 1$ 
7: return  $\bar{\mathbf{H}}_1$ 

```

and let $\mathbf{K}^{(R)}$ be a matrix of size $(\kappa^{(i)} + 1) \times n^{(R)}$, whose m th row is given by,

$$\left[-\frac{d^{m-1}}{d\xi^{m-1}} \hat{b}_{(\mu_i \bmod \mu_{n_s})+1} \left(\xi_2^{(i)} \right) \quad \cdots \quad -\frac{d^{m-1}}{d\xi^{m-1}} \hat{b}_{(\mu_i \bmod \mu_{n_s})+m} \left(\xi_2^{(i)} \right) \quad 0 \quad \cdots \quad 0 \right], \quad (2.5)$$

where $\xi_2^{(i)} := \phi^{(i)}(x_2^{(i)})$. Using these matrices, we can build the matrix $\mathbf{K}^{(i)}$ of size $(\kappa^{(i)} + 1) \times \mu_{n_s}$ which contains all constraints required to enforce $C^{\kappa^{(i)}}$ at $\xi_2^{(i)}$. This matrix is defined column-wise in the following manner:

1. the $(\mu_{i-1} + k)$ th column of $\mathbf{K}^{(i)}$ is equal to the k th column of $\mathbf{K}^{(L)}$,
2. the $((\mu_i \bmod \mu_{n_s}) + k)$ th column of $\mathbf{K}^{(i)}$ is equal to the k th column of $\mathbf{K}^{(R)}$, and,
3. all other columns of $\mathbf{K}^{(i)}$ are identically zero.

It can be easily verified that for a vector of coefficients f such that $\mathbf{K}^{(i)}f = \mathbf{0}$, the spline defined by $f^T \hat{b}$ is going to be $C^{\kappa^{(i)}}$ across $\xi_2^{(i)}$. Therefore, once all the matrices $\mathbf{K}^{(i)}$ have been assembled, the only remaining step is the construction of \mathbf{H} such that its rows span their null-spaces. The matrix \mathbf{H} is a *multi-degree spline extraction operator*, and we employ Algorithm 1 for its construction.

Essentially, in each step of the algorithm we construct the null space $\bar{\mathbf{H}}$ of a row of $\mathbf{K}^{(i)}$ (this is detailed in Section 2.3.2). Then, we update $\mathbf{K}^{(j)}$ (for all j) by post-multiplying them with $\bar{\mathbf{H}}$, thus obtaining a reduced system of constraints that need to be satisfied. The process is repeated until the constraints at all segment joins have been satisfied.

2.3.2. Sparse null-space construction

In this section we present an algorithm for the null-space computation motivated by the following observation: B-splines form a partition of unity, and the coefficient matrices (Bézier extraction) that enable their representation in terms of Bernstein polynomials are highly sparse. More precisely, B-splines have minimal support (see Lemma 2.1 and [16]) and are composed of the smallest possible number of Bernstein polynomials for given degree and continuity

requirements. Therefore, we strive to build the sparsest possible null-spaces of matrices $\mathbf{K}^{(i)}$ such that each row of the null-space matrix sums to 1. This second condition ensures we preserve partition of unity.

Assume that we are trying to resolve the m th constraint at the i th segment join. Then, at this stage, we have already resolved a total of ω constraints (see Line 8 in Algorithm 1). In other words, we have reduced the total number of degrees of freedom from μ_{n_s} to $(\mu_{n_s} - \omega) =: \tilde{\mu}_{n_s}$, and the size of \mathbf{H} at this stage is $\tilde{\mu}_{n_s} \times \mu_{n_s}$. Conceptually, the computation can be divided in two steps.

- Denote the m th row of $\mathbf{K}^{(i)}$ with $\mathbf{c} := [c_1, \dots, c_{\tilde{\mu}_{n_s}}]$, whose length is $\tilde{\mu}_{n_s}$. If $c_k = 0$, then a vector of length $\tilde{\mu}_{n_s}$ defined as,

$$[0, \dots, 0, \underbrace{1}_{k\text{th entry}}, 0, \dots, 0]^T,$$

is clearly in the null-space of \mathbf{c} . Assuming there are q non-zero entries in \mathbf{c} , we can construct $\tilde{\mu}_{n_s} - q$ vectors of the above form and they will all be in the null-space of \mathbf{c} . These vectors are arranged as columns of a matrix $\bar{\mathbf{H}}_0$. Its size is $\tilde{\mu}_{n_s} \times (\tilde{\mu}_{n_s} - q)$, and it is trivial to see that the columns of $\bar{\mathbf{H}}_0$ will be linearly independent. Therefore, the basis functions that do not contribute to the continuity constraints appear unchanged in the new set of basis functions being created.

- Referring to the indices of the NURBS basis functions involved in Eqs. (2.4) and (2.5), we set

$$\iota^{(L)} := \{\mu_i - m + 1, \dots, \mu_i\},$$

and,

$$\iota^{(R)} := \{(\mu_i \bmod \mu_{n_s}) + 1, \dots, (\mu_i \bmod \mu_{n_s}) + m\}.$$

Let $\iota := (i_1, i_2, \dots, i_q)$ be the indices of all non-zero entries in \mathbf{c} arranged such that,

- $i_1 = \min \{k : H_{k\alpha} \neq 0\}$ with α the minimum index in $\iota^{(L)} (= \mu_i - m + 1)$,
- $i_q = \max \{k : H_{k\beta} \neq 0\}$ with β the maximum index in $\iota^{(R)} (= (\mu_i \bmod \mu_{n_s}) + m)$,
- the vector $[i_2 - i_1, i_3 - i_2, \dots, i_q - i_{q-1}]$ contains no sign-changes if $i_1 \leq i_q$ and at most two sign-changes otherwise.

In other words, the indices in ι are in “increasing” order taking into account the periodic nature of the multi-degree framework. Then, select the sub-vector $\mathbf{c}(\iota)$ and denote it with $\bar{\mathbf{c}} := [\bar{c}_1, \dots, \bar{c}_q]$. The vector $\bar{\mathbf{c}}$ is such that its row-sum is equal to 0, a consequence of the partition of unity property of NURBS basis functions. We use Algorithm 2 to construct the null-space $\bar{\mathbf{H}}_1$ of $\bar{\mathbf{c}}$. This algorithm constructs $\bar{\mathbf{H}}_1$ such that all columns have 2 non-zero entries, and each row sums to 1. The size of $\bar{\mathbf{H}}_1$ is $q \times (q - 1)$.

Once $\bar{\mathbf{H}}_0$ and $\bar{\mathbf{H}}_1$ have been constructed, they can be assembled together (after appropriately padding $\bar{\mathbf{H}}_1$ with zeros) in the matrix $\bar{\mathbf{H}}$ and utilized in Algorithm 1, line 7. [Example 2.4](#) illustrates the procedure for a 2-segment configuration.

Example 2.4 (2-Segment Extraction Operator). Consider $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$, a periodic cubic spline space built over a 2-segment configuration with,

$$U^{(1)} = U^{(2)} = [0, 0, 0, 0, 1, 1, 1, 1], \quad \boldsymbol{\kappa} = (1, 1),$$

$$\left[w_j^{(1)} \right] = \left[w_j^{(2)} \right] = \left[1, \frac{1}{3}, \frac{1}{3}, 1 \right].$$

The NURBS basis functions over the two segments are shown in [Fig. 7](#) with solid and dashed lines. The constraint matrices $\mathbf{K}^{(i)}$ are evaluated as,

$$\mathbf{K}^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 \end{bmatrix}, \quad \mathbf{K}^{(2)} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

The first and second rows enforce equality of the values and first derivatives at the segment joins, respectively. Following Algorithm 1, we will have to resolve the constraints one row at a time by looping over the segments. The matrix \mathbf{H} is initialized as an identity matrix of size 8×8 .

In the following we state the relevant quantities computed during the resolution of each constraint: constraint-vector \mathbf{c} (with $\bar{\mathbf{c}}$ depicted as a sub-vector within a black frame inside it), ι , sparse null-space $\bar{\mathbf{H}}$ (with $\bar{\mathbf{H}}_1$ depicted as a sub-matrix within a red frame inside it), and the updated constraint-matrices $\mathbf{K}^{(i)}$.

- $C^0, i = 1$:

$$\mathbf{c} = \begin{bmatrix} 0 & 0 & 0 & \boxed{1 -1} & 0 & 0 & 0 \end{bmatrix}, \quad \iota = (4, 5),$$

$$\bar{\mathbf{H}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \begin{aligned} \mathbf{K}^{(1)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \end{bmatrix}, \\ \mathbf{K}^{(2)} &= \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}. \end{aligned}$$

- $C^0, i = 2$:

$$\mathbf{c} = \begin{bmatrix} \boxed{-1} & 0 & 0 & 0 & 0 & 0 & \boxed{1} \end{bmatrix}, \quad \iota = (7, 1),$$

$$\bar{\mathbf{H}} = \begin{bmatrix} \boxed{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \boxed{1} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{aligned} \mathbf{K}^{(1)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \end{bmatrix}, \\ \mathbf{K}^{(2)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -1 & 0 & 0 & 0 & -1 \end{bmatrix}. \end{aligned}$$

- $C^1, i = 1$:

$$\mathbf{c} = \begin{bmatrix} 0 & 0 & \boxed{-1 2 -1} & 0 \end{bmatrix}, \quad \iota = (3, 4, 5),$$

$$\bar{\mathbf{H}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \boxed{1 0} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \boxed{0 1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \begin{aligned} \mathbf{K}^{(1)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\ \mathbf{K}^{(2)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 2 & -1 & 0 & 0 & -1 \end{bmatrix}. \end{aligned}$$

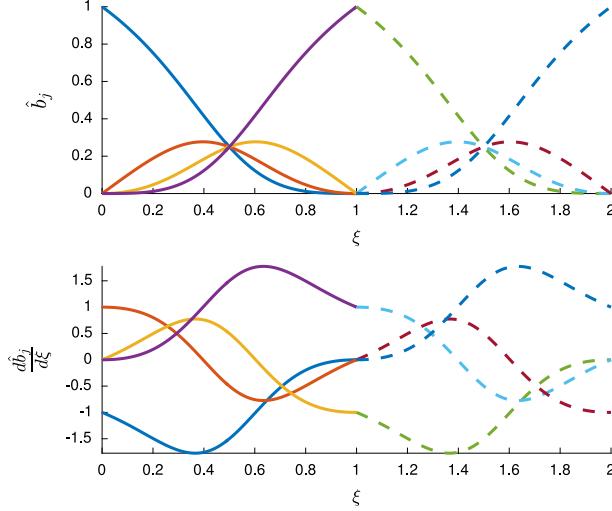


Fig. 7. Cubic NURBS basis defined over [0, 1] and [1, 2]. The basis shown is C^{-1} . See Example 2.4 for the numerical data used.

- C^1 , $i = 2$:

$$\mathbf{c} = \left[\begin{array}{|c|c|} \hline 2 & -1 \\ \hline \end{array} \right] 0 \ 0 \ \left[\begin{array}{|c|} \hline -1 \\ \hline \end{array} \right], \quad \boldsymbol{\iota} = (5, 1, 2),$$

$$\bar{\mathbf{H}} = \left[\begin{array}{|c|c|} \hline \frac{1}{2} & \frac{1}{2} \\ \hline 0 & 1 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline 1 & 0 \\ \hline \end{array} \right] \begin{array}{|c|c|} \hline 0 & 0 \\ \hline 0 & 0 \\ \hline 1 & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array}, \quad \mathbf{K}^{(1)} = \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right],$$

$$\mathbf{K}^{(2)} = \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right].$$

Therefore, at the end all constraints have been satisfied. We obtain the final extraction operator as,

$$\mathbf{H} = \left[\begin{array}{cccccc} \frac{1}{2} & 0 & 0 & 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & 0 \end{array} \right].$$

Applying this extraction operator, we obtain the 4 C^1 basis functions shown in Fig. 8.

2.3.3. Properties of the multi-degree spline extraction operator

The construction of \mathbf{H} (and $\bar{\mathbf{H}}$) as outlined above endows the functions \hat{B}_l with some very useful properties. These are stated in the following result.

Proposition 2.3. *The functions \hat{B}_l , $l = 1, \dots, n$, have local support, are linearly independent and form a partition of unity.*

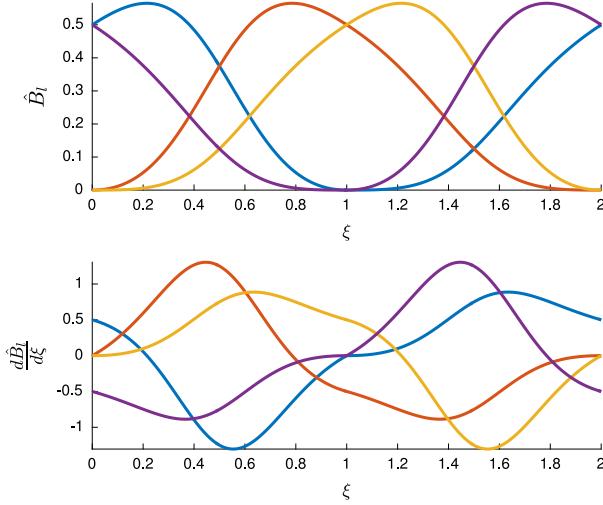


Fig. 8. As explained in Example 2.4, we can transform the discontinuous NURBS basis functions shown in Fig. 7 to the 4 C^1 spline basis functions shown here.

Proof. We always build a full-rank, sparsest possible null-space $\tilde{\mathbf{H}}$ in Line 7 of Algorithm 1, and each of its rows sum to 1. Then, by construction, \mathbf{H} is a full-rank, ultimately sparse matrix, and each of its columns sums to 1. Therefore, the functions \hat{B}_l defined using Eq. (2.3) have the following properties:

- they have local supports (from the sparsity of \mathbf{H} and from the minimal supports of the NURBS basis functions [16]),
- they are linearly independent (from the full rank of \mathbf{H}), and
- they form a partition of unity (from the column-sum of \mathbf{H} equal to 1). \square

Using the computed basis functions \hat{B}_l that span $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$, we can define a multi-degree spline function r using a vector of coefficients $[r_1, \dots, r_n]$,

$$r(\xi) = \sum_{l=1}^n r_l \hat{B}_l(\xi), \quad (2.6)$$

and a multi-degree spline curve \mathbf{r} in \mathbb{R}^d using a vector of control points $[\mathbf{r}_1, \dots, \mathbf{r}_n]$,

$$\mathbf{r}(\xi) = \sum_{l=1}^n \mathbf{r}_l \hat{B}_l(\xi). \quad (2.7)$$

Example 2.5 (C^1 Cubic Circle). Using the 4 C^1 spline basis functions shown in Fig. 8, we can assign the following control points to each to obtain a 4-point C^1 representation of the circle as shown in Fig. 1(c):

$$[\mathbf{r}_l] = [(-2, 1), (2, 1), (2, -1), (-2, -1)].$$

As can be seen in Fig. 1(d), moving \mathbf{r}_2 to $(2, 0)$ deforms the geometry smoothly. This is obviously a representation that would be preferred over the one in Fig. 1(b) when designing smooth geometries.

Example 2.6 (Uniform-Degree Spline). Going back to the spline space $\mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$ described in Example 2.2, the spline function r in Fig. 4 can be represented as in Eq. (2.6) using the coefficients,

$$[r_l] = \left[0, \frac{1}{3}, 1, -\frac{1}{3}, \frac{1}{20}, \frac{1}{10} \right].$$

Compare this with Example 2.3 where the same spline function was found to be an element of a quintic NURBS space with dimension 12. In this case, using the notion of multi-degree splines, we can build the same geometry with half the number of degrees of freedom and only cubic piecewise-NURBS functions.

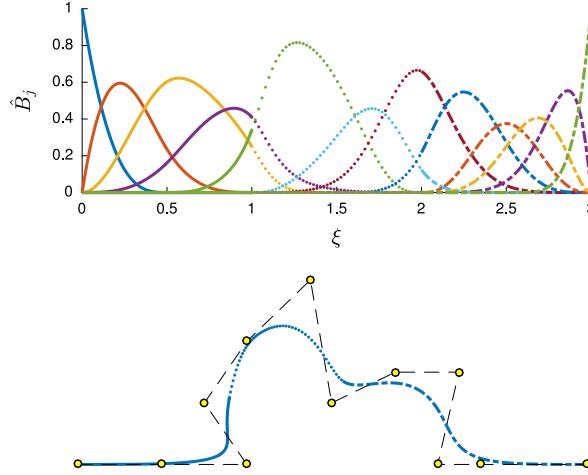


Fig. 9. A set of multi-degree spline basis functions are shown in the top plot, and a spline curve built using them in the bottom plot. The solid, dotted and dashed lines represent cubic, quartic and quintic rational parts, respectively. These basis functions have been built according to the data in Example 2.7.

Example 2.7 (Multi-Degree Spline). We want to construct a non-periodic C^2 spline function, $r : \hat{\Omega} = [0, 3] \rightarrow \mathbb{R}$ according to Configuration 2.1. However, this time we are interested in a spline function of non-uniform degree. For this purpose, let us define a 3-segment knot vector configuration, \mathbf{U} , using,

- knot vectors,

$$\mathbf{U}^{(i)} := \left[\underbrace{i-1, \dots, i-1}_{(3+i) \text{ times}}, i - \frac{1}{2}, \underbrace{i, \dots, i}_{(3+i) \text{ times}} \right],$$

- weights,

$$\left[w_j^{(1)} \right] = \left[1, 1, 1, \frac{5}{4}, 1 \right], \quad \left[w_j^{(2)} \right] = \left[1, \frac{3}{4}, 2, 1, 1, 1 \right], \quad \left[w_j^{(3)} \right] = [1, 1, 1, 1, 1, 1, 1],$$

- and continuity multi-index $\kappa = (2, 2, -1)$.

All elements of $\mathcal{R}[\mathbf{U}, \kappa, \mathbf{w}]$ belong to Configuration 2.1 and consist of cubic, quartic and quintic rational pieces.

The top plot in Fig. 9 shows the basis functions spanning $\mathcal{R}[\mathbf{U}, \kappa, \mathbf{w}]$. The solid, dotted and dashed parts represent the cubic, quartic and quintic parts of the spline functions, respectively. The basis functions are C^2 over $\hat{\Omega}$. The bottom plot in Fig. 9 shows a spline curve embedded in \mathbb{R}^2 built using the multi-degree spline basis functions along with the control points used for the construction.

2.3.4. Bézier extraction of B-splines as a special case

We now show that \mathbf{H} is a generalization of the Bézier extraction operator for classical B-splines. In other words, our multi-degree basis functions are simply B-splines when the degree is taken to be uniform over the entire domain.

Proposition 2.4. *The Bézier extraction operator of B-splines is a special case of the multi-degree spline extraction operator computed using Algorithms 1 and 2.*

Proof. Let $n_s = 2$ and initialize $\mathbf{U} = (\mathbf{U}^{(1)}, \mathbf{U}^{(2)})$, where,

$$\mathbf{U}^{(1)} = [\underbrace{0, \dots, 0}_{p+1 \text{ times}}, \underbrace{x_1, \dots, x_1}_{p+1 \text{ times}}], \quad \mathbf{U}^{(2)} = [\underbrace{x_1, \dots, x_1}_{p+1 \text{ times}}, \underbrace{x_2, \dots, x_2}_{p+1 \text{ times}}].$$

Let us number the $p + 1$ B-splines (of degree p) on each element (they are Bézier basis functions or Bernstein polynomials) such that the basis functions on $\mathbf{U}^{(1)}$ are $\hat{b}_{-p-1}, \dots, \hat{b}_{-1}$, and those on $\mathbf{U}^{(2)}$ are $\hat{b}_1, \dots, \hat{b}_{p+1}$.

Furthermore, we denote with $U(k)$ the B-spline knot-vector corresponding to a C^k join between Bézier segments $U^{(1)}$ and $U^{(2)}$,

$$U(k) := [\underbrace{0, \dots, 0}_{p+1 \text{ times}}, \underbrace{x_1, \dots, x_1}_{p-k \text{ times}}, \underbrace{x_2, \dots, x_2}_{p+1 \text{ times}}].$$

Now, set

$$\hat{B}_{l,-1} := \begin{cases} \hat{b}_{l-p-2}, & l = 1, \dots, p+1, \\ \hat{b}_{l-p-1}, & l = p+2, \dots, 2p+2. \end{cases}$$

We will prove using induction that the output basis functions constructed for $\mathcal{S}[U, \kappa]$ using Algorithm 1, where $\kappa = (k, -1)$ and $0 \leq k \leq p$, can be represented as,

$$\hat{B}_{l,k} = \begin{cases} \hat{B}_{l,k-1}, & l = 1, \dots, p-k, \\ \alpha_l \hat{B}_{l,k-1} + \beta_{l+1} \hat{B}_{l+1,k-1}, & l = p-k+1, \dots, p+1, \\ \hat{B}_{l+1,k-1}, & l = p+2, \dots, 2p-k+1, \end{cases} \quad (2.8)$$

and is the B-spline basis of $\mathcal{S}[U, \kappa]$.

- **Basic step:** For $k = 0$, the only Bézier basis functions that participate in smoothness constraints are $\hat{b}_{\pm 1}$. Therefore, only one new spline function $\hat{B}_{l,0}$ is constructed as a linear combination of $\hat{b}_{\pm 1}$, and the new set of spline functions is simply,

$$\{\hat{b}_{-p-1}, \dots, \hat{b}_{-2}\} \cup \{\alpha_{-1} \hat{b}_{-1} + \beta_1 \hat{b}_1\} \cup \{\hat{b}_2, \dots, \hat{b}_{p+1}\},$$

for some constants α_{-1} and β_1 . Thus, Eq. (2.8) holds. Moreover, since we know the supports of \hat{b}_i , it is obvious to see that with the knot-vector,

$$U(0) := [\underbrace{0, \dots, 0}_{p+1 \text{ times}}, \underbrace{x_1, \dots, x_1}_{p \text{ times}}, \underbrace{x_2, \dots, x_2}_{p+1 \text{ times}}],$$

the support of each $\hat{B}_{l,0}$ is $[u_l, u_{l+p+1}]$, i.e., the support of C^0 B-splines over $U(0)$. Then, using their partition of unity property (see Proposition 2.3) and Lemma 2.1, it follows that $\hat{B}_{l,0}$ are B-splines.

- **Induction step:** Let the claim be true for splines $\hat{B}_{l,\kappa}$ for some $0 \leq \kappa \leq k-1$, so $\hat{B}_{l,\kappa}$ are B-splines over $U(\kappa)$ and Eq. (2.8) holds. For $C^{\kappa+1}$ imposition, only the splines $\hat{B}_{l,\kappa}$ for $l = p-\kappa, \dots, p+2$ contribute to the smoothness constraint. The unchanged spline functions are simply set as,

$$\begin{aligned} \hat{B}_{l,\kappa+1} &= \hat{B}_{l,\kappa} = \hat{b}_{l-p-2}, & l = 1, \dots, p-\kappa-1, \\ \hat{B}_{l,\kappa+1} &= \hat{B}_{l+1,\kappa} = \hat{b}_{l-p+\kappa+1}, & l = p+2, \dots, 2p-\kappa. \end{aligned}$$

Following Algorithm 2, the other spline functions are constructed by considering at most two B-splines at a time, i.e.,

$$\hat{B}_{l,\kappa+1} = \alpha_l \hat{B}_{l,\kappa} + \beta_{l+1} \hat{B}_{l+1,\kappa}, \quad l = p-\kappa, \dots, p+1,$$

for some constants α_l and β_{l+1} . Then, once again, by inspection it is easy to see that for the knot-vector,

$$U(\kappa+1) := [\underbrace{0, \dots, 0}_{p+1 \text{ times}}, \underbrace{x_1, \dots, x_1}_{p-\kappa-1 \text{ times}}, \underbrace{x_2, \dots, x_2}_{p+1 \text{ times}}],$$

the support of each $\hat{B}_{l,\kappa+1}$ is $[u_l, u_{l+p+1}]$, i.e., the support of $C^{\kappa+1}$ B-splines over $U(\kappa+1)$. Then, using their partition of unity property (see Proposition 2.3) and Lemma 2.1, it follows that $\hat{B}_{l,\kappa+1}$ are B-splines.

Generalization of the result to $n_s > 2$ can be done in the same way. \square

Remark 2.4. Eq. (2.8) indicates that Algorithm 1 produces in each step a new B-spline basis with increasing smoothness. This corresponds to the process of removing knots from the initial knot vector configuration U . Knot removal algorithms are useful for approximation and data reduction [43].

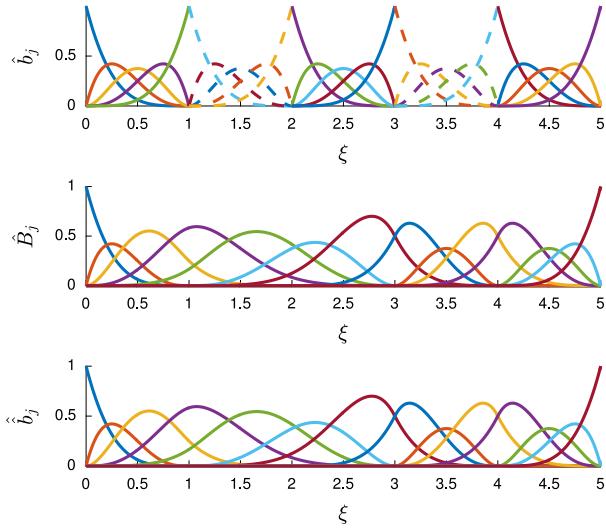


Fig. 10. The basis (\hat{B}_l) built for a uniform degree space $\mathcal{S}[U, \kappa]$ is exactly equivalent to the B-splines (\hat{b}_l) built on a knot vector compatible with it (see Example 2.8).

Example 2.8 (Uniform-Degree Spline Basis). Consider a 5-segment quartic spline space built using,

$$U^{(1)} = \dots = U^{(5)} = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1], \quad \kappa = (2, 3, 1, 1, -1), \\ [w_j^{(1)}] = \dots = [w_j^{(5)}] = [1, 1, 1, 1, 1].$$

The top plot in Fig. 10 shows C^{-1} B-splines over the 5 segments using alternating solid and dashed lines. The middle plot shows the smooth basis functions corresponding to $\mathcal{S}[U, \kappa]$, while the bottom plot shows the B-splines built using the knot vector,

$$U = [0, 0, 0, 0, 0, 1, 1, 2, 3, 3, 4, 4, 4, 5, 5, 5, 5].$$

Note that U is a knot vector compatible with $\mathcal{S}[U, \kappa]$. As can be seen, the basis functions in the middle and bottom plots are identical. Therefore, in this case the null-space computation algorithm simply computes \mathbf{H} as the Bézier extraction operator. This is in line with the result in Proposition 2.4.

2.3.5. Observations

We make the following observations about the multi-degree spline basis functions and their extraction operators.

- Comparing Examples 2.6 and 2.3, we see that the multi-degree splines could potentially be used to construct the same geometric objects with fewer degrees of freedom and with lower degree polynomials.
- As proved in Proposition 2.4, and illustrated in Example 2.8, Fig. 10, the extraction operator \mathbf{H} recovers classical B-splines in case of uniform degree. Of course, in this case the basis functions form a convex partition of unity and we have an IGA-suitable basis.
- Using symbolic computations we have verified that when $\max_i \kappa^{(i)} \leq 3$, \mathbf{H} contains only non-negative entries for all multi-degree spaces $\mathcal{S}[U, \kappa]$. As we already know that the basis functions form a partition of unity (Proposition 2.3), this means that we have an IGA-suitable basis. We conjecture that this result should hold for all κ .
- It is easy to verify that when $\max_i \kappa^{(i)} \leq 1$, \mathbf{H} contains only non-negative entries for all multi-degree spaces $\mathcal{R}[U, \kappa, w]$ too. However, this is not true in general when C^2 or higher smoothness is required and the weighting splines are allowed to be completely arbitrary. Nonetheless, as shown in Fig. 9, there do exist admissible weighting spline configurations that allow non-negative (and so IGA-suitable) basis functions; identifying this restricted subset is a topic left for future research.

5. At a segment join, only $\kappa + 1$ basis functions from each segment contribute towards C^κ continuity. Typically, we only require κ to be a small number (≤ 3) for both design and analysis. This implies that for only a modest number of basis functions on each segment, the continuity constraints at the i th segment join would be independent of those at the j th segment join for $i \neq j$. They could thus be resolved independently of each other, leading to extremely fast computations of \mathbf{H} in practice.
6. From the above point, for fixed κ , the construction of \mathbf{H} for $\mathcal{R}[\mathbf{U}, \kappa, \mathbf{w}]$ is not only inexpensive, but it is a constant-work procedure with respect to refinements made to any of the segments in \mathbf{U} . Indeed, as the non-zero parts of matrices $K^{(i)}$ remain unchanged in size for fixed $\kappa^{(i)}$, the computation cost of \mathbf{H} is going to remain unchanged as well when h -, p - or k -refinements are made to any or all of the segments. Moreover, variations in κ are also not a problem because we are only interested in nested spaces. Therefore, the entries of κ can only become smaller in the process of refinement, leading to a reduction in the cost of construction of \mathbf{H} . We utilize this in Section 2.4.3 to propose a first approach towards refinement of such spline spaces.

2.4. Further properties

In this section we discuss some further properties of the multi-degree spline representation. First, in Section 2.4.1 we focus on derivatives of multi-degree splines, and in Section 2.4.2 we derive a local representation in terms of NURBS basis functions. Then, in Section 2.4.3 we propose a simple and efficient procedure to do h -, p - or k -refinements. Finally, in Section 2.4.4 we prove that the multi-degree spline basis is a stable basis.

2.4.1. Derivatives

Once the multi-degree spline basis functions \hat{B}_l have been computed using Eq. (2.3), we can define their derivatives easily using those of NURBS basis functions as,

$$\frac{d^m \hat{\mathbf{B}}}{d\xi^m} = \mathbf{H} \frac{d^m \hat{\mathbf{b}}}{d\xi^m}.$$

For non-periodic spline spaces ($\kappa^{(n_s)} = -1$), a feature of the particular construction of \mathbf{H} is that the m th derivatives at the left (right) end of $\hat{\Omega}$ will be completely defined by the derivatives of the first (last) $m + 1$ basis functions on the first (last) segment, under the assumption $m \leq p^{(1)}$ ($m \leq p^{(n_s)}$).

Now, suppose $\kappa^{(n_s)} = -1$ and $1 \leq m \leq p^{(1)}$. A consequence of the partition of unity property of the basis functions is that at the left end point $\xi_1^{(1)}$,

$$\sum_{j=1}^{m+1} \frac{d^m \hat{B}_j}{d\xi^m} (\xi_1^{(1)}) = 0. \quad (2.9)$$

For a spline function r as in Eq. (2.6), let us look at the explicit expressions of some of its derivatives at $\xi_1^{(1)}$.

- $m = 1$: The first derivative at the left end involves only the first two basis functions and can be expressed using Eq. (2.9) as,

$$\frac{dr}{d\xi} (\xi_1^{(1)}) = r_1 \frac{d\hat{B}_1}{d\xi} (\xi_1^{(1)}) + r_2 \frac{d\hat{B}_2}{d\xi} (\xi_1^{(1)}) = (r_2 - r_1) \frac{d\hat{B}_2}{d\xi} (\xi_1^{(1)}).$$

- $m = 2$: The second derivative at the left end involves only the first three basis functions and can be expressed using Eq. (2.9) as,

$$\begin{aligned} \frac{d^2 r}{d\xi^2} (\xi_1^{(1)}) &= r_1 \frac{d^2 \hat{B}_1}{d\xi^2} (\xi_1^{(1)}) + r_2 \frac{d^2 \hat{B}_2}{d\xi^2} (\xi_1^{(1)}) + r_3 \frac{d^2 \hat{B}_3}{d\xi^2} (\xi_1^{(1)}) \\ &= (r_2 - r_1) \frac{d^2 \hat{B}_2}{d\xi^2} (\xi_1^{(1)}) + (r_3 - r_1) \frac{d^2 \hat{B}_3}{d\xi^2} (\xi_1^{(1)}). \end{aligned}$$

Higher derivatives can be similarly expanded and yield the following concise representation,

$$\frac{d^m r}{d\xi^m} (\xi_1^{(1)}) = \sum_{j=1}^m C_{m,j} (r_{j+1} - r_1), \quad (2.10)$$

where $C_{m,j} := \frac{d^m \hat{B}_{j+1}}{d\xi^m}(\xi_1^{(1)})$ are constant values. Of course, similar formulas hold for derivatives of spline curves \mathbf{r} as in Eq. (2.7).

2.4.2. From a multi-degree spline representation to a local NURBS representation

A multi-degree spline function r , as defined in Eq. (2.6), can be represented locally in terms of NURBS basis functions. Using Eq. (2.3) we obtain the coefficients for the associated NURBS representation,

$$r(\xi) = \sum_{l=1}^n r_l \hat{B}_l(\xi) = \sum_{l=1}^n r_l \sum_{j=1}^{\mu_{n_s}} H_{lj} \hat{b}_j(\xi) = \sum_{j=1}^{\mu_{n_s}} s_j \hat{b}_j(\xi), \quad (2.11)$$

where H_{lj} is the (l, j) th entry of \mathbf{H} , and

$$s_j := \sum_{l=1}^n H_{lj} r_l. \quad (2.12)$$

There are only few non-zero terms in the sum in Eq. (2.12). In the following we will make this statement more precise.

Proposition 2.5. *There are exactly $n^{(i)}$ basis functions \hat{B}_l non-zero over each interval $\hat{\Omega}^{(i)}$.*

Proof. The restrictions of basis functions \hat{B}_l to $\hat{\Omega}^{(i)}$ will span the NURBS space $\mathcal{N}[U^{(i)}, w^{(i)}]$, which has dimension $n^{(i)}$. The ultimately-sparse construction of \mathbf{H} implies that the minimal support property shown for the uniform-degree case in Proposition 2.4 carries over to the general multi-degree case. The minimal support property of \hat{B}_l implies that exactly $n^{(i)}$ basis functions will have non-zero restrictions on $\hat{\Omega}^{(i)}$. \square

The next result immediately follows from Propositions 2.3 and 2.5 and Eq. (2.11).

Corollary 2.2. *Given $r \in \mathcal{R}[U, \kappa, w]$, there exists a sub-matrix $\mathbf{M}^{(i)}$ of \mathbf{H} such that,*

- (i) $\mathbf{M}^{(i)}$ is a non-singular matrix of size $n^{(i)} \times n^{(i)}$,
- (ii) each column of $\mathbf{M}^{(i)}$ sums to 1,
- (iii) the NURBS coefficients of the spline $r|_{\hat{\Omega}^{(i)}}$ are given by

$$s_{\mu_{i-1}+j} = \sum_{l=1}^{n^{(i)}} M_{lj}^{(i)} r_{m+l}, \quad (2.13)$$

for $j \in \{1, \dots, n^{(i)}\}$ and for some integer m .

2.4.3. Refinement

As observed in Section 2.3.5, the construction of \mathbf{H} is an efficient operation in most practical cases, and is a constant-work process with respect to refinements made to any of the segments in \mathbf{U} . Therefore, refinement of $\mathcal{R}[U, \kappa, w]$ can be simply performed by,

- h -, p - or k -refinement of one or more of its segments, for which efficient algorithms exist, and/or
- lowering the required orders of smoothness by changing κ appropriately,

and by subsequent re-computation of \mathbf{H} for the modified \mathbf{U} , w and κ .

Recall that using Eq. (2.11), we can obtain NURBS control points s_j from the control points of a multi-degree spline curve \mathbf{r} using \mathbf{H} . Assuming that the h -, p - or k -refinement matrix for the i th segment is denoted by $\mathbf{R}^{(i)}$, we can obtain the modified NURBS control points for \mathbf{r} as,

$$[\tilde{s}_k] := \text{diag}(\mathbf{R}^{(1)}, \dots, \mathbf{R}^{(n_s)}) [s_j].$$

Once the refinement has been done, let the modified knot vector configuration be denoted by $\tilde{\mathbf{U}}$. At this stage, we can decrease the inter-segment smoothness by decreasing the entries of κ , if required. Next, we recompute the new multi-degree extraction operator, $\tilde{\mathbf{H}}$. Then, we are looking for control points $\tilde{\mathbf{r}}_j$ such that,

$$\tilde{\mathbf{H}}^T [\tilde{\mathbf{r}}_m] = [\tilde{s}_k].$$

The matrix $\tilde{\mathbf{H}}$ is highly sparse, and the new control points can be obtained by solving the above equation using a least-squares approach. Note that this does not mean that we are approximating the geometry. Indeed, since the original geometry was smooth and the entries in κ do not increase, the right-hand side of the equation is always in the range of $\tilde{\mathbf{H}}^T$ thereby yielding a unique solution to the least-squares problem.

The refinement operator that takes control points of the original geometry to those of the refined one can be represented as the matrix,

$$\bar{\mathbf{R}} = (\tilde{\mathbf{H}}\tilde{\mathbf{H}}^T)^{-1}\tilde{\mathbf{H}} \operatorname{diag}(\mathbf{R}^{(1)}, \dots, \mathbf{R}^{(n_s)})\mathbf{H}^T, \quad (2.14)$$

where $\mathbf{R}^{(i)}$ could represent any kind of h -, p - or k -refinement matrix for the i th segment.

Remark 2.5. Eq. (2.14) may be slightly misleading with regard to the anticipated cost of refinement. For all practical purposes ($\max_i \kappa^{(i)} \leq 3$ and a modest number of basis functions on each segment; see Section 2.3.5), only small, local systems are required to be solved. Indeed, in practice a large number of columns in $\tilde{\mathbf{H}}$ will only have a single non-zero entry equal to 1. In such a case, if the m th entry in the k th column is equal to 1, then we can simply set

$$\tilde{r}_m = \tilde{s}_k,$$

thereby reducing the number of unknowns by 1. This is illustrated in Example 2.9.

Remark 2.6. Obviously, we can choose to refine only some of the segments of \mathbf{U} , thus making it possible to perform local knot refinement or degree elevation on curves.

Example 2.9 (Refining the C^1 Circle). Fig. 11 illustrates the refinement procedure for the C^1 circle shown in Fig. 1(c); see also Examples 2.4 and 2.5. We perform,

- knot refinement: by inserting a knot at $\frac{1}{2}$ in both $U^{(1)}$ and $U^{(2)}$;
- degree elevation: by elevating the degree from 3 to 4 on both $U^{(1)}$ and $U^{(2)}$.

The extraction operators for the knot and degree refined geometries are both equal to $\tilde{\mathbf{H}}$,

$$\tilde{\mathbf{H}} = \begin{bmatrix} 1 & & & & & & & & & 1 \\ \frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{2} \end{bmatrix},$$

while the control points associated to the knot and degree refined NURBS segments are,

$$\begin{bmatrix} \tilde{s}_k^{\text{knot}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{1}{2} & 1 \\ 2 & 0 \\ \frac{1}{2} & -1 \\ 0 & -1 \\ 0 & -1 \\ -\frac{1}{2} & -1 \\ -2 & 0 \\ -\frac{1}{2} & 1 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} \tilde{s}_k^{\text{deg}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 0 \\ 1 & -1 \\ 0 & -1 \\ 0 & -1 \\ -1 & -1 \\ -2 & 0 \\ -1 & 1 \\ 0 & 1 \end{bmatrix},$$

respectively. Then, recalling Remark 2.5, we see that we can directly set the knot and degree refined control points for the C^1 geometry as,

$$\begin{bmatrix} \tilde{\mathbf{r}}_m^{\text{knot}} \\ \tilde{\mathbf{r}}_m^{\text{deg}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 1 \\ \frac{1}{2} & 0 \\ 2 & 0 \\ \frac{1}{2} & -1 \\ -\frac{1}{2} & -1 \\ -\frac{1}{2} & 0 \\ -2 & 0 \\ -\frac{1}{2} & 1 \end{bmatrix}, \quad \begin{bmatrix} \tilde{\mathbf{r}}_m^{\text{deg}} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \\ 1 & -1 \\ -1 & -1 \\ -2 & 0 \\ -1 & 1 \end{bmatrix},$$

respectively, without the need for solving a system similar to Eq. (2.14).

Example 2.10. Consider the multi-degree spline shown in Fig. 9, and let us refine $U^{(2)}$ and then $U^{(1)}$. The refined curves with the associated control nets are shown in Fig. 12. After each refinement, the control points that cannot be computed as outlined in Remark 2.5 are shown in the color red. At each stage, only two new control points need to be solved for and this can be done independently for both of them. Obviously, solving for these control points would be insignificant overhead in the process of refinement.

2.4.4. Stability

The stability of a basis is a crucial property for numerical computations. It means that small perturbations of the coefficients can only lead to small perturbations of the corresponding function representation. In this section we show that an IGA-suitable multi-degree spline basis $\{\hat{B}_l : l = 1, \dots, n\}$ is stable with respect to the infinity norm. In other words, there exist positive values A_1 and A_2 such that,

$$A_1^{-1} \max_{l=1,\dots,n} |r_l| \leq \max_{\xi \in \hat{\Omega}} |r(\xi)| \leq A_2 \max_{l=1,\dots,n} |r_l|, \quad (2.15)$$

for any spline function $r \in \mathcal{R}[\mathbf{U}, \boldsymbol{\kappa}, \mathbf{w}]$ represented as in Eq. (2.6). Both A_1 and A_2 should be bounded under mesh refinement. The right inequality clearly holds with $A_2 = 1$ because of the convex partition of unity property of the basis. The left inequality will be established in Proposition 2.6.

Let us first delve a little deeper into the (non-singular) sub-matrix $\mathbf{M}^{(i)}$ from Corollary 2.2 with respect to refinement. In particular, assuming that we continuously perform uniform knot refinements on *all* knot vectors in \mathbf{U} , let us analyze how the inverse of \mathbf{M} behaves. Assume for this purpose that we have carried out an arbitrary number of refinements, and let us denote the refined quantities with a tilde ($\tilde{\cdot}$) over them.

Consider the i th segment for arbitrary i , and consider the sub-matrix $\tilde{\mathbf{M}}^{(i)}$. Then, denote the 1-norm of the inverse of $\tilde{\mathbf{M}}^{(i)}$ with $\tilde{a}_0^{(i)}$. By Eq. (2.13),

$$\max_{l=1,\dots,\tilde{n}^{(i)}} |\tilde{r}_{m+l}| \leq \tilde{a}_0^{(i)} \max_{j=1,\dots,\tilde{n}^{(i)}} |\tilde{s}_{\mu_{i-1}+j}|.$$

Then, setting \tilde{A}_0 as the maximum of $\tilde{a}_0^{(i)}$ over i , the following holds,

$$\max_l |\tilde{r}_l| \leq \tilde{A}_0 \max_j |\tilde{s}_j|. \quad (2.16)$$

Moreover, assuming a sufficiently large number of refinements, we observe that $\tilde{\mathbf{M}}^{(i)}$ will have the structure,

$$\tilde{\mathbf{M}}^{(i)} = \begin{bmatrix} \tilde{\mathbf{M}}_1^{(i)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{M}}_2^{(i)} \end{bmatrix},$$

because only $\kappa^{(i)} + 1$ basis functions \hat{b}_j closest to the end-points of $\hat{\Omega}^{(i)}$ “see” the continuity constraints; the “inner” basis functions are included in the set of multi-degree basis functions \hat{B}_l without any modifications. Therefore, the

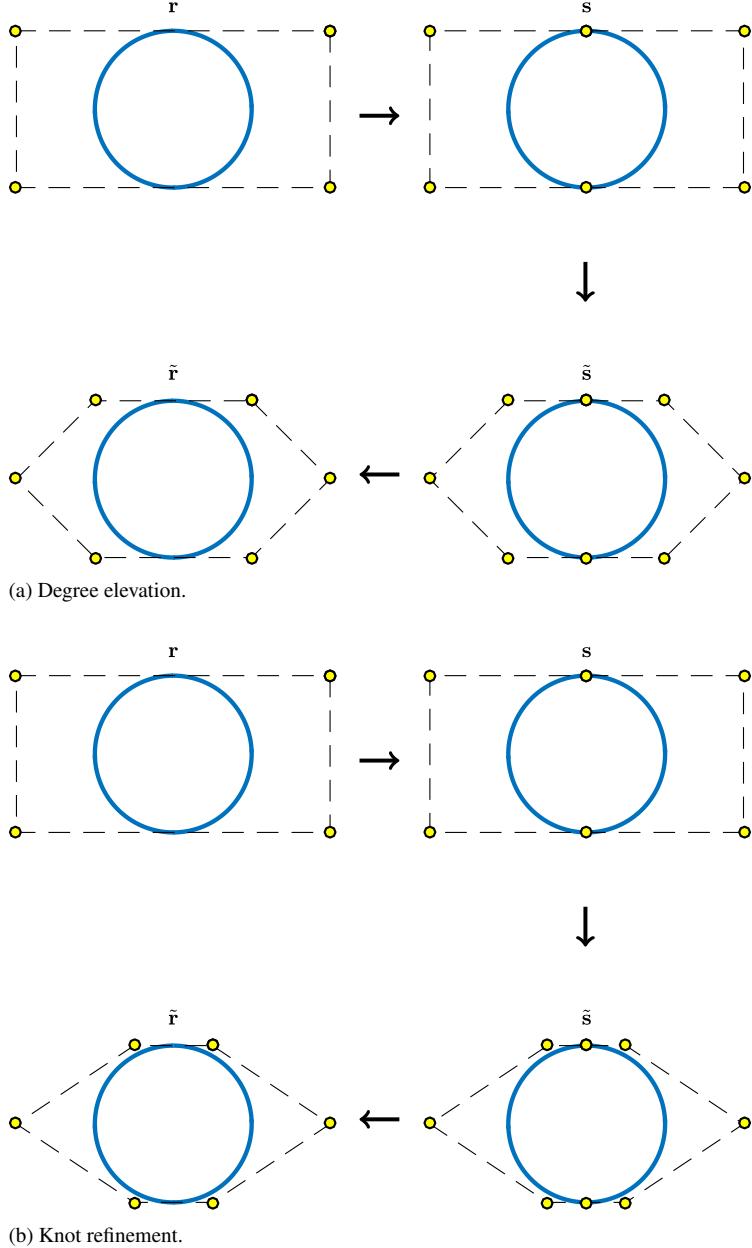


Fig. 11. The C^1 cubic circle shown in Fig. 1(c) is updated by degree elevating or knot refining both of its segments. The curve remains invariant and, moreover, no least-squares systems need to be solved for computing the control points of the refined geometry.

1-norm of $\tilde{\mathbf{M}}^{(i)}$ depends only on the 1-norms of $\tilde{\mathbf{M}}_1^{(i)}$ and $\tilde{\mathbf{M}}_2^{(i)}$. In the next lemma we show that this 1-norm can be bounded by a constant independent of the mesh-refinement.

Lemma 2.4. *There exists a constant $A_0 > 0$ such that the following uniform bound holds:*

$$\max_l |\tilde{r}_l| \leq A_0 \max_j |\tilde{s}_j|. \quad (2.17)$$

Proof. From Lemma 2.3 and Eqs. (2.4) and (2.5), we know that for a sufficiently large number of refinements, the appropriately scaled non-zero sub-vectors of constraints in $\tilde{\mathbf{K}}^{(i)}$ will approach the non-zero sub-vectors of constraints in $\mathbf{K}^{(i)}$, where $\mathbf{K}^{(i)}$ correspond to $\mathcal{S}[\bar{\mathbf{U}}, \kappa]$; in compliance with Lemmas 2.2 and 2.3, we take $\bar{\mathbf{U}}$ to be the original

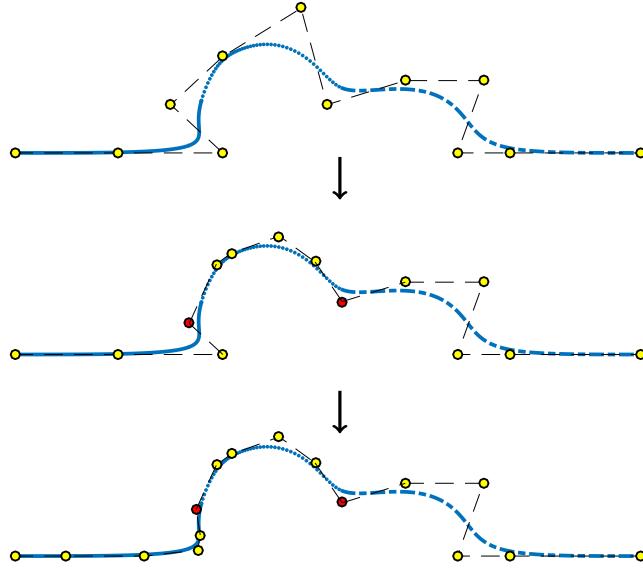


Fig. 12. The C^2 curve shown in Fig. 9 is updated by successively knot refining $U^{(2)}$ and $U^{(1)}$. The curve remains invariant under refinement, and the control points associated to each curve are shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

knot-vector configuration refined a finite number of times dependent only on the multi-degree. In other words, the constraint matrices converge to a quantity dependent only on the initial configuration \bar{U} . Therefore, $\tilde{\mathbf{M}}_1^{(i)}$ and $\tilde{\mathbf{M}}_2^{(i)}$ will also have a limit dependent only on the initial configuration. This reasoning combined with Eq. (2.16) leads us to Eq. (2.17). \square

We are now able to prove the left inequality in Eq. (2.15) using the result in Lemma 2.4.

Proposition 2.6. *For any spline function $r \in \mathcal{R}[\mathbf{U}, \kappa, \mathbf{w}]$ represented as in Eq. (2.6), there exists a constant $A > 0$ such that,*

$$\max_{l=1,\dots,n} |r_l| \leq A \max_{\xi \in \hat{\Omega}} |r(\xi)|. \quad (2.18)$$

Proof. We can use Eqs. (2.3) and (2.13) to obtain coefficients for the NURBS representation on the i th segment,

$$r|_{\hat{\Omega}^{(i)}} = r \circ \phi^{(i)} = \sum_{j=1}^{n^{(i)}} s_j \hat{b}_j.$$

From Definition 2.1 we recall that $r \circ \phi^{(i)} \in \mathcal{N}[U^{(i)}, w^{(i)}]$. By the stability of the NURBS representation, we know that there exists a constant $a_1^{(i)} > 0$ such that,

$$\forall i \in \{1, \dots, n_s\}, \quad \max_{j=1,\dots,n^{(i)}} |s_{\mu_{i-1}+j}| \leq a_1^{(i)} \max_{x \in I^{(i)}} \left| (r \circ \phi^{(i)})(x) \right|. \quad (2.19)$$

By combining Eqs. (2.17) and (2.19) we arrive at Eq. (2.18) with $A := A_0 \max_i a_1^{(i)}$. \square

2.5. Tensor-product splines

Consider the univariate spline spaces $\mathcal{R}^\xi := \mathcal{R}[\mathbf{U}, \kappa^\xi, \mathbf{w}^\xi]$ and $\mathcal{R}^\eta := \mathcal{R}[\mathbf{V}, \kappa^\eta, \mathbf{w}^\eta]$ defined over $\hat{\Omega}^\xi \subset \mathbb{R}$ and $\hat{\Omega}^\eta \subset \mathbb{R}$, respectively. Then, we can construct a tensor-product spline space over $\hat{\Omega} := \hat{\Omega}^\xi \times \hat{\Omega}^\eta \subset \mathbb{R}^2$ in the usual way,

$$\mathcal{R}^{\xi\eta} := \mathcal{R}^\xi \otimes \mathcal{R}^\eta,$$

where we omit the parameters on the left for brevity. Assuming that the sets of basis functions $\{\hat{B}_i^\xi : i = 1, \dots, n^\xi\}$ and $\{\hat{B}_j^\eta : j = 1, \dots, n^\eta\}$ span \mathcal{R}^ξ and \mathcal{R}^η , respectively, it is clear that the functions,

$$\hat{B}_{ij} := \hat{B}_i^\xi \hat{B}_j^\eta,$$

form the $n^\xi n^\eta$ basis functions spanning $\mathcal{R}^{\xi\eta}$. In the following, when talking about tensor-product spaces, if we omit one of the superscripts then this implies that we are talking about the univariate spaces used to build the tensor-product space.

3. Smooth multi-degree polar splines

In order to work with smooth polar splines in an integrated design and analysis environment, we first construct an intermediate parametric domain, Ω , called the *polar parametric domain*. It is built by collapsing one of the edges of a rectangular parametric domain $\hat{\Omega} \subset \mathbb{R}^2$ via a suitable polar map \mathbf{F} . The point to which the collapsed edge is mapped will be called the *polar point*. Then, given a tensor-product multi-degree spline space $\mathcal{R}^{\xi\eta} = \mathcal{R}^\xi \otimes \mathcal{R}^\eta$ defined on $\hat{\Omega}$ (see Section 2.5), we map its basis functions onto Ω using \mathbf{F} . The map \mathbf{F} is assumed to be sufficiently “nice” such that C^k smooth splines in $\mathcal{R}^{\xi\eta}$ mapped onto Ω are C^k everywhere except at the polar point. Apart from the collapsed edge, we do not allow any higher-order singularities in \mathbf{F} . This is made precise in Section 3.1. Appropriate imposition of smoothness constraints at the polar point enables us to compute an IGA-suitable extraction operator. This operator helps us to obtain a set of basis functions, called the *polar spline basis functions*, that are C^k smooth everywhere. We prove that our construction is always possible, and provide explicit details for $k \in \{0, 1, 2\}$.

3.1. Compatibility of $\mathcal{R}^{\xi\eta}$ and \mathbf{F}

The smoothness constraints at the polar point used to obtain C^k polar spline basis functions correspond to imposition of a linearly independent Hermite data set. Before detailing the specifics of our construction, we elaborate a little on what such a paradigm entails.

Let $\{\hat{B}_{ij} : i = 1, \dots, n^\xi; j = 1, \dots, n^\eta\}$ be the set of basis functions defined on $\hat{\Omega}$ that span $\mathcal{R}^{\xi\eta}$, and let $\{B_{ij} : i = 1, \dots, n^\xi; j = 1, \dots, n^\eta\}$ be their images under a suitable polar map \mathbf{F} from $\hat{\Omega}$ to Ω ,

$$(u, v) = \mathbf{F}(\xi, \eta), \quad (3.1)$$

so

$$B_{ij}(u, v) = B_{ij}(\mathbf{F}(\xi, \eta)) = \hat{B}_{ij}(\xi, \eta). \quad (3.2)$$

We assume that the functions \hat{B}_{ij} are periodic in ξ and non-periodic in η . In addition, let the degeneracy introduced by \mathbf{F} be such that $\mathbf{F}(\xi, 0) = (0, 0)$ for all ξ . Note that this necessitates that,

$$\forall m \geq 0, \quad \left. \frac{\partial^m \mathbf{F}}{\partial \xi^m} \right|_{\eta=0} \equiv \mathbf{0}. \quad (3.3)$$

We do not allow any other partial derivatives of \mathbf{F} to vanish. Then, for given coefficients s_{ij} , a spline function $s(u, v)$ over Ω can be constructed as usual,

$$s(u, v) = \sum_{i=1}^{n^\xi} \sum_{j=1}^{n^\eta} s_{ij} B_{ij}(u, v). \quad (3.4)$$

We can pull $s(u, v)$ back to $\hat{\Omega}$ as follows,

$$\hat{s}(\xi, \eta) := s(\mathbf{F}(\xi, \eta)) = \sum_{i=1}^{n^\xi} \sum_{j=1}^{n^\eta} s_{ij} B_{ij}(\mathbf{F}(\xi, \eta)) = \sum_{i=1}^{n^\xi} \sum_{j=1}^{n^\eta} s_{ij} \hat{B}_{ij}(\xi, \eta). \quad (3.5)$$

The derivatives of s and \hat{s} can be related using Eq. (B.3). For our immediate discussion, let us present an alternate, succinct presentation of this higher-order chain rule.

Let $\mathbf{F}(\xi, \eta) =: (F_u(\xi, \eta), F_v(\xi, \eta))$ and $\chi_m := \{(i_1, i_2) \in (\mathbb{N} \cup \{0\})^2 \setminus (0, 0) : i_1 + i_2 \leq m\}$. Then, Eq. (B.3) can be rewritten as follows for $m_1 + m_2 \geq 1$,

$$\frac{\partial^{m_1+m_2}\hat{s}}{\partial\xi^{m_1}\partial\eta^{m_2}}(\xi, \eta) = \sum_{i \in \chi_{m_1+m_2}} d_i^{m_1, m_2}(\xi, \eta) \frac{\partial^{i_1+i_2}s}{\partial u^{i_1}\partial v^{i_2}}(u, v), \quad (3.6)$$

where $d_i^{m_1, m_2}$ are sums of products of derivatives of the coordinate maps F_u and F_v . For s to be C^k smooth at the polar point, there must exist real values a_m such that,

$$\forall m := (m_1, m_2) \in (\mathbb{N} \cup \{0\})^2 : m_1 + m_2 \leq k, \quad \lim_{(u, v) \rightarrow (0, 0)} \frac{\partial^{m_1+m_2}s}{\partial u^{m_1}\partial v^{m_2}}(u, v) = a_m. \quad (3.7)$$

Since the spline space $\mathcal{R}^{\xi\eta}$ is defined on $\hat{\Omega}$, we compute the derivatives on Ω using the derivatives on $\hat{\Omega}$ and Eq. (3.6). However, as was mentioned earlier, we will build smooth polar spline basis functions by requiring them to satisfy certain Hermite data at the polar point, which belongs to Ω . Note that this is equivalent to specifying numbers a_m in the above equation, and thus specifying the derivatives on Ω and then computing what the corresponding derivatives on $\hat{\Omega}$ should be using Eq. (3.6).

Allowing $(u, v) \rightarrow (0, 0)$ in Eq. (3.6) and, equivalently, $(\xi, \eta) \rightarrow (\xi_0, 0)$, the above amounts to requiring the following to be true for all ξ_0 ,

$$\frac{\partial^{m_1+m_2}\hat{s}}{\partial\xi^{m_1}\partial\eta^{m_2}}(\xi_0, 0) = \sum_{i \in \chi_{m_1+m_2}} d_i^{m_1, m_2}(\xi_0, 0) a_i. \quad (3.8)$$

The upshot of this methodology is that, depending on the coordinate maps F_u , F_v , and on the desired order of smoothness k , the admissible spaces $\mathcal{R}^{\xi\eta}$ that can be used to construct C^k polar spline basis functions need to be rich enough to be able to satisfy certain restrictions. Let us explain what we mean in more detail.

In Eq. (3.8), the right-hand side is a linear combination of $d_i^{m_1, m_2}(\xi_0, 0)$, recalling that, in general, $d_i^{m_1, m_2}$ are sums of products of derivatives of F_u and F_v . For the time-being, let us assume that F_u and F_v are piecewise-polynomial splines of multi-degree $\mathbf{p} = (p^{(1)}, \dots, p^{(n_s)})$ in ξ . Then, the products of their derivatives may be piecewise-polynomial in ξ with multi-degree $\tilde{\mathbf{p}} > \mathbf{p}$. In fact, the space \mathcal{R}^ξ should be rich enough to contain k th powers of $F_{u,01}|_{\eta=0}$ and $F_{v,01}|_{\eta=0}$. This can be argued as follows. Eq. (B.3) simplifies at $\eta = 0$ for any $m_1 + m_2 \geq 1$ as,

$$\frac{\partial^{m_1+m_2}\hat{s}}{\partial\xi^{m_1}\partial\eta^{m_2}}(\xi_0, 0) = \sum_{i \in \chi_{m_1+m_2}} C_i D_{\eta, i}^{m_1, m_2} s(\mathbf{F}(\xi_0, 0)).$$

This is obvious for $m_1 = 0$, and for $m_1 > 0$ we have that $D_{\xi, i}^{m_1}$ in Eq. (B.1) is always zero due to Eq. (3.3). By inspecting the above equation, we see that for $1 \leq m_1 + m_2 \leq k$, the highest multi-degree in ξ will be obtained when $m_1 = 0$ and $m_2 = k$. Indeed, in this case,

$$\frac{\partial^k \hat{s}}{\partial \eta^k}(\xi_0, 0) = \left(F_{u,01}(\xi_0, 0) \frac{\partial}{\partial u} + F_{v,01}(\xi_0, 0) \frac{\partial}{\partial v} \right)^k s(\mathbf{F}(\xi_0, 0)) + \dots. \quad (3.9)$$

The first term on the right-hand side will contain k th powers of $F_{u,01}(\xi_0, 0)$ and $F_{v,01}(\xi_0, 0)$, thereby leading to splines of multi-degree $k\mathbf{p}$ in ξ . At the same time, on the left-hand side we simply have an element of \mathcal{R}^ξ . Therefore, to be able to satisfy Eq. (3.9), the space \mathcal{R}^ξ must contain k th powers of $F_{u,01}|_{\eta=0}$ and $F_{v,01}|_{\eta=0}$. Other combinations of (m_1, m_2) impose similar restrictions on the derivative spaces of \mathcal{R}^ξ , as long as the functions $F_u|_{\eta=0}$ and $F_v|_{\eta=0}$ are in the same continuity class as elements of \mathcal{R}^ξ . Indeed, it is clear that the highest-order derivatives of \mathbf{F} that appear in $D_{\eta, i}^{m_1, m_2}$ are differentiated m_1 times with respect to ξ , see Eq. (B.2). This is exactly the number of times the left-hand side has been differentiated with respect to ξ .

The above exposition motivates the following compatibility restrictions on $F_u, F_v \in \mathcal{R}_F^{\xi\eta}$ for imposition of C^k smoothness using Eq. (3.8).

Definition 3.1 (*Compatibility of $\mathcal{R}_F^{\xi\eta}$ and $\mathcal{R}^{\xi\eta}$*). A space of mappings $\mathcal{R}_F^{\xi\eta}$ is said to be k -compatible with the space $\mathcal{R}^{\xi\eta}$ if

- (i) $\mathcal{R}_F^{\xi\eta}$ contains C^k functions of ξ and η that are periodic in ξ and non-periodic in η ;
- (ii) for any $F \in \mathcal{R}_F^{\xi\eta}$ it holds $(F|_{\eta=0})^k \in \mathcal{R}^\xi$.

Thus, roughly speaking, for imposition of C^k smoothness, for instance, \mathcal{R}^ξ must contain piecewise polynomials of degree kp , when F_u and F_v restricted to $\eta = 0$ are piecewise polynomials of degree p ; this will become more clear in Section 3.3.4.

In line with the above exposition, we will assume that the following holds:

- (i) $F_u, F_v \in \mathcal{R}_F^{\xi\eta}$ and $\mathcal{R}^{\xi\eta}$ are both tensor-product (multi-degree) spline spaces;
- (ii) for a given k , the spaces $\mathcal{R}_F^{\xi\eta}$ and $\mathcal{R}^{\xi\eta}$ are k -compatible according to Definition 3.1, and,
- (iii) for $k \geq 2$ we will limit $\mathcal{R}_F^{\xi\eta}$ and $\mathcal{R}^{\xi\eta}$ to only non-rational splines in ξ direction.

We will denote with \tilde{B}_{ij} and \hat{B}_{ij} the basis functions spanning $\mathcal{R}_F^{\xi\eta}$ and $\mathcal{R}^{\xi\eta}$, respectively. However, when it is possible to choose $\mathcal{R}_F^{\xi\eta} = \mathcal{R}^{\xi\eta}$, i.e., for $k = 0$ and $k = 1$, we will denote the basis functions for both spaces using \hat{B}_{ij} . Furthermore, we will also assume that we only work with IGA-suitable multi-degree spline basis functions \tilde{B}_{ij} and \hat{B}_{ij} that, therefore, form a convex partition of unity.

Given suitable spline spaces $\mathcal{R}_F^{\xi\eta}$ and $\mathcal{R}^{\xi\eta}$, we present specifics regarding the construction of \mathbf{F} in Section 3.2. Thereafter, we state in Section 3.3 the precise recipe used for construction of C^k smooth polar spline basis functions as linear combinations of the basis functions that span $\mathcal{R}^{\xi\eta}$. It will be shown that it is always possible to construct them such that they form a convex partition of unity.

3.2. Polar parametric domain

Assume that we are given a spline space $\mathcal{R}_F^{\xi\eta}$ defined on a rectangular parametric domain $\hat{\Omega}$, and denote its set of spanning basis functions with $\{\tilde{B}_{ij} : i = 1, \dots, \tilde{n}^\xi; j = 1, \dots, \tilde{n}^\eta\}$. Assign the control point $(\rho_j \cos(\theta_i), \rho_j \sin(\theta_i)) \in \mathbb{R}^2$ to the basis function \tilde{B}_{ij} where,

$$\rho_j := \frac{j-1}{\tilde{n}^\eta - 1} \in [0, 1], \text{ and } \theta_i := 2\pi \times \frac{i-1}{\tilde{n}^\xi} \in [0, 2\pi]. \quad (3.10)$$

The above choice of control-point values has been made in the interest of standardization and is not unique. Using these control points, we can construct a disk-like domain, Ω , with the aid of the map $\mathbf{F} : \hat{\Omega} \mapsto \Omega$ in Eq. (3.1) defined as,

$$\mathbf{F}(\xi, \eta) := (F_u(\xi, \eta), F_v(\xi, \eta)) := \sum_{i=1}^{\tilde{n}^\xi} \sum_{j=1}^{\tilde{n}^\eta} (\rho_j \cos(\theta_i), \rho_j \sin(\theta_i)) \tilde{B}_{ij}(\xi, \eta). \quad (3.11)$$

We will call Ω the *polar parametric domain* or, simply, the *polar domain*. Note that the above construction will not always yield an exactly circular domain Ω (depending on the choice of the space $\mathcal{R}_F^{\xi\eta}$). This will serve as the reference element for polar configurations. It is clear that,

$$\forall \xi \in \hat{\Omega}^\xi, \quad \mathbf{F}(\xi, 0) = (0, 0), \quad (3.12)$$

where $(0, 0) \in \Omega$ is called the *polar point*. See Fig. 13 for an example of such a construction.

Now, given a suitable space $\mathcal{R}^{\xi\eta}$ defined on $\hat{\Omega}$, recall that its basis functions are denoted with \hat{B}_{ij} and their images under \mathbf{F} with B_{ij} as in Eq. (3.2). The basis functions \hat{B}_{ij} are assumed to be at least k -times continuously differentiable everywhere on $\hat{\Omega}$. Therefore, B_{ij} are going to be k -times continuously differentiable everywhere on Ω except at the polar point because of the degeneracy introduced by \mathbf{F} . Spline functions can be built on Ω using B_{ij} as shown in Eq. (3.4). Their values and derivatives can be pulled back to $\hat{\Omega}$ using Eqs. (3.5) and (3.6). Note that not all linear combinations of B_{ij} will yield smooth spline functions.

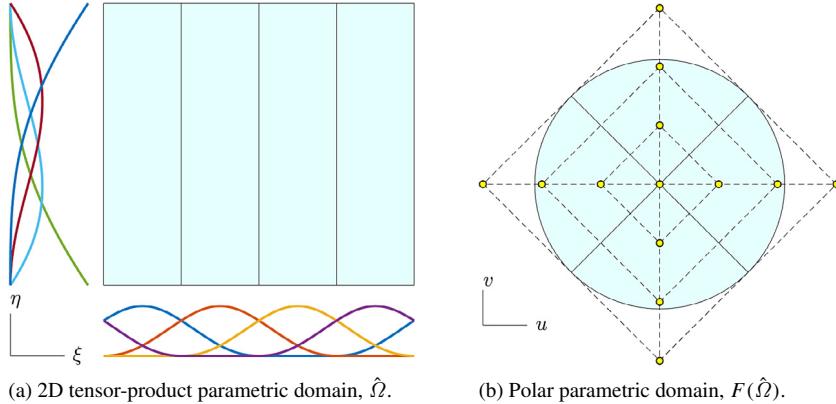


Fig. 13. The left figure shows an example of a 2D parametric domain which serves as the starting point for the construction of the polar parametric domain shown in the right figure via Eq. (3.11). The univariate basis functions shown are used to create the tensor-product basis functions \tilde{B}_{ij} used in Eq. (3.11). The control points from Eq. (3.10) have been marked as yellow disks in the right figure. The numerical data used to make these figures is outlined in Example 3.1.

Example 3.1 (Polar Parametric Domain). Consider univariate spaces \mathcal{R}_F^ξ and \mathcal{R}_F^η built using,

$$\begin{aligned} U &= (U, U, U, U), \quad \kappa^\xi = (1, 1, 1, 1), \quad w^\xi = (w, w, w, w), \\ V &= (V), \quad \kappa^\eta = (-1), \quad w^\eta = \left(\left[1, \frac{1+\sqrt{2}}{3}, \frac{1+\sqrt{2}}{3}, 1 \right] \right), \end{aligned}$$

where,

$$U = [0, 0, 0, 1, 1, 1], \quad w = \left[1, \frac{1}{\sqrt{2}}, 1 \right], \quad V = [0, 0, 0, 0, 1, 1, 1, 1].$$

They contain quadratic and cubic splines, respectively, with the former being periodic. The associated univariate basis functions are shown in Fig. 13(a), along with the tensor-product parametric domain of definition, $\hat{\Omega}$.

Following Eqs. (3.10) and (3.11), we can map this rectangular domain onto the polar parametric domain Ω shown in Fig. 13(b). The boundary of Ω is C^1 and is, in fact, an exact circle made up of quadratic rational pieces. This boundary description is equivalent to the C^1 description used in [13].

3.3. C^k polar splines

As noted earlier, not all linear combinations of basis functions B_{ij} will give us the required smoothness at the polar point. Therefore, in this section we look at ways to obtain minimally constrained configurations that yield the desired smoothness at $(0, 0)$. For this purpose, we present in Section 3.3.1 a recipe that can be used to construct C^k polar splines for arbitrary $k \geq 0$. Subsequently, we detail explicit constructions for $k \in \{0, 1, 2\}$ in Sections 3.3.2–3.3.4. From our experience, C^1 and C^2 are the most common requirements imposed on geometries and solutions in geometric modeling and engineering analysis. The C^0 construction is standard and simple, and is presented only to help comprehension.

3.3.1. Polar spline extraction operator

Let us arrange the set of basis functions $\{B_{ij} : i = 1, \dots, n^\xi; j = 1, \dots, n^\eta\}$ in a vector \mathbf{B} , where B_{ij} occupies the $(i + (j - 1)n^\xi)$ th entry. Our goal is to construct smooth polar basis functions on the polar domain as suitable linear combinations of the functions B_{ij} . In other words, given the desired order of smoothness k , we are looking for an extraction operator \mathbf{E}^k such that the polar spline basis functions defined by the following relation,

$$\mathbf{N} := \mathbf{E}^k \mathbf{B}, \tag{3.13}$$

are C^k at the polar point.

For given j , the set $\{B_{ij} : i = 1, \dots, n^\xi\}$ is called the $(j - 1)$ th ring of basis functions. From Section 2.4.1, only the 0th to k th rings of basis functions contribute towards C^k continuity conditions at the polar point; this is a total of $n_o := n^\xi(k + 1)$ basis functions. Our approach towards a flexible C^k polar construction is to replace these by a total of $n_k := (k + 1)(k + 2)/2$ smooth polar basis functions. The j th ring basis functions for all $j > k$ already satisfy the C^k continuity conditions at the polar point (their derivatives are identically zero there up to the k th order), so they can be included without modifications in the set of polar basis being created. This dictates that \mathbf{E}^k must have $n := (n^\xi n^\eta - n_o + n_k)$ rows and $(n^\xi n^\eta)$ columns. The extraction operator takes the following sparse block-diagonal form (thereby enforcing locality of the basis functions),

$$\mathbf{E}^k = \begin{bmatrix} \bar{\mathbf{E}}^k & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^k \end{bmatrix}, \quad (3.14)$$

where \mathbf{I}^k is an identity matrix of size $(n - n_k) \times (n - n_k)$ and $\bar{\mathbf{E}}^k$ is a matrix of size $n_k \times n^\xi(k + 1)$. The entry of $\bar{\mathbf{E}}^k$ corresponding to its l th row and $(i + (j - 1)n^\xi)$ th column will be denoted with $\bar{E}_{l,(ij)}^k$. We can then rewrite Eq. (3.13) as follows for $l = 1, 2, \dots, n_k$,

$$N_l(u, v) = \sum_{i=1}^{n^\xi} \sum_{j=1}^{k+1} \bar{E}_{l,(ij)}^k B_{ij}(u, v). \quad (3.15)$$

We can pull these back to $\hat{\Omega}$ using Eq. (3.5) to obtain the equivalent representation,

$$\hat{N}_l(\xi, \eta) = \sum_{i=1}^{n^\xi} \sum_{j=1}^{k+1} \bar{E}_{l,(ij)}^k \hat{B}_{ij}(\xi, \eta). \quad (3.16)$$

With respect to geometric modeling and engineering analysis, we are interested in polar spline basis functions which form a convex partition of unity. Therefore, we would like \mathbf{E}^k (or equivalently $\bar{\mathbf{E}}^k$) to satisfy the conditions given in Definition 1.1.

As mentioned earlier, we will enforce C^k continuity at the polar point by requiring the basis functions N_l to satisfy a linearly independent Hermite data set at the polar point. To this end, we will first define n_k source basis functions $\{T_l : l = 1, \dots, n_k\}$, that provide us with the appropriate Hermite data. In this work, we choose basis functions T_l such that they can reproduce all bivariate polynomials of total degree less than or equal to k on a neighborhood of $(0, 0)$. In particular, we choose them as *triangular Bernstein polynomials* [44] defined on a particular domain triangle.

Definition 3.2 (*Triangular Bernstein Polynomials*). Given a domain triangle \mathcal{T}_k with vertices $\mathbf{v}_1, \mathbf{v}_2$ and \mathbf{v}_3 , we can define $n_k := (k + 1)(k + 2)/2$ triangular Bernstein polynomials of degree k . These polynomials are indexed by triplets $(i_1, i_2, i_3) \in (\mathbb{N} \cup \{0\})^3$, where $i_1 + i_2 + i_3 = k$, and the (i_1, i_2, i_3) th polynomial is defined as,

$$T_{i_1 i_2 i_3}(u, v) := \binom{k}{i_1 \ i_2 \ i_3} (\lambda_1)^{i_1} (\lambda_2)^{i_2} (\lambda_3)^{i_3},$$

where $(\lambda_1, \lambda_2, \lambda_3)$ are the unique barycentric coordinates of point (u, v) with respect to \mathcal{T}_k such that,

$$\lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3 = (u, v), \quad \lambda_1 + \lambda_2 + \lambda_3 = 1.$$

The n_k triangular Bernstein polynomials of degree k are non-negative on their domain triangle \mathcal{T}_k . Moreover, they are linearly independent, they form a partition of unity, and they span the space of bivariate polynomials of total degree less than or equal to k . The Bernstein polynomial $T_{i_1 i_2 i_3}$ is usually associated to the domain point $\mathbf{t}_{i_1 i_2 i_3}$ which has as barycentric coordinates $(\frac{i_1}{k}, \frac{i_2}{k}, \frac{i_3}{k})$ with respect to \mathcal{T}_k ; see Fig. 14. Let us arrange the triplets (i_1, i_2, i_3) in a linear ordering defined by,

$$(i_1, i_2, i_3) < (j_1, j_2, j_3) \Leftrightarrow (i_3 < j_3) \vee ((i_3 = j_3) \wedge (i_2 < j_2)).$$

Then, assuming (i_1, i_2, i_3) is the l th element in this ordering, we take the l th triangular Bernstein polynomial as,

$$T_l \equiv T_{i_1 i_2 i_3}.$$

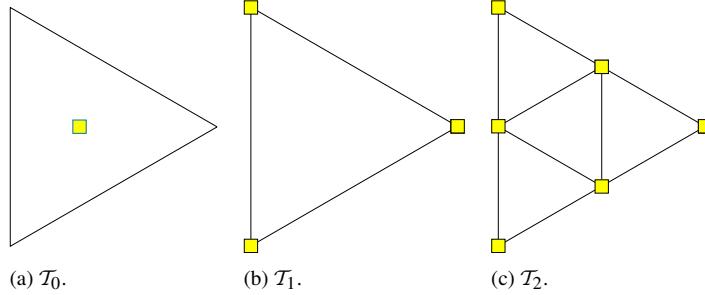


Fig. 14. The domain triangles associated with triangular Bernstein polynomials of degree $k = 0, 1, 2$. The yellow squares are the domain points to which the triangular Bernstein polynomials are associated.

Once an appropriate domain triangle \mathcal{T}_k has been chosen for a given k , we will require that $\{N_l : l = 1, \dots, n_k\}$ satisfy the following conditions, in the spirit of Eq. (3.7),

$$\forall m_1, m_2 \in \mathbb{N} \cup \{0\} : m_1 + m_2 \leq k, \quad \lim_{(u,v) \rightarrow (0,0)} \frac{\partial^{m_1+m_2} N_l}{\partial u^{m_1} \partial v^{m_2}}(u, v) = \frac{\partial^{m_1+m_2} T_l}{\partial u^{m_1} \partial v^{m_2}}(0, 0). \quad (3.17)$$

These relations yield continuity constraints, and the rows of $\tilde{\mathbf{E}}^k$ will span the null-space of these constraints. In the context of splines on triangulations, it is a common technique to exploit the Bernstein polynomial basis in the construction of smooth basis functions across a vertex [45]. In the interest of standardization, we choose the triangles \mathcal{T}_k as equilateral with vertices $(\tau_k, 0)$, $\left(-\frac{\tau_k}{2}, \frac{\sqrt{3}\tau_k}{2}\right)$ and $\left(-\frac{\tau_k}{2}, -\frac{\sqrt{3}\tau_k}{2}\right)$. The size of \mathcal{T}_k , dictated by τ_k , is determined based on the requirement that \mathbf{E}^k be IGA-suitable.

Proposition 3.1. *The functions N_l , $l = 1, \dots, n$, have local support, are linearly independent and form a partition of unity.*

Proof. Suppose that we are asking for C^k continuity at the polar point, $k \geq 0$. Looking at Eq. (3.14), we only need to prove the properties for N_l , $l = 1, \dots, n_k$. The linear independence immediately follows from the linearly independent Hermite data set given by Eq. (3.17), while the locality of the basis functions N_l is a consequence of the special sparse structure of \mathbf{E}^k shown in Eq. (3.14).

Next, we prove the partition of unity property using induction. Referring to Eq. (1.2), it suffices to show that for $i \in \{1, 2, \dots, n^\xi\}$ and $j \in \{1, 2, \dots, k+1\}$,

$$\sum_{l=1}^{n_k} \bar{E}_{l,(ij)}^k = 1. \quad (3.18)$$

• **Basic step:** For $m_1 = m_2 = 0$, using Eq. (3.5),

$$\hat{N}_l(\xi, \eta) = N_l(u, v).$$

Enforcing Eq. (3.17), we require,

$$\lim_{(u,v) \rightarrow (0,0)} N_l(u, v) = T_l(0, 0).$$

Then, using Eq. (3.16),

$$\lim_{(\xi,\eta) \rightarrow (\xi_0,0)} \sum_{i=1}^{n^\xi} \sum_{j=1}^{k+1} \bar{E}_{l,(ij)}^k \hat{B}_{ij}(\xi, \eta) = \sum_{i=1}^{n^\xi} \bar{E}_{l,(i1)}^k \hat{B}_{i1}(\xi_0, 0) = T_l(0, 0),$$

for an arbitrary ξ_0 . Summing over l ,

$$\sum_{l=1}^{n_k} \sum_{i=1}^{n^\xi} \bar{E}_{l,(i1)}^k \hat{B}_{i1}(\xi_0, 0) = \sum_{l=1}^{n_k} T_l(0, 0) = 1,$$

as T_l form a partition of unity. Rearranging the summation order,

$$\sum_{i=1}^{n_k^{\xi}} \hat{B}_{i1}(\xi_0, 0) \left(\sum_{l=1}^{n_k} \bar{E}_{l,(i1)}^k \right) = 1.$$

Therefore, by the partition of unity of \hat{B}_{ij} we must have,

$$\sum_{l=1}^{n_k} \bar{E}_{l,(i1)}^k = 1, \quad (3.19)$$

for $i \in \{1, 2, \dots, n_k^{\xi}\}$.

- **Induction step:** For $0 \leq m_1 + m_2 \leq \kappa < k$, assume that enforcement of Eq. (3.17) entails,

$$\sum_{l=1}^{n_k} \bar{E}_{l,(ij)}^k = 1, \quad (3.20)$$

for arbitrary $i \in \{1, 2, \dots, n_k^{\xi}\}$ and $j \in \{1, 2, \dots, \kappa + 1\}$. Then, for all $m_1 + m_2 = \kappa + 1 \geq 1$, we require the following from Eq. (3.17),

$$\lim_{(u,v) \rightarrow (0,0)} \frac{\partial^{\kappa+1} N_l}{\partial u^{m_1} \partial v^{m_2}}(u, v) = \frac{\partial^{\kappa+1} T_l}{\partial u^{m_1} \partial v^{m_2}}(0, 0).$$

Allowing $(u, v) \rightarrow (0, 0)$ – and, equivalently $(\xi, \eta) \rightarrow (\xi_0, 0)$ – and using Eqs. (3.6) and (3.16), we obtain,

$$\sum_{i=1}^{n_k^{\xi}} \sum_{j=1}^{\kappa+2} \bar{E}_{l,(ij)}^k \frac{\partial^{\kappa+1} \hat{B}_{ij}}{\partial \xi^{m_1} \partial \eta^{m_2}}(\xi_0, 0) = \sum_{i \in \chi_{\kappa+1}} d_i^{m_1, m_2}(\xi_0, 0) \frac{\partial^{i_1+i_2} T_l}{\partial u^{i_1} \partial v^{i_2}}(0, 0),$$

for an arbitrary ξ_0 . Note that, as explained in Section 3.1, the above is assumed to hold for suitable maps F_u , F_v and \hat{B}_{ij} . Summing over l and rearranging the orders of summation,

$$\sum_{i=1}^{n_k^{\xi}} \sum_{j=1}^{\kappa+2} \frac{\partial^{\kappa+1} \hat{B}_{ij}}{\partial \xi^{m_1} \partial \eta^{m_2}}(\xi_0, 0) \left(\sum_{l=1}^{n_k} \bar{E}_{l,(ij)}^k \right) = \sum_{i \in \chi_{\kappa+1}} d_i^{m_1, m_2}(\xi_0, 0) \left(\sum_{l=1}^{n_k} \frac{\partial^{i_1+i_2} T_l}{\partial u^{i_1} \partial v^{i_2}}(0, 0) \right) = 0,$$

where the right-hand side vanishes because the functions T_l form a partition of unity. Splitting the sum over j and using Eq. (3.20),

$$\sum_{i=1}^{n_k^{\xi}} \sum_{j=1}^{\kappa+1} \frac{\partial^{\kappa+1} \hat{B}_{ij}}{\partial \xi^{m_1} \partial \eta^{m_2}}(\xi_0, 0) + \sum_{i=1}^{n_k^{\xi}} \frac{\partial^{\kappa+1} \hat{B}_{i(\kappa+2)}}{\partial \xi^{m_1} \partial \eta^{m_2}}(\xi_0, 0) \left(\sum_{l=1}^{n_k} \bar{E}_{l,(i(\kappa+2))}^k \right) = 0.$$

Since the \hat{B}_{ij} form a partition of unity,

$$\sum_{i=1}^{n_k^{\xi}} \sum_{j=1}^{\kappa+2} \frac{\partial^{\kappa+1} \hat{B}_{ij}}{\partial \xi^{m_1} \partial \eta^{m_2}}(\xi_0, 0) = 0,$$

we can simplify the continuity constraint as,

$$\sum_{i=1}^{n_k^{\xi}} \frac{\partial^{\kappa+1} \hat{B}_{i(\kappa+2)}}{\partial \xi^{m_1} \partial \eta^{m_2}}(\xi_0, 0) \left(\sum_{l=1}^{n_k} \bar{E}_{l,(i(\kappa+2))}^k - 1 \right) = 0.$$

The above needs to hold for all $m_1, m_2 \in \mathbb{N} \cup \{0\}$ such that $m_1 + m_2 = \kappa + 1$. Therefore,

$$\sum_{l=1}^{n_k} \bar{E}_{l,(i(\kappa+2))}^k = 1, \quad (3.21)$$

for $i \in \{1, 2, \dots, n_k^{\xi}\}$.

By combining Eqs. (3.19)–(3.21) we conclude that Eq. (3.18) must hold. \square

While our polar basis functions always satisfy the properties given in [Proposition 3.1](#), this is not necessarily the case for non-negativity. Nevertheless, as shown in the next proposition, we can always guarantee non-negativity by choosing the triangle T_k large enough. We recall here that the multi-degree spline basis functions \hat{B}_{ij} are assumed to be IGA-suitable.

Proposition 3.2. *For τ_k sufficiently large, we are guaranteed an IGA-suitable \mathbf{E}^k .*

Proof. We need to show that enforcement of Eq. [\(3.17\)](#) implies existence of an IGA-suitable \mathbf{E}^k . Looking at Eq. [\(3.14\)](#), we only need to show the existence of a suitable $\bar{\mathbf{E}}^k$. [Proposition 3.1](#) guarantees that $\bar{\mathbf{E}}^k$ satisfies the conditions (i), (ii) and (iv) in [Definition 1.1](#). In the following, we show the last IGA-suitable condition (iii) for τ_k sufficiently large, which ensures the non-negativity of the basis functions N_l .

We first note that,

$$\lim_{(u,v) \rightarrow (0,0)} N_l(u, v) = \sum_{i=1}^{n^\xi} \bar{E}_{l,(i1)}^k \hat{B}_{i1}(\xi_0, 0) = T_l(0, 0),$$

which implies by the partition of unity of \hat{B}_{ij} ,

$$\forall i \in \{1, 2, \dots, n^\xi\}, \quad \bar{E}_{l,(i1)}^k = T_l(0, 0) > 0.$$

Now, for a given k , the m th order derivatives of T_l scale inversely with $(\tau_k)^m$; see [Appendix A](#) for $m = 0, 1, 2$. Then, allowing τ_k to become arbitrarily large, using Eq. [\(3.6\)](#), we can force all derivatives of \hat{N}_l up to order k to become arbitrarily small. In particular, for τ_k sufficiently large and some $\epsilon > 0$, we can enforce the following for all ξ_0 ,

$$\forall m \in \{1, 2, \dots, k\}, \quad \left| \frac{\partial^m \hat{N}_l}{\partial \eta^m} (\xi_0, 0) \right| < \epsilon.$$

Using Eqs. [\(3.16\)](#) and [\(2.10\)](#) and the tensor-product nature of \hat{B}_{ij} ,

$$\forall m \in \{1, 2, \dots, k\}, \quad \left| \sum_{i=1}^{n^\xi} \sum_{j=1}^m C_{m,j}^\eta (\bar{E}_{l,(i(j+1))}^k - \bar{E}_{l,(i1)}^k) \hat{B}_i^\xi(\xi_0) \right| < \epsilon.$$

By [Proposition 2.6](#) there exists a constant A such that,

$$\forall m \in \{1, 2, \dots, k\}, \quad \max_i \left| \sum_{j=1}^m C_{m,j}^\eta (\bar{E}_{l,(i(j+1))}^k - \bar{E}_{l,(i1)}^k) \right| < A\epsilon =: \delta_\epsilon.$$

Therefore, there must exist $\gamma_\delta > 0$ such that,

$$\forall j \in \{1, 2, \dots, k+1\}, \quad |\bar{E}_{l,(ij)}^k - \bar{E}_{l,(i1)}^k| < \gamma_\delta.$$

Since $\bar{E}_{l,(i1)}^k > 0$, by choosing a large enough τ_k we can obtain sufficiently small $\epsilon, \delta_\epsilon$ and γ_δ such that,

$$\forall j \in \{1, 2, \dots, k+1\}, \quad \bar{E}_{l,(ij)}^k \geq 0. \quad \square$$

Remark 3.1. The proof of non-negativity outlined in [Proposition 3.2](#) is constructive in the sense that, from it, we can intuit a sequence of steps to be followed for the construction of an IGA-suitable \mathbf{E}^k . As will become clear from Sections [3.3.2–3.3.4](#), these steps amount to the following. For a given τ_k and $i = 1, \dots, n^\xi$:

- (i) Set the extraction coefficients $\bar{E}_{l,(i1)}^k = T_l(0, 0)$ for all l .
- (ii) Using Eqs. [\(2.10\)](#), [\(3.16\)](#) and [\(3.17\)](#), for $j = 2, \dots, k+1$, compute coefficients $\bar{E}_{l,(i(j+1))}^k$ using the previously computed $\bar{E}_{l,(ij)}^k, \dots, \bar{E}_{l,(i1)}^k$.
- (iii) If all $\bar{E}_{l,(ij)}^k \geq 0$, then stop; else, increase τ_k and repeat the above steps.

For $k = 0$ and $k = 1$, an explicit expression for selecting a suitable τ_k is possible; see Sections 3.3.2 and 3.3.3. However, it is not clear if this is possible for $k \geq 2$ and τ_k is determined iteratively; see Section 3.3.4.

Once we have computed an IGA-suitable \mathbf{E}^k , given n control points $s_l \in \mathbb{R}^d, l = 1, \dots, n$, we can construct a bivariate polar surface \mathbf{s} embedded in \mathbb{R}^d ,

$$\mathbf{s}(u, v) = \sum_{l=1}^n s_l N_l(u, v). \quad (3.22)$$

Remark 3.2. For a fixed surface, the transpose of the extraction operator defines the relationship between control points of the B_{ij} (discontinuous at the polar point) and control points of the smooth N_l ,

$$\sum_{i=1}^{n^\xi} \sum_{j=1}^{n^\eta} \mathbf{r}_{ij} B_{ij}(u, v) = \mathbf{s}(u, v) = \sum_{l=1}^n s_l N_l(u, v) \Leftrightarrow \mathbf{r}_{ij} = \sum_{l=1}^n E_{l,(ij)}^k s_l. \quad (3.23)$$

Remark 3.3. The behavior of \mathbf{s} at the polar point is going to be fully specified by the first n_k control points $s_l, l = 1, \dots, n_k$. These control points can be thought of as forming a control net with the same connectivity as illustrated in Fig. 14. We will equivalently say that they form a control triangle of order k . Fig. 14 shows examples of such control triangles for $k \in \{0, 1, 2\}$.

Adhering to the recipe presented here, we detail explicit constructions for the cases $k \in \{0, 1, 2\}$ in the following. For these cases, the basis functions T_l and their m th derivatives at the polar point are provided in Appendix A.

3.3.2. C^0

For $k = 0$, we have a single ($n_0 = 1$) Bernstein basis function, T_1 , defined on \mathcal{T}_0 (Fig. 14(a)) that drives the value of the polar spline basis N_1 at the polar point. Since $T_1 \equiv 1$, following Eq. (3.17) we require,

$$\lim_{(u,v) \rightarrow (0,0)} N_1(u, v) = 1. \quad (3.24)$$

Pulling back the spline basis to $\hat{\Omega}$ using Eq. (3.5) and using Eq. (3.16), we require,

$$\sum_{i=1}^{n^\xi} \bar{E}_{1,(i1)}^0 \hat{B}_{i1}(\xi_0, 0) = 1,$$

and since the basis functions \hat{B}_{ij} form a partition of unity, the equation is satisfied for all ξ_0 if,

$$\forall i, \bar{E}_{1,(i1)}^0 = 1.$$

Therefore, $\bar{\mathbf{E}}^0$ is obtained as,

$$\bar{\mathbf{E}}^0 = [\underbrace{1, \dots, 1}_{n^\xi \text{ times}}], \quad (3.25)$$

and is trivially seen to be IGA-suitable.

Remark 3.4. Definition 3.1 does not introduce any additional constraints for $k = 0$. Therefore, we set $\mathcal{R}^{\xi\eta}$ and $\mathcal{R}_F^{\xi\eta}$ to be equal for $k = 0$.

Example 3.2 (C^0 Polar Basis). Consider the setup in Example 3.1. Then, making use of Eqs. (3.15) and (3.25), we can construct the polar spline basis function N_1 using the 0th ring of basis functions \hat{B}_{i1} . This basis function has been plotted in Fig. 15.

3.3.3. C^1

For $k = 1$, we have three Bernstein basis functions $\{T_l : l = 1, 2, 3\}$ defined on \mathcal{T}_1 (Fig. 14(b)). Following Eq. (3.17), we require N_l to be such that for $l = 1, 2, 3$,

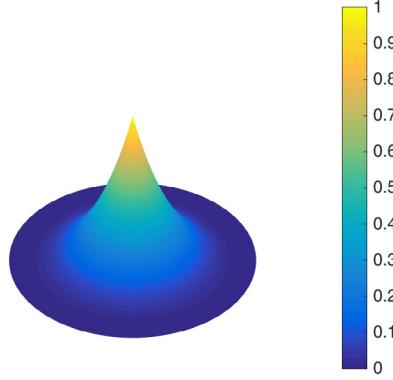


Fig. 15. The requirement of C^0 continuity at $(0, 0)$ yields the single spline basis function $N_l(u, v)$. This basis function is obtained via the extraction operator \bar{E}^0 given in Eq. (3.25) and is the only non-zero basis function at the polar point. The basis shown here corresponds to the setup in Example 3.2.

$$\begin{aligned} \lim_{(u,v) \rightarrow (0,0)} N_l(u, v) &= T_l(0, 0) = \frac{1}{3}, \\ \lim_{(u,v) \rightarrow (0,0)} \left[\frac{\partial N_l}{\partial u}, \frac{\partial N_l}{\partial v} \right] \Big|_{(u,v)} &= \left[\frac{\partial T_l}{\partial u}, \frac{\partial T_l}{\partial v} \right] \Big|_{(0,0)}, \end{aligned} \quad (3.26)$$

where the values and derivatives of the source basis functions are listed in Appendix A. Pulling back the spline basis functions and derivatives to $\hat{\Omega}$ using Eqs. (3.5) and (B.3) for $m_1 + m_2 \leq 1$, we require for all ξ_0 ,

$$\begin{aligned} \lim_{(\xi,\eta) \rightarrow (\xi_0,0)} \hat{N}_l(\xi, \eta) &= \frac{1}{3}, \\ \lim_{(\xi,\eta) \rightarrow (\xi_0,0)} \begin{bmatrix} \frac{\partial \hat{N}_l}{\partial \xi} \\ \frac{\partial \hat{N}_l}{\partial \eta} \end{bmatrix} \Big|_{(\xi,\eta)} &= \lim_{(\xi,\eta) \rightarrow (\xi_0,0)} \begin{bmatrix} \frac{\partial F_u}{\partial \xi} & \frac{\partial F_v}{\partial \xi} \\ \frac{\partial F_u}{\partial \eta} & \frac{\partial F_v}{\partial \eta} \end{bmatrix} \Big|_{(\xi,\eta)} \begin{bmatrix} \frac{\partial T_l}{\partial u} \\ \frac{\partial T_l}{\partial v} \end{bmatrix} \Big|_{(0,0)}. \end{aligned}$$

Since,

$$\hat{N}_l(\xi_0, 0) = \sum_{i=1}^{n^\xi} \bar{E}_{l,(i1)}^1 \hat{B}_{i1}(\xi_0, 0) = \sum_{i=1}^{n^\xi} \bar{E}_{l,(i1)}^1 \hat{B}_i^\xi(\xi_0),$$

and by the partition of unity of the basis functions, we must have

$$\forall i, \quad \bar{E}_{l,(i1)}^1 = \frac{1}{3}.$$

In particular, this implies that $\frac{\partial \hat{N}_l}{\partial \xi}(\xi_0, 0) = 0$ for all ξ_0 . From Eq. (3.12) we also have,

$$\frac{\partial F_u}{\partial \xi} \Big|_{\eta=0} \equiv 0 \equiv \frac{\partial F_v}{\partial \xi} \Big|_{\eta=0},$$

and therefore the only constraint that still needs to be satisfied is,

$$\frac{\partial \hat{N}_l}{\partial \eta}(\xi_0, 0) = \frac{\partial F_u}{\partial \eta}(\xi_0, 0) \frac{\partial T_l}{\partial u}(0, 0) + \frac{\partial F_v}{\partial \eta}(\xi_0, 0) \frac{\partial T_l}{\partial v}(0, 0). \quad (3.27)$$

Remark 3.5. Similarly to the case of $k = 0$ (see Remark 3.4), Definition 3.1 allows us to set $\mathcal{R}^{\xi\eta}$ and $\mathcal{R}_F^{\xi\eta}$ to be equal for $k = 1$.

Using Eq. (2.10),

$$\frac{\partial \hat{N}_l}{\partial \eta}(\xi_0, 0) = C_{1,1}^{\eta} \sum_{i=1}^{n^{\xi}} (\bar{E}_{l,(i2)}^1 - \bar{E}_{l,(i1)}^1) \hat{B}_i^{\xi}(\xi_0).$$

There are similar expansions of the other terms and upon use of Eqs. (3.10) and (3.11), we can rewrite the constraint in Eq. (3.27) as,

$$\begin{aligned} C_{1,1}^{\eta} \sum_{i=1}^{n^{\xi}} (\bar{E}_{l,(i2)}^1 - \bar{E}_{l,(i1)}^1) \hat{B}_i^{\xi}(\xi_0) &= C_{1,1}^{\eta} (\rho_2 - \rho_1) \frac{\partial T_l}{\partial u}(0, 0) \sum_{i=1}^{n^{\xi}} \cos(\theta_i) \hat{B}_i^{\xi}(\xi_0) \\ &\quad + C_{1,1}^{\eta} (\rho_2 - \rho_1) \frac{\partial T_l}{\partial v}(0, 0) \sum_{i=1}^{n^{\xi}} \sin(\theta_i) \hat{B}_i^{\xi}(\xi_0). \end{aligned}$$

Then, the following equality must hold for $i = 1, 2, \dots, n^{\xi}$ (recall that $\rho_1 = 0$),

$$(\bar{E}_{l,(i2)}^1 - \bar{E}_{l,(i1)}^1) - \rho_2 \cos(\theta_i) \frac{\partial T_l}{\partial u}(0, 0) - \rho_2 \frac{\partial T_l}{\partial v}(0, 0) \sin(\theta_i) = 0,$$

or,

$$\bar{E}_{l,(i2)}^1 = \frac{1}{\sqrt{3}} \left(\rho_2 \cos(\theta_i) \frac{2i_1 - i_2 - i_3}{\sqrt{3}\tau_1} + \rho_2 \sin(\theta_i) \frac{i_2 - i_3}{\tau_1} + \frac{1}{\sqrt{3}} \right).$$

Then, since Definition 3.2 implies that $l = 1, 2, 3$ correspond to $(i_1, i_2, i_3) = (1, 0, 0), (0, 1, 0), (0, 0, 1)$, respectively, we obtain,

$$\begin{bmatrix} \bar{E}_{1,(i2)}^1 \\ \bar{E}_{2,(i2)}^1 \\ \bar{E}_{3,(i2)}^1 \end{bmatrix} = \begin{bmatrix} \frac{2}{3\tau_1} & 0 & \frac{1}{3} \\ -\frac{1}{3\tau_1} & \frac{1}{\sqrt{3}\tau_1} & \frac{1}{3} \\ -\frac{1}{3\tau_1} & -\frac{1}{\sqrt{3}\tau_1} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} \rho_2 \cos(\theta_i) \\ \rho_2 \sin(\theta_i) \\ 1 \end{bmatrix}.$$

The above relation says that $(\bar{E}_{1,(i2)}^1, \bar{E}_{2,(i2)}^1, \bar{E}_{3,(i2)}^1)$ are simply the barycentric coordinates of the control point $(\rho_2 \cos(\theta_i), \rho_2 \sin(\theta_i))$ with respect to T_1 . Therefore, to guarantee their non-negativity, it is sufficient to choose a large enough T_1 such that it encloses the circle centered at $(0, 0)$ with a radius of ρ_2 , i.e., we require,

$$\tau_1 \geq 2\rho_2.$$

Remark 3.6. As long as T_1 is chosen to be the smallest triangle enclosing the 1st ring of control points for a given polar configuration, we are guaranteed non-negative extraction coefficients. It is only in the interest of standardization that we have chosen to fix T_1 as an equilateral triangle with a fixed pattern of vertices, and the condition $\tau_1 \geq 2\rho_2$ is a result of this choice.

Choosing $\tau_1 = 2\rho_2$ to obtain the most well conditioned basis, the final form of the $(n^{\xi} + i)$ th column of \bar{E}^1 is,

$$\begin{bmatrix} \bar{E}_{1,(i2)}^1 \\ \bar{E}_{2,(i2)}^1 \\ \bar{E}_{3,(i2)}^1 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & 0 & \frac{1}{3} \\ -\frac{1}{6} & \frac{1}{2\sqrt{3}} & \frac{1}{3} \\ -\frac{1}{6} & -\frac{1}{2\sqrt{3}} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \\ 1 \end{bmatrix}.$$

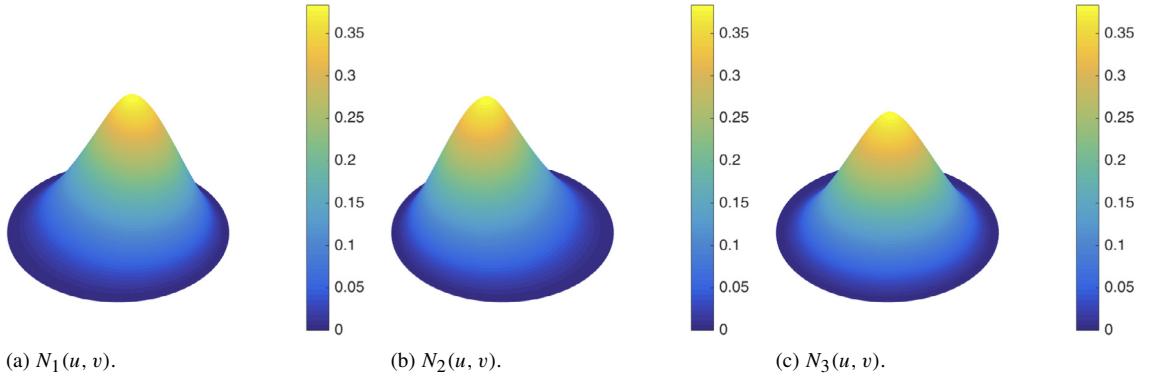


Fig. 16. The three polar spline basis functions that have non-zero values and first derivatives at $(0, 0)$. Any linear combination of these basis functions will be C^1 across the polar point. These basis functions correspond to the data used in Example 3.3.

\bar{E}^1 is assembled as,

$$\bar{E}^1 = \begin{bmatrix} \frac{1}{3} & \dots & \frac{1}{3} & \bar{E}_{1,(12)}^1 & \dots & \bar{E}_{1,(i2)}^1 & \dots & \bar{E}_{1,(n^\xi 2)}^1 \\ \frac{1}{3} & \dots & \frac{1}{3} & \bar{E}_{2,(12)}^1 & \dots & \bar{E}_{2,(i2)}^1 & \dots & \bar{E}_{2,(n^\xi 2)}^1 \\ \frac{1}{3} & \dots & \frac{1}{3} & \bar{E}_{3,(12)}^1 & \dots & \bar{E}_{3,(i2)}^1 & \dots & \bar{E}_{3,(n^\xi 2)}^1 \end{bmatrix}. \quad (3.28)$$

It can be directly verified that the column sum of \bar{E}^1 is equal to one, thus confirming that E^1 is IGA-suitable.

Remark 3.7. From Remark 3.2, the transpose of the extraction operator in Eq. (3.28) essentially computes the control points of the 0th and the 1st ring of basis functions B_{ij} as convex combinations of s_1, s_2 and s_3 . In other words, it forces the 0th and the 1st ring of control points to be coplanar. The plane in which they lie is the one passing through s_1, s_2 and s_3 . In fact, as is shown later in Proposition 3.3, this plane is the tangent-plane of s at the point $s(0, 0)$.

Example 3.3 (C^1 Polar Basis). Consider once again the setup used in Example 3.1. Then, using Eq. (3.28), the three polar basis functions that have non-zero values and first derivatives at the polar point are shown in Fig. 16. Any surface built using these polar basis functions will be C^1 across the polar point.

Example 3.4 (C^1 Hemisphere). Hemispherical caps are often used in geometric design of everyday objects, and in this example we provide a C^1 description of a hemisphere using the polar basis functions built in Example 3.3. This hemisphere is made up of quadratic and cubic splines in the angular and radial directions, respectively.

Recall that the tensor-product spline space we are using is the one built in Example 3.1; the univariate basis functions used for construction of the tensor-product space are shown in Fig. 13(a). Using Eq. (3.28), we can build the polar basis functions on the polar parametric domain shown in Fig. 13(b); the 3 polar basis functions enforcing C^1 continuity at the polar point are shown in Fig. 16. The total dimension of this space is going to be $n = (n^\xi n^\eta - n_o + n_k) = 11$. Then, assigning the following control points in \mathbb{R}^3 ,

$$\begin{aligned} s_1 &= \left(\frac{2\sqrt{2}}{1+\sqrt{2}}, 0, \frac{1}{\sqrt{2}} \right), \quad s_2 = \left(-\frac{\sqrt{2}}{1+\sqrt{2}}, \frac{\sqrt{6}}{1+\sqrt{2}}, \frac{1}{\sqrt{2}} \right), \\ s_3 &= \left(-\frac{\sqrt{2}}{1+\sqrt{2}}, -\frac{\sqrt{6}}{1+\sqrt{2}}, \frac{1}{\sqrt{2}} \right), \quad s_4 = \left(1, 0, \frac{1}{1+\sqrt{2}} \right), \\ s_5 &= \left(0, 1, \frac{1}{1+\sqrt{2}} \right), \quad s_6 = \left(-1, 0, \frac{1}{1+\sqrt{2}} \right), \quad s_7 = \left(0, -1, \frac{1}{1+\sqrt{2}} \right), \\ s_8 &= (1, 0, 0), \quad s_9 = (0, 1, 0), \quad s_{10} = (-1, 0, 0), \quad s_{11} = (0, -1, 0), \end{aligned}$$

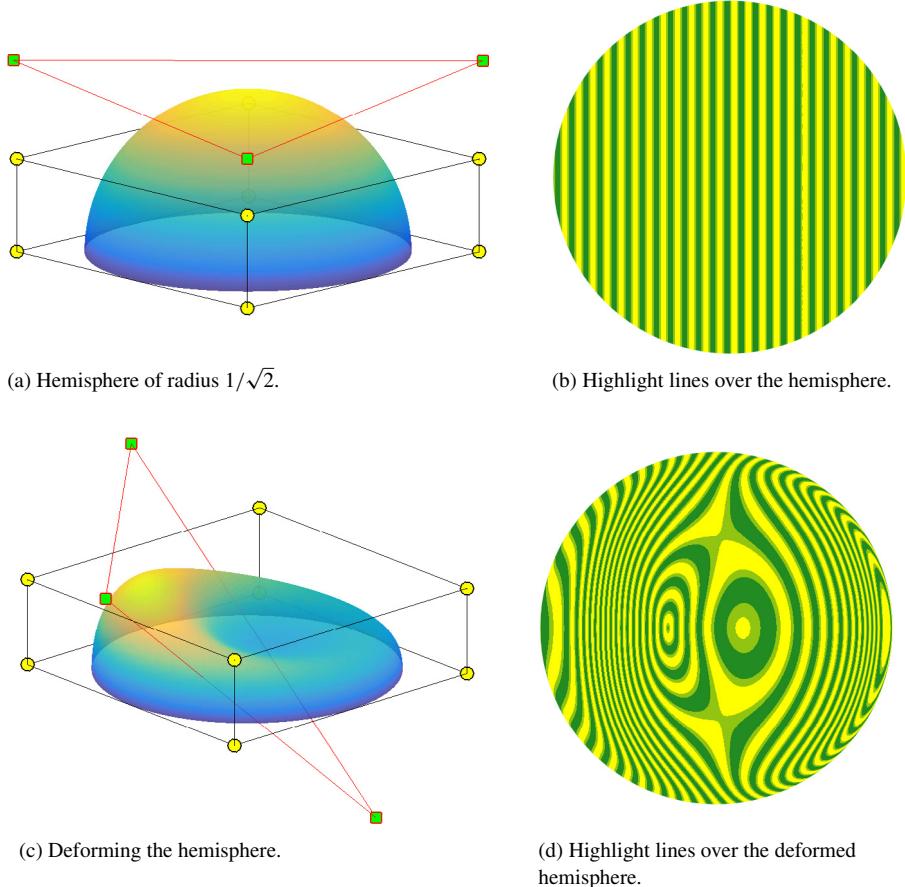


Fig. 17. The representation of a hemisphere using the C^1 polar basis functions computed in Example 3.3. As can be seen, movement of an arbitrary control point in this representation deforms the surface smoothly, making this a convenient representation for a designer to work with.

to the 11 C^1 basis functions N_l , we obtain a C^1 representation of the hemisphere as shown in Fig. 17(a). This hemisphere has a radius $\frac{1}{\sqrt{2}}$. Shown in Fig. 17(b) is the top view of highlight lines for this surface.

In order to show the advantage that such a C^1 representation has over conventional approaches, let us modify the control point s_1 to $\left(\frac{\sqrt{2}}{1+\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}\right)$. This deforms the surface as shown in Fig. 17(c), and the resulting geometry is still C^1 . Shown in Fig. 17(d) are the highlight lines for this surface and they are clearly continuous. This is obviously a convenient and low degree representation of the hemisphere for a designer to work with.

Remark 3.8. The choice of one cubic element in radial direction in Example 3.4 is not necessary for the construction of a hemisphere, as a quadratic element would suffice. However, this presentation highlights better the locality of deformations. If the hemisphere is joined in a C^1 way with an outside surface, then moving the triangular control net would only modify the hemisphere and not the outside surface in case of one cubic element (or two quadratic elements) in radial direction.

In the case of C^1 polar splines, we can introduce an additional feature to aid the process of design – the so-called *weight point* – but first, let us prove the following elementary result.

Proposition 3.3. Let $s \subset \mathbb{R}^d$ be a C^1 polar spline surface defined as in Eq. (3.22), and set $p_0 := \frac{1}{3}(s_1 + s_2 + s_3)$. Then,

- the point p_0 is a point lying on the surface s corresponding to the evaluation at the polar point, i.e., $s(0, 0) = p_0$;
- the triangle formed by $\{s_1, s_2, s_3\}$ is tangent to the surface s at the point p_0 .

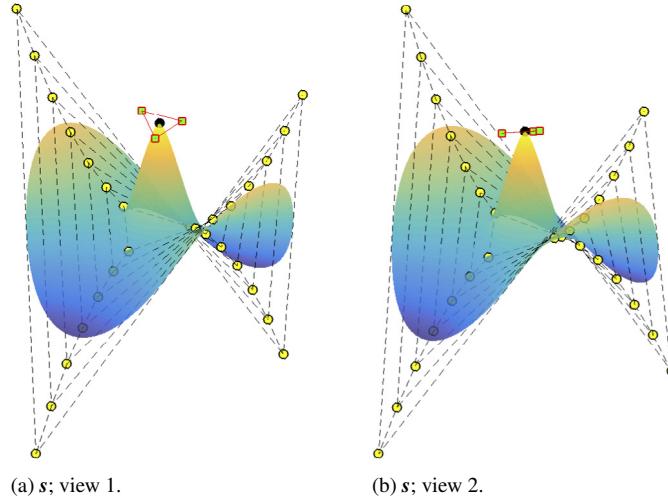


Fig. 18. A C^1 polar surface with its associated control points. The point p_0 is plotted as a black disk; the view in Fig. 18(b) clearly shows that this point lies on s , and the control triangle is tangent to s at this point.

Proof. At the polar point, the only non-zero polar basis functions are N_l , $l = 1, 2, 3$, and Eq. (3.15) reduces to,

$$N_l(0, 0) = \sum_{i=1}^{n^\xi} \bar{E}_{l,(i1)}^1 B_{i1}(0, 0),$$

as the j th ring of basis functions B_{ij} are all zero at $(0, 0)$ for $j > 0$. Then, substituting the extraction coefficients from Eq. (3.28), it is easy to see that the surface takes the following value at $(0, 0)$,

$$s(0, 0) = \sum_{l=1}^3 s_l N_l(0, 0) = \frac{1}{3} \sum_{l=1}^3 s_l \sum_{i=1}^{n^\xi} B_{i1}(0, 0).$$

Since $\sum_{i=1}^{n^\xi} B_{i1}(0, 0) = 1$ by partition of unity, we see that the surface evaluates to p_0 at $(0, 0)$.

Denote the control point associated to B_{ij} with r_{ij} , see Remark 3.2 and let $\mathbf{r}(u, v)$ be the equivalent surface description. From Remark 3.7, the triangle \mathcal{T} formed by $\{s_1, s_2, s_3\}$ forces the 0th and 1st ring of control points $\{r_{ij} : i = 1, \dots, n^\xi; j = 1, 2\}$ to be coplanar, with \mathcal{T} lying in the plane passing through them. In particular, the 0th ring control points coincide, and the lines connecting r_{i2} to r_{i1} lie in the plane of \mathcal{T} for all i . Moreover, by Eq. (2.10),

$$\begin{aligned} \left. \frac{\partial \hat{s}}{\partial \xi} \right|_{\eta=0} &= \left. \frac{\partial \hat{\mathbf{r}}}{\partial \xi} \right|_{\eta=0} \equiv 0, \\ \left. \frac{\partial \hat{s}}{\partial \eta} \right|_{\eta=0} &= \left. \frac{\partial \hat{\mathbf{r}}}{\partial \eta} \right|_{\eta=0} = C_{1,1}^\eta \sum_{i=1}^{n^\xi} (\mathbf{r}_{i2} - \mathbf{r}_{i1}) \hat{B}_i^\xi. \end{aligned}$$

Using Eq. (B.3) for $m_1 + m_2 = 1$, we conclude that $\frac{\partial s}{\partial u}(0, 0)$ and $\frac{\partial s}{\partial v}(0, 0)$ are parallel to the plane of \mathcal{T} . At the same time, $p_0 = s(0, 0)$ is a surface point on this plane (it is a convex combination of the vertices of \mathcal{T}). Therefore, the plane passing through \mathcal{T} is in fact the tangent-plane to s at the point p_0 . In other words, the control triangle is tangent to the surface at p_0 . \square

Example 3.5 (Surface Point). A C^1 polar spline surface is shown in Fig. 18 along with its associated control net. The control triangle assigned to the basis functions N_1 , N_2 and N_3 has been plotted in red at the middle. The point plotted with a black disk is p_0 as defined in Proposition 3.3. As is clear from the two views of the surface, the point lies on s and the control triangle is tangent to the surface at this point.

The statement of [Proposition 3.3](#) begs the following question: *what if, keeping the control net of s fixed, a designer could visually manipulate the point p_0 in an intuitive fashion?* This leads to the concept of a weight point, (p, ς) , in the spirit of [46].

Definition 3.3 (Weight Point). Let us be given a C^1 polar surface s and the associated control points s_l . Then, a weight point, (p, ς) , is simply an ordered pair where the first element, p , is a point lying inside the control triangle formed by $\{s_1, s_2, s_3\}$, and the second element, ς , is a positive real number.

Proposition 3.4. Let us be given a C^1 polar surface s , the associated control points s_l , and a weight point (p, ς) . The barycentric coordinates of p are given by $(\lambda_1, \lambda_2, \lambda_3)$ with respect to the control triangle formed by $\{s_1, s_2, s_3\}$. Then, the surface \tilde{s} defined as,

$$\tilde{s}(u, v) = \sum_{l=1}^n \frac{s_l w_l N_l(u, v)}{\sum_{i=1}^n w_i N_i(u, v)},$$

is such that $\tilde{s}(0, 0) = p$ if,

$$w_l := \begin{cases} 3\varsigma\lambda_l, & l = 1, 2, 3, \\ 1, & l = 4, \dots, n. \end{cases} \quad (3.29)$$

Proof. Switching to homogeneous coordinates in \mathbb{R}^{d+1} , we set the weighted control points equal to $s_l^w := (w_l s_l, w_l)$. Then, using [Proposition 3.3](#), we know that,

$$\tilde{s}^w(0, 0) = \tilde{p}_0^w = \frac{1}{3} \sum_{l=1}^3 s_l^w.$$

However, from Eq. (3.29),

$$\frac{1}{3} \sum_{l=1}^3 s_l^w = \frac{1}{3} \times 3\varsigma \sum_{l=1}^3 (\lambda_l s_l, \lambda_l) = \varsigma (p, 1).$$

Then, projecting back to \mathbb{R}^d , we get the required result. \square

Remark 3.9. In [Proposition 3.4](#), we make no assumptions about the basis functions N_l being rational or non-rational; the relation holds regardless.

The weight point essentially modifies how the control triangle attracts the surface s towards it. The point p controls the relative influences of s_1, s_2 and s_3 , while the scaling factor ς controls the strength with which the control triangle as a whole pulls the surface towards it. The next example helps us clarify these effects.

Example 3.6 (Weight Point). [Fig. 19](#) illustrates how the weight point (p, ς) could be employed as an intuitive tool for surface manipulation. [Fig. 19\(a\)](#) shows the original surface. In order to emphasize the effect of the control triangle and the weight point, all other control points have been set to $(0, 0, 0)$, and only the control triangle has been plotted. Starting from this configuration, the designer could manipulate the surface in the following ways while keeping the control net fixed:

- By moving p , the surface is pulled along the control triangle ([Fig. 19\(b\)](#)).
- By increasing/decreasing ς , the control triangle attracts the surface with more/less strength ([Figs. 19\(c\) and 19\(d\)](#)). A heavier attraction towards the triangle leads to a flattening of the surface in this case, while a weaker attraction leads to a sharpening.

As can be clearly seen from [Fig. 19](#), the above is a highly intuitive surface manipulation tool in the vicinity of the polar point.

Remark 3.10. The weight point could be a very convenient design tool to visually interact with the geometry: set the point of tangency and, subsequently, by choosing a single positive scalar (using a slider GUI or representing the length of a normal vector, for instance) modify the attraction of the surface to the control net.

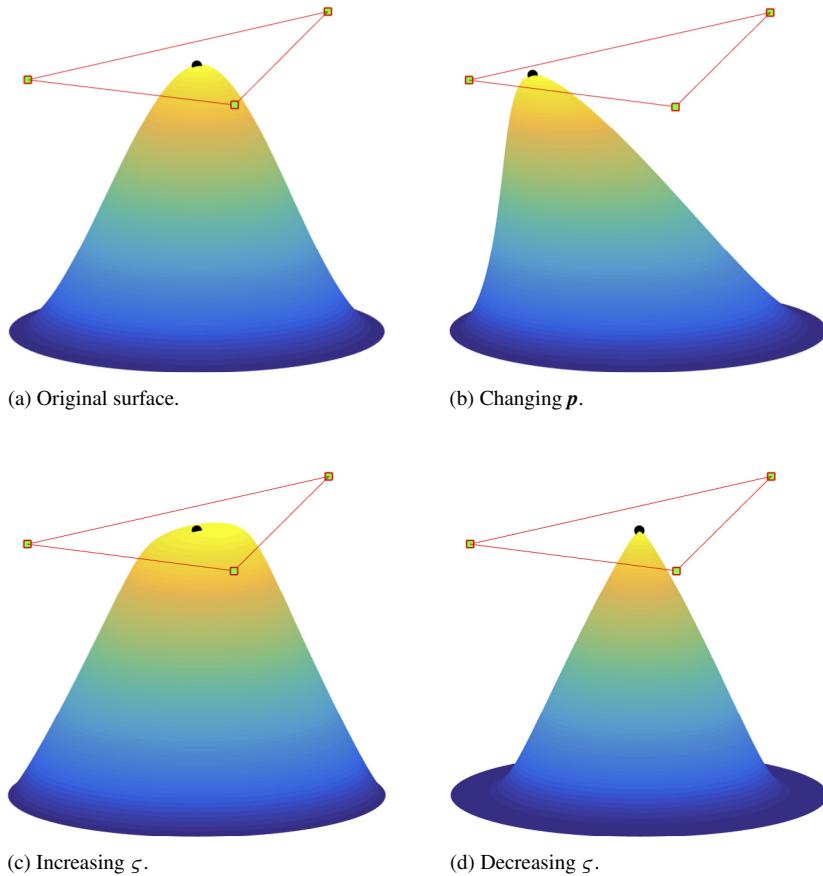


Fig. 19. The weight point (p , ς) can be used for surface manipulation in a visual and highly intuitive fashion. Moving p pulls the surface along the control triangle (top-right figure) while increasing or decreasing ς (bottom row figures) increases and decreases the pull of the control triangle on s , respectively.

3.3.4. C^2

As was stated towards the end of Section 3.1, for all $k \geq 2$, we assume that the basis functions are non-rational in ξ . Therefore, we will perform the construction of C^2 polar splines here with this assumption in mind.

For $k = 2$, we have six Bernstein basis functions $\{T_l : l = 1, \dots, 6\}$ defined on \mathcal{T}_2 (see Fig. 14(c)). Following Eq. (3.17), we require the polar basis functions N_l to be such that for $l = 1, \dots, 6$,

$$\begin{aligned} N_l(0,0) &= T_l(0,0) = \binom{2}{i_1 i_2 i_3} \frac{1}{9}, \\ \lim_{(u,v) \rightarrow (0,0)} \left[\frac{\partial N_l}{\partial u}, \frac{\partial N_l}{\partial v} \right] &= \left[\frac{\partial T_l}{\partial u}, \frac{\partial T_l}{\partial v} \right] \Big|_{(0,0)}, \\ \lim_{(u,v) \rightarrow (0,0)} \left[\frac{\partial^2 N_l}{\partial u^2}, \frac{\partial^2 N_l}{\partial u \partial v}, \frac{\partial^2 N_l}{\partial v^2} \right] &= \left[\frac{\partial^2 T_l}{\partial u^2}, \frac{\partial^2 T_l}{\partial u \partial v}, \frac{\partial^2 T_l}{\partial v^2} \right] \Big|_{(0,0)}, \end{aligned}$$

where the derivatives of the source basis functions can be found in [Appendix A](#). We can pull back the spline basis functions to $\hat{\Omega}$ using Eqs. (3.5) and (B.3), to require the following for $m_1 + m_2 \leq 1$ when $(u, v) \rightarrow (0, 0)$,

$$\hat{N}_l(\xi, \eta) = \begin{pmatrix} 2 \\ i_1 & i_2 & i_3 \end{pmatrix} \frac{1}{9}, \quad \lim_{(\xi, \eta) \rightarrow (\xi_0, 0)} \begin{bmatrix} \frac{\partial \hat{N}_l}{\partial \xi} \\ \frac{\partial \hat{N}_l}{\partial \eta} \end{bmatrix} = \lim_{(\xi, \eta) \rightarrow (\xi_0, 0)} \begin{bmatrix} \frac{\partial F_u}{\partial \xi} & \frac{\partial F_v}{\partial \xi} \\ \frac{\partial F_u}{\partial \eta} & \frac{\partial F_v}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial T_l}{\partial u} \\ \frac{\partial T_l}{\partial v} \end{bmatrix} \Big|_{(0,0)},$$

and for $m_1 + m_2 = 2$,

$$\begin{aligned} \lim_{(\xi, \eta) \rightarrow (\xi_0, 0)} \begin{bmatrix} \frac{\partial^2 \hat{N}_l}{\partial \xi^2} \\ \frac{\partial^2 \hat{N}_l}{\partial \xi \partial \eta} \\ \frac{\partial^2 \hat{N}_l}{\partial \eta^2} \end{bmatrix} &= \lim_{(\xi, \eta) \rightarrow (\xi_0, 0)} \begin{bmatrix} \left(\frac{\partial F_u}{\partial \xi} \right)^2 & 2 \frac{\partial F_u}{\partial \xi} \frac{\partial F_v}{\partial \xi} & \left(\frac{\partial F_v}{\partial \xi} \right)^2 \\ \frac{\partial F_u}{\partial \xi} \frac{\partial F_u}{\partial \eta} & \frac{\partial F_u}{\partial \eta} \frac{\partial F_v}{\partial \xi} + \frac{\partial F_u}{\partial \xi} \frac{\partial F_v}{\partial \eta} & \frac{\partial F_v}{\partial \xi} \frac{\partial F_v}{\partial \eta} \\ \left(\frac{\partial F_u}{\partial \eta} \right)^2 & 2 \frac{\partial F_u}{\partial \eta} \frac{\partial F_v}{\partial \eta} & \left(\frac{\partial F_v}{\partial \eta} \right)^2 \end{bmatrix} \begin{bmatrix} \frac{\partial^2 T_l}{\partial u^2} \\ \frac{\partial^2 T_l}{\partial u \partial v} \\ \frac{\partial^2 T_l}{\partial v^2} \end{bmatrix} \Big|_{(0,0)} \\ &+ \lim_{(\xi, \eta) \rightarrow (\xi_0, 0)} \begin{bmatrix} \frac{\partial^2 F_u}{\partial \xi^2} & \frac{\partial^2 F_v}{\partial \xi^2} \\ \frac{\partial^2 F_u}{\partial \xi \partial \eta} & \frac{\partial^2 F_v}{\partial \xi \partial \eta} \\ \frac{\partial^2 F_u}{\partial \eta^2} & \frac{\partial^2 F_v}{\partial \eta^2} \end{bmatrix} \begin{bmatrix} \frac{\partial T_l}{\partial u} \\ \frac{\partial T_l}{\partial v} \end{bmatrix} \Big|_{(0,0)}. \end{aligned} \quad (3.30)$$

Remark 3.11. In Eq. (3.30), on the right-hand side we have products of spline functions. This is a setting where Definition 3.1 introduces additional constraints: \mathcal{R}_F^ξ must be such that the product of two splines from \mathcal{R}_F^ξ is in it.

Let us recall the notation introduced at the end of Section 3.1: basis functions \hat{B}_{ij} span $\mathcal{R}^{\xi\eta}$, while the basis functions \tilde{B}_{ij} span $\mathcal{R}_F^{\xi\eta}$. Hereafter, we will adopt the following convention. Assume that we are given a spline $y \in \mathcal{R}_F^{\xi\eta}$. Therefore, $y(\xi, \eta) = \sum_{i=1}^{\tilde{n}^\xi} \sum_{j=1}^{\tilde{n}^\eta} y_{ij} \tilde{B}_{ij}(\xi, \eta)$. Then, since $\mathcal{R}_F^{\xi\eta} \subset \mathcal{R}^{\xi\eta}$, we will denote by \check{y}_{ij} the coefficients such that,

$$y(\xi, \eta) = \sum_{i=1}^{\tilde{n}^\xi} \sum_{j=1}^{\tilde{n}^\eta} \check{y}_{ij} \hat{B}_{ij}(\xi, \eta).$$

Taking an approach similar to the one in Section 3.3.2, imposition of C^0 continuity yields the following for $i = 1, 2, \dots, \tilde{n}^\xi$,

$$\bar{E}_{l,(i1)}^2 = \binom{2}{i_1 \ i_2 \ i_3} \frac{1}{9}. \quad (3.31)$$

Next, let u_{ij} and v_{ij} be coefficients such that,

$$(F_u, F_v) = \sum_{i=1}^{\tilde{n}^\xi} \sum_{j=1}^{\tilde{n}^\eta} \rho_j (\cos(\theta_i), \sin(\theta_i)) \tilde{B}_{ij} =: \sum_{i=1}^{\tilde{n}^\xi} \sum_{j=1}^{\tilde{n}^\eta} (u_{ij}, v_{ij}) \tilde{B}_{ij}.$$

Then, using Remark 3.11 and an approach similar to the one in Section 3.3.3, we can resolve the C^1 continuity constraints by setting,

$$\bar{E}_{l,(i2)}^2 = \binom{2}{i_1 \ i_2 \ i_3} \frac{1}{3\sqrt{3}} \left(\check{u}_{i2} \frac{2i_1 - i_2 - i_3}{\sqrt{3}\tau_2} + \check{v}_{i2} \frac{i_2 - i_3}{\tau_2} + \frac{1}{\sqrt{3}} \right), \quad (3.32)$$

where we have used the fact that $u_{i1} = v_{i1} = 0$ for all i . This leaves $\bar{E}_{l,(i3)}^2$, $i = 1, \dots, \tilde{n}^\xi$, as the only remaining coefficients that need to be computed. However, before proceeding let us make the following crucial observations:

- Once the coefficients $\bar{E}_{l,(i1)}^2$ have been set as in Eq. (3.31), the first row of the system of constraints in Eq. (3.30) is automatically satisfied as all the terms differentiated with respect to ξ are identically zero.
- Once a suitable τ_2 has been chosen and the coefficients $\bar{E}_{l,(i2)}^2$ have been set as in Eq. (3.32), it can be verified that the second row in Eq. (3.30) is also automatically satisfied. Indeed, all first derivatives of F_u and F_v with respect

to ξ evaluate to zero at $\eta = 0$, and using Eqs. (3.16) and (2.10), we can rewrite this mixed-derivative constraint as,

$$C_{1,1}^{\eta} \sum_{i=1}^{n^{\xi}} \left(\bar{E}_{l,(i2)}^2 - \bar{E}_{l,(i1)}^2 \right) \frac{d\hat{B}_i^{\xi}}{d\xi}(\xi_0) = C_{1,1}^{\eta} \frac{\partial T_l}{\partial u}(0,0) \sum_{i=1}^{n^{\xi}} \check{u}_{i2} \frac{d\hat{B}_i^{\xi}}{d\xi}(\xi_0) + C_{1,1}^{\eta} \frac{\partial T_l}{\partial v}(0,0) \sum_{i=1}^{n^{\xi}} \check{v}_{i2} \frac{d\hat{B}_i^{\xi}}{d\xi}(\xi_0).$$

However, because of enforcement of C^1 continuity, we already know that for $i \in \{1, 2, \dots, n^{\xi}\}$,

$$\left(\bar{E}_{l,(i2)}^2 - \bar{E}_{l,(i1)}^2 \right) - \check{u}_{i2} \frac{\partial T_l}{\partial u}(0,0) - \check{v}_{i2} \frac{\partial T_l}{\partial v}(0,0) = 0.$$

Therefore, the constraint is automatically satisfied.

The above two observations imply that the only new constraint that needs to be satisfied for C^2 continuity at the polar point is the last row in Eq. (3.30). Note that on the right-hand side of this row, some of the terms are $\left(\frac{\partial F_u}{\partial \eta}\right)^2$, $\left(\frac{\partial F_v}{\partial \eta}\right)^2$ and $\frac{\partial F_u}{\partial \eta} \frac{\partial F_v}{\partial \eta}$, each evaluated at $\eta = 0$ – these are nothing but the products of splines in \mathcal{R}_F^{ξ} . On the left-hand side, we have the second derivative of \hat{N}_l with respect to η , also evaluated at $\eta = 0$ – this is simply a spline in \mathcal{R}^{ξ} . Therefore, in order for this constraint to be satisfied, \mathcal{R}^{ξ} must contain products of splines in \mathcal{R}_F^{ξ} , as was mentioned in Remark 3.11.

Therefore, assuming that the spaces $\mathcal{R}^{\xi\eta}$ and $\mathcal{R}_F^{\xi\eta}$ have been chosen such that the exposition above is valid, we can eliminate the last row in Eq. (3.30) by solving the following equation for coefficients $\bar{E}_{l,(i3)}^2$,

$$\begin{aligned} & C_{2,2}^{\eta} \left(\bar{E}_{l,(i3)}^2 - \bar{E}_{l,(i1)}^2 \right) + C_{2,1}^{\eta} \left(\bar{E}_{l,(i2)}^2 - \bar{E}_{l,(i1)}^2 \right) \\ &= \check{\beta}_i^{(1)} \frac{\partial^2 T_l}{\partial u^2}(0,0) + 2\check{\beta}_i^{(2)} \frac{\partial^2 T_l}{\partial u \partial v}(0,0) + \check{\beta}_i^{(3)} \frac{\partial^2 T_l}{\partial v^2}(0,0) + \check{\alpha}_i^{(1)} \frac{\partial T_l}{\partial u}(0,0) + \check{\alpha}_i^{(2)} \frac{\partial T_l}{\partial v}(0,0), \end{aligned} \quad (3.33)$$

where we have performed the following relabeling,

$$\beta^{(1)} \equiv \left(\frac{\partial F_u}{\partial \eta} \right)^2 \Big|_{\eta=0}, \quad \beta^{(2)} \equiv \frac{\partial F_u}{\partial \eta} \frac{\partial F_v}{\partial \eta} \Big|_{\eta=0}, \quad \beta^{(3)} \equiv \left(\frac{\partial F_v}{\partial \eta} \right)^2 \Big|_{\eta=0}, \quad \alpha^{(1)} \equiv \frac{\partial^2 F_u}{\partial \eta^2} \Big|_{\eta=0}, \quad \alpha^{(2)} \equiv \frac{\partial^2 F_v}{\partial \eta^2} \Big|_{\eta=0}.$$

Using Eqs. (3.31)–(3.33) we can compute the elements of \bar{E}^2 . However, we still need to choose an appropriate τ_2 such that \bar{E}^2 is IGA-suitable. While the C^1 case allowed a straightforward geometric criterion for choosing τ_1 in Section 3.3.3, it is unfortunately not so easy to formulate a similar criterion for τ_2 . Instead, we present the following practical implementation for choosing an appropriate τ_2 and computing \bar{E}^2 . Assume for the following that we are interested in constructing C^2 polar basis functions using $\mathcal{R}^{\xi\eta}$, where the elements of $\mathcal{R}^{\xi\eta}$ are of multi-degree $\tilde{\mathbf{p}} := (p^{(1)}, \dots, p^{(n_s)})$ in the ξ (or angular) direction. We will require $p^{(i)} \geq 6$. The spline configuration in η is not affected by the C^2 constraints at the polar point.

1. Create $\mathcal{R}_F^{\xi\eta} = \mathcal{R}_F^{\xi} \otimes \mathcal{R}^{\eta}$, where $\mathcal{R}_F^{\xi} \subset \mathcal{R}^{\xi}$ contains splines that have multi-degree in ξ equal to,

$$\tilde{\mathbf{p}} := (\tilde{p}^{(1)}, \dots, \tilde{p}^{(n_s)}) := \left(\left\lfloor \frac{p^{(1)}}{2} \right\rfloor, \dots, \left\lfloor \frac{p^{(n_s)}}{2} \right\rfloor \right),$$
and are in the same continuity class as elements of \mathcal{R}^{ξ} everywhere.
2. Create \mathbf{F} and the polar parametric domain using Eqs. (3.10) and (3.11).
3. Compute the coefficients \check{u}_{ij} , \check{v}_{ij} , $\check{\beta}_i^{(1)}$, $\check{\beta}_i^{(2)}$, $\check{\beta}_i^{(3)}$, $\check{\alpha}_i^{(1)}$ and $\check{\alpha}_i^{(2)}$ from Eqs. (3.32) and (3.33).
4. Set $\tau_2 = 4\rho_2$.
5. Using Appendix A, compute the source basis functions and their derivatives on \mathcal{T}_2 for the chosen τ_2 .
6. For $l = 1, 2, \dots, 6$ compute the extraction coefficients $\bar{E}_{l,(ij)}^2$ using Eqs. (3.31)–(3.33).
7. If $\bar{E}_{l,(ij)}^2 \geq 0$ for all i, j , then the computation of \bar{E}^2 is complete. Else, set $\tau_2 \leftarrow \tau_2 + 2\rho_2$ and go back to Step 5.

Remark 3.12. The increase of τ_2 to $\tau_2 + 2\rho_2$ in Step 7 is chosen so that the control triangle will enclose one more ring of control points. However, other choices can be made as well.

Remark 3.13. In the special case when $\mathcal{R}_F^{\xi\eta}$ contains splines of uniform bi-degree $(6, 3)$, the polar spline basis functions span the same space as the “shape” basis from [3]. However, we believe that the B-spline-like character of the former makes them more suitable for both geometric modeling and engineering analysis.

Example 3.7 (C^2 Polar Basis). We want to construct a C^2 polar surface with degree 6 splines in the ξ direction and degree 4 splines in the η direction. Note that degree 6 in ξ is the smallest degree that can be used to obtain C^2 smoothness, and this in turn necessitates that the degree in ξ for $\mathcal{R}_F^{\xi\eta}$ be equal to 3. Let us construct $\mathcal{R}_F^{\xi\eta}$ using,

$$\tilde{U} = (\tilde{U}), \quad \kappa^\xi = (2), \quad w^\xi = (w),$$

$$V = (V), \quad \kappa^\eta = (-1), \quad w^\eta = (w),$$

where,

$$\tilde{U} = [0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 6, 6, 6], \quad w \equiv 1,$$

$$V = [0, 0, 0, 0, 0, 1, 2, 3, 3, 3, 3, 3].$$

This spline space contains only non-rational basis functions. Then, we choose $\mathcal{R}^{\xi\eta}$ as the space built using,

$$U = (U), \quad \kappa^\xi = (2), \quad w^\xi = (w),$$

$$V = (V), \quad \kappa^\eta = (-1), \quad w^\eta = (w),$$

where,

$$U = [0, \underbrace{\dots, 0}_{7 \text{ times}}, \underbrace{1, \dots, 1}_{4 \text{ times}}, \underbrace{2, \dots, 2}_{4 \text{ times}}, \underbrace{3, \dots, 3}_{4 \text{ times}}, \underbrace{4, \dots, 4}_{4 \text{ times}}, \underbrace{5, \dots, 5}_{4 \text{ times}}, \underbrace{6, \dots, 6}_{7 \text{ times}}],$$

and the other parameters are as used for $\mathcal{R}_F^{\xi\eta}$. It can be checked that $\mathcal{R}_F^{\xi\eta}$ and $\mathcal{R}^{\xi\eta}$ are 2-compatible.

The polar basis functions N_l , $l = 1, \dots, 6$, are the only ones that have non-zero values, first- and second-derivatives at the polar point; and these are shown for this configuration in Fig. 20. Note that, in this example, the algorithm for computation of E^2 converged for $\tau_2 = 8\rho_2$.

Example 3.8 (C^2 Surface). Using the basis functions constructed in Example 3.7, we create a polar surface that is C^2 as shown in Fig. 21. Fig. 21(a) shows the chosen control net used to create the surface. Only the control points corresponding to the degree 2 control triangle had non-zero heights. The resulting surface is shown in Fig. 21(b). For $\xi \in \hat{\Omega}^\xi = [0, 6]$ and $\eta \in [0, 0.1] \subset \hat{\Omega}^\eta$, we plot the Gaussian curvatures and the highlight lines for this surface in Figs. 21(c) and 21(d). It can be seen that the distribution of highlight lines is smooth around the polar point ($\eta = 0$; center of the figures), and that the Gaussian curvature is continuous.

4. Application of smooth multi-degree polar splines in design and analysis

In this section we illustrate the efficacy and versatility of the proposed polar spline framework with the help of applications in the context of design and analysis.

4.1. Polar splines in design

An example of smooth hole-filling using a multi-degree polar spline cap was shown in Fig. 2. Next, we demonstrate an example of free-form design.

Example 4.1 (Free-Form Design: Mushroom). Using the framework for C^2 polar splines, we build a mushroom using a single polar surface with a dimple on top as shown in Fig. 22. The control net used for the construction is shown in Fig. 22(a) and the surface can be seen in Fig. 22(b). Note that every single control point is associated with a C^2 basis function, making this representation extremely convenient for designers to work with. Since the extraction operators calculated have non-negative entries, the surface lies within the convex hull of the control net. Fig. 22(c) visualizes the multi-degree nature of the polar surface used for this construction, with bi-degrees $(6, 3)$, $(7, 3)$ and $(8, 3)$ shown using the colors red, green and blue, respectively. Figs. 22(e) and 22(f) show the Gaussian curvature and highlight

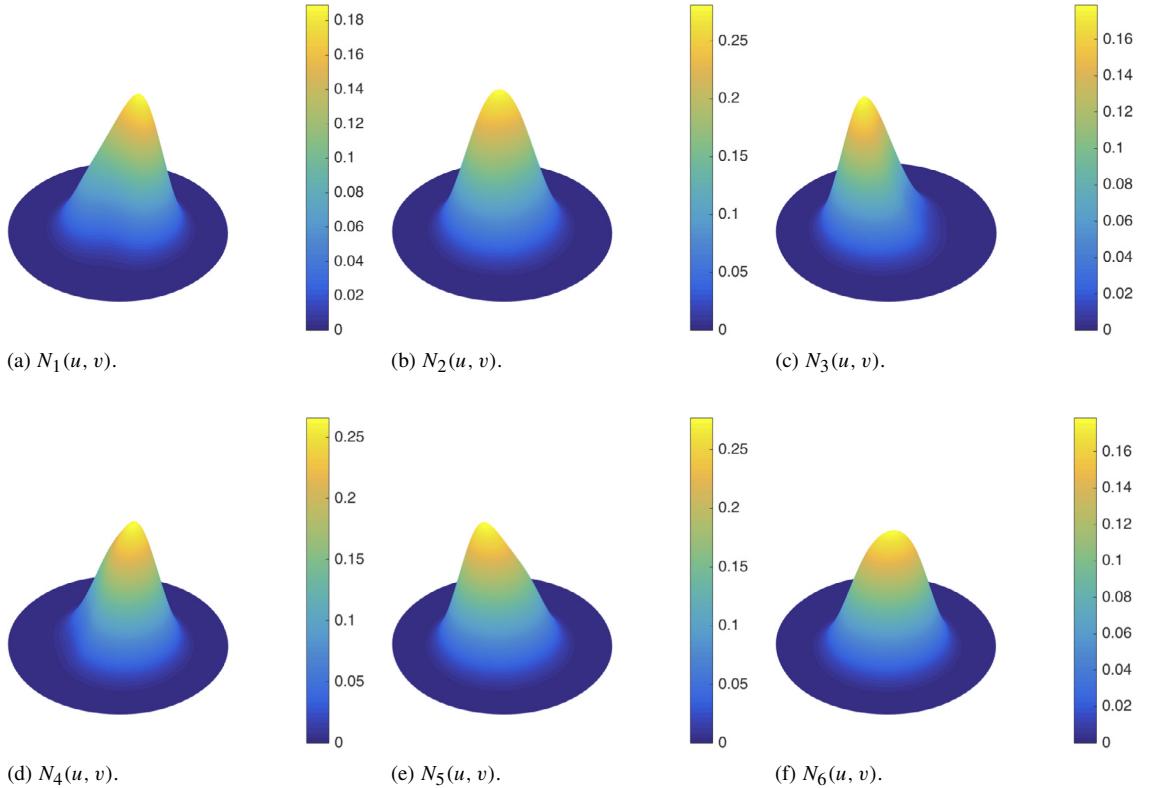


Fig. 20. The 6 polar spline basis functions that have non-zero values, first- and second-derivatives at $(0, 0)$. Any linear combination of these basis functions will be C^2 at the polar point. These basis functions correspond to the data used in Example 3.7.

line distribution in a neighborhood of the polar point (shown in Fig. 22(d)). Note that this neighborhood corresponds to the dimple at the top of the mushroom. As can be seen in Fig. 22(f), the highlight lines are distributed smoothly indicating curvature continuity; Fig. 22(e) confirms this.

4.2. Polar splines in analysis

In order to numerically analyze the approximation properties of the polar spline spaces constructed, we perform analysis using the following spline space configurations. We solve L^2 , H^1 and H^2 projection problems, and observe optimal error convergence rates in all cases.

Configuration 4.1. Consider two C^1 and C^2 polar spline spaces using polynomial tensor-product splines with the following bi-degrees and number of segments.

- C^1 : Bi-degree $(2, 2)$ and $(3, 3)$; both configurations were initialized with 4 segments in the angular direction and 1 segment in the radial direction.
- C^2 : Bi-degree $(6, 5)$ and $(6, 6)$; both configurations were initialized with 6 segments in the angular direction and 1 segment in the radial direction.

Each segment was initialized with a single unit Bézier element, and uniform refinements were carried out for all configurations.

In the following examples, when speaking about convergence rates, we denote by p the lowest degree of the polynomials used for building the considered space. The domain Ω on which the problems are solved is simply taken to be the polar parametric domain for each spline configuration obtained using Eq. (3.11).

Example 4.2 (Analysis: Function Approximation). Using the spline spaces from Configuration 4.1, we want to find a spline s^h that best approximates $\sigma(u, v) = \sin(2\pi(u + 1/3)) \cos(2\pi(v + 1/5))$ in the L^2 norm. For analysis-suitable

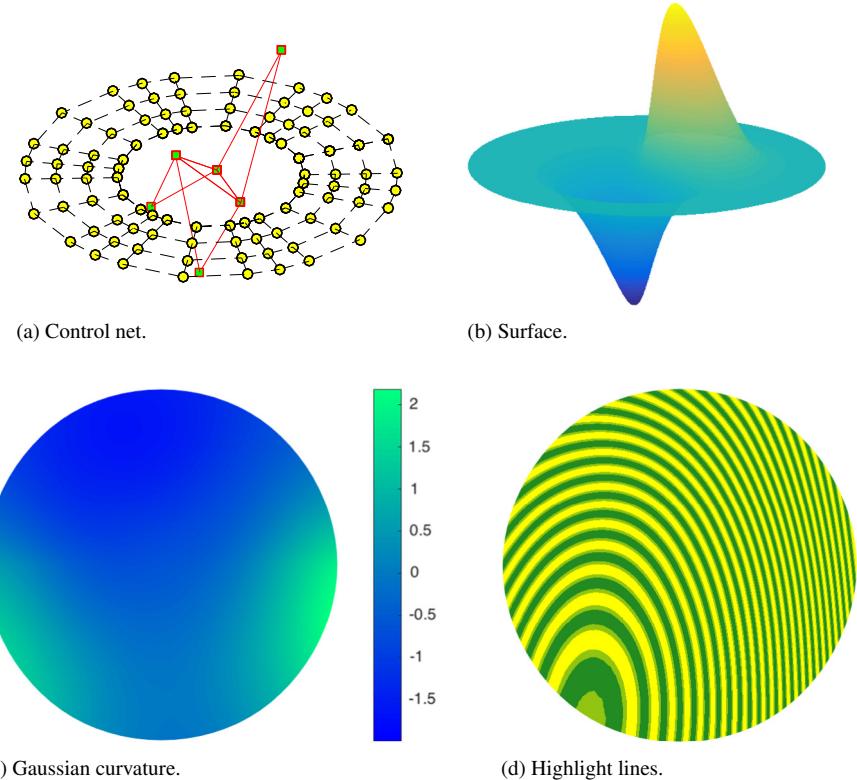


Fig. 21. A C^2 surface built as an arbitrary linear combination of the basis functions created in Example 3.7. See Example 3.8 for details.

spaces, the errors are expected to converge with rates of $p + 1$. The results are shown in Fig. 23. As can be seen in Figs. 23(a) and 23(b), the error converges optimally in the L^2 and L^∞ norms, in each case. Since one might expect the solution to be poorly behaved at the polar point, we also plot the errors at the polar point in Fig. 23(c). We observe that the solution is well-behaved at the polar point in all cases; super convergence is observed for the bi-quadratic configuration.

Example 4.3 (Analysis: Poisson Equation). Using the spline spaces from Configuration 4.1, we solve the following Poisson equation with Dirichlet boundary conditions,

$$\begin{aligned} -\Delta \sigma(u, v) &= 8\pi^2 \sin(2\pi(u + 1/3)) \cos(2\pi(v + 1/5)) \quad \text{on } \Omega, \\ \sigma(u, v) &= \sin(2\pi(u + 1/3)) \cos(2\pi(v + 1/5)) \quad \text{on } \partial\Omega. \end{aligned}$$

The exact solution of the above problem is $\sin(2\pi(u + 1/3)) \cos(2\pi(v + 1/5))$. The errors of the approximate solutions s^h were measured in the L^2 , L^∞ and H^1 norms, and these are shown in Figs. 24(a), 24(b) and 24(c), respectively. For analysis-suitable spaces, the errors are expected to have a convergence rate of $p + 1$ for the L^2 norm (and also L^∞ norm), while a convergence rate of p for the H^1 norm. These optimal rates are clearly observed in Fig. 24. Comparing our results with [42, Figure 10], we see that our approach performs much better than other spline techniques over disks.

Example 4.4 (Analysis: Biharmonic Equation). Using the spline spaces from Configuration 4.1, we solve the following biharmonic problem,

$$\begin{aligned}\Delta^2\sigma(u, v) &= 64\pi^4 \sin(2\pi(u + 1/3)) \cos(2\pi(v + 1/5)) \quad \text{on } \Omega, \\ \sigma(u, v) &= \sin(2\pi(u + 1/3)) \cos(2\pi(v + 1/5)) \quad \text{on } \partial\Omega, \\ \frac{1}{2\pi} \nabla\sigma \cdot \mathbf{n} &= \cos(2\pi(u + 1/3)) \sin(2\pi(v + 1/5)) n_u + \sin(2\pi(u + 1/3)) \cos(2\pi(v + 1/5)) n_v \quad \text{on } \partial\Omega,\end{aligned}$$

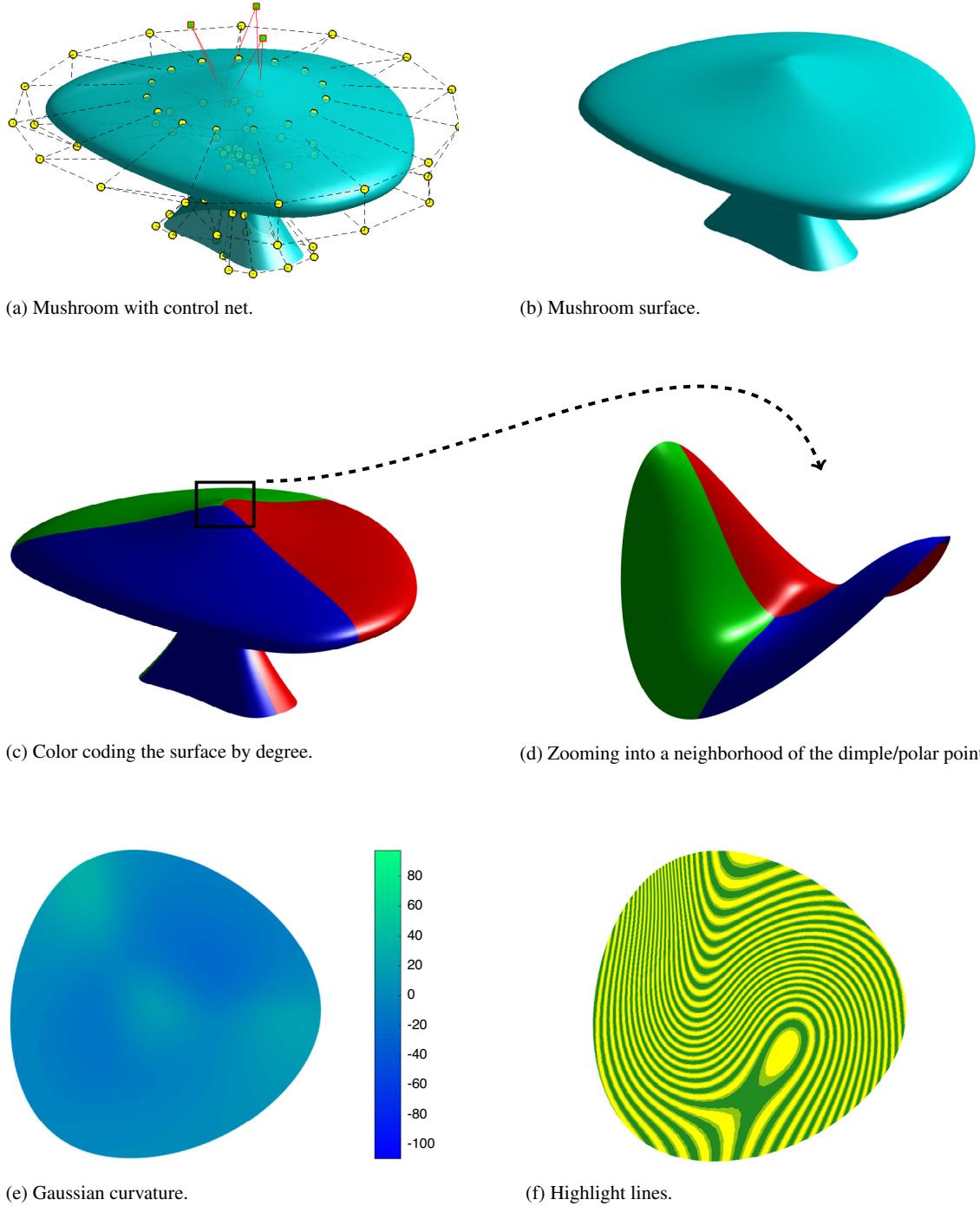


Fig. 22. The mushroom built in Example 4.1 using C^2 polar splines is shown in the top-row figures along with its associated control net. The configuration of the control triangle at the center introduces a dimple at the cap of the mushroom. The bottom-row figures show the Gaussian curvature and highlight line distributions over the cap of the mushroom. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\mathbf{n} := (n_u, n_v)$ is the outward unit normal. The exact solution of the above problem is, again, $\sin(2\pi(u + 1/3)) \cos(2\pi(v + 1/5))$. The errors were measured in the L^2 , L^∞ , H^1 and H^2 norms, and these are shown in Figs. 25(a), 25(b), 25(c) and 25(d), respectively. For analysis-suitable spaces with $p \geq 3$, the errors are

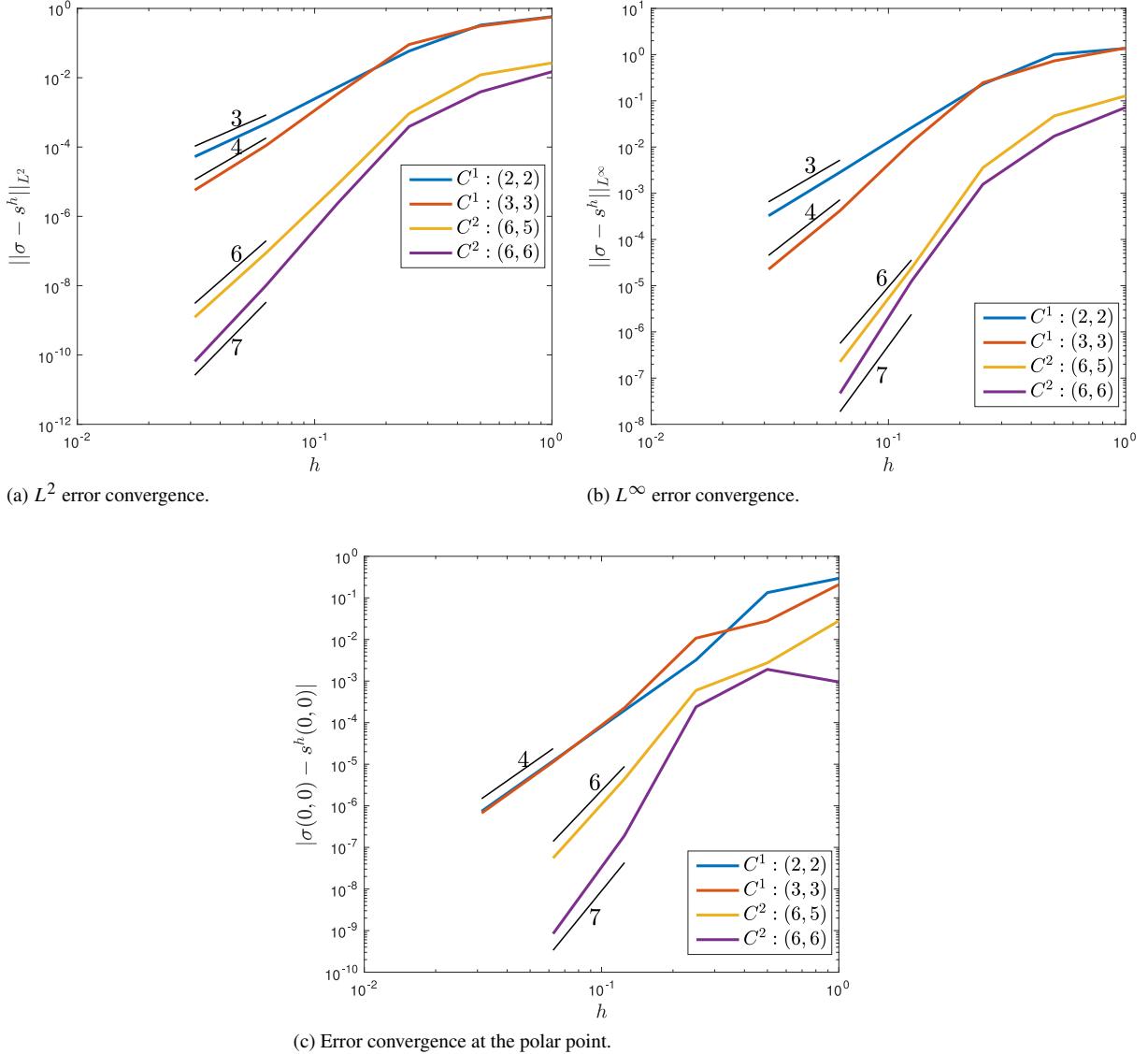


Fig. 23. Errors measured in the L^2 and L^∞ norms for the function approximation problem outlined in Example 4.2, together with the pointwise errors at the polar point.

expected to have a convergence rate of $p + 1$ for the L^2 norm (and also L^∞ norm). According to Aubin–Nitsche theory [47] for the case of $p = 2$, the best possible convergence rate is 2. On the other hand, convergence rates of p and $p - 1$ are expected for the H^1 and H^2 norms, respectively. These optimal rates are clearly observed Fig. 25.

Example 4.5 (Analysis: Cahn–Hilliard). A C^1 polar configuration built using rational quadratic splines in angular and radial directions is used to solve the fourth order non-linear Cahn–Hilliard problem on a disk Ω with radius $R = 1$. The following non-dimensional form is considered (see also [49]):

$$\begin{aligned} \frac{\partial c}{\partial t} &= \nabla \cdot (c(1 - c)\nabla(\mathbb{N}_2\mu_c - \Delta c)) \quad \text{on } \Omega \times [0, T], \\ c(1 - c)\nabla\mu_c \cdot \mathbf{n} &= 0 \quad \text{on } \partial\Omega \times [0, T], \\ c(1 - c)\nabla c \cdot \mathbf{n} &= 0 \quad \text{on } \partial\Omega \times [0, T], \\ c(\mathbf{x}, 0) &= c_0(\mathbf{x}) \quad \text{on } \Omega, \end{aligned}$$

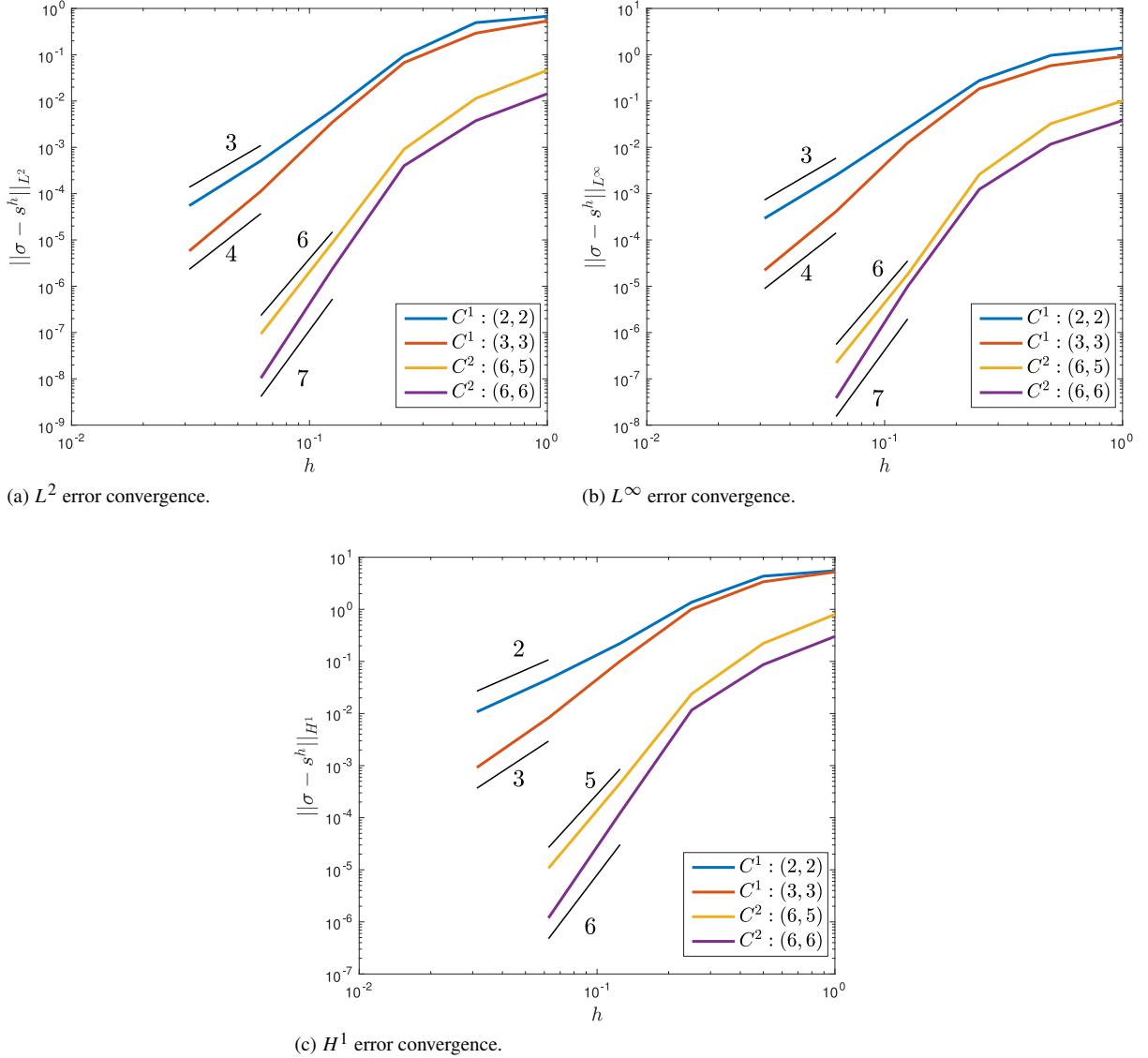


Fig. 24. Errors measured in the L^2 , L^∞ and H^1 norms for the Poisson problem outlined in Example 4.3.

where $\mu_c := \frac{1}{3} \log\left(\frac{c}{1-c}\right) + 1 - 2c$. The considered mesh had 128 knot spans in the angular direction and 20 knot spans in the radial direction. We solved the equations for initial volume-fractions $\bar{c} = 0.3$ and 0.5, and the corresponding values of N_2 were 753.08 and 161.45, respectively. The initial values of c , namely c_0 , are determined by randomly perturbing \bar{c} , as described in [48,49]. The results are shown in Fig. 26. Steady state was reached for the two configurations in 667 and 303 time-steps, respectively. In both cases, the solution coefficients were observed to be strictly between 0 and 1. Then, since the basis functions used were IGA-suitable, no further analysis is needed to conclude that the computed solutions lie strictly between 0 and 1 point-wise.

5. Conclusions

We have developed a framework for bivariate smooth splines in polar coordinates suitable for applications in geometric modeling as well as engineering analysis. We thereby resolve two main drawbacks of current polar spline configurations. First, due to a lack of smooth description of conic sections, traditional low degree ($p \leq 3$) NURBS representations are globally C^0 continuous. In this paper we introduced a generalization of univariate NURBS that

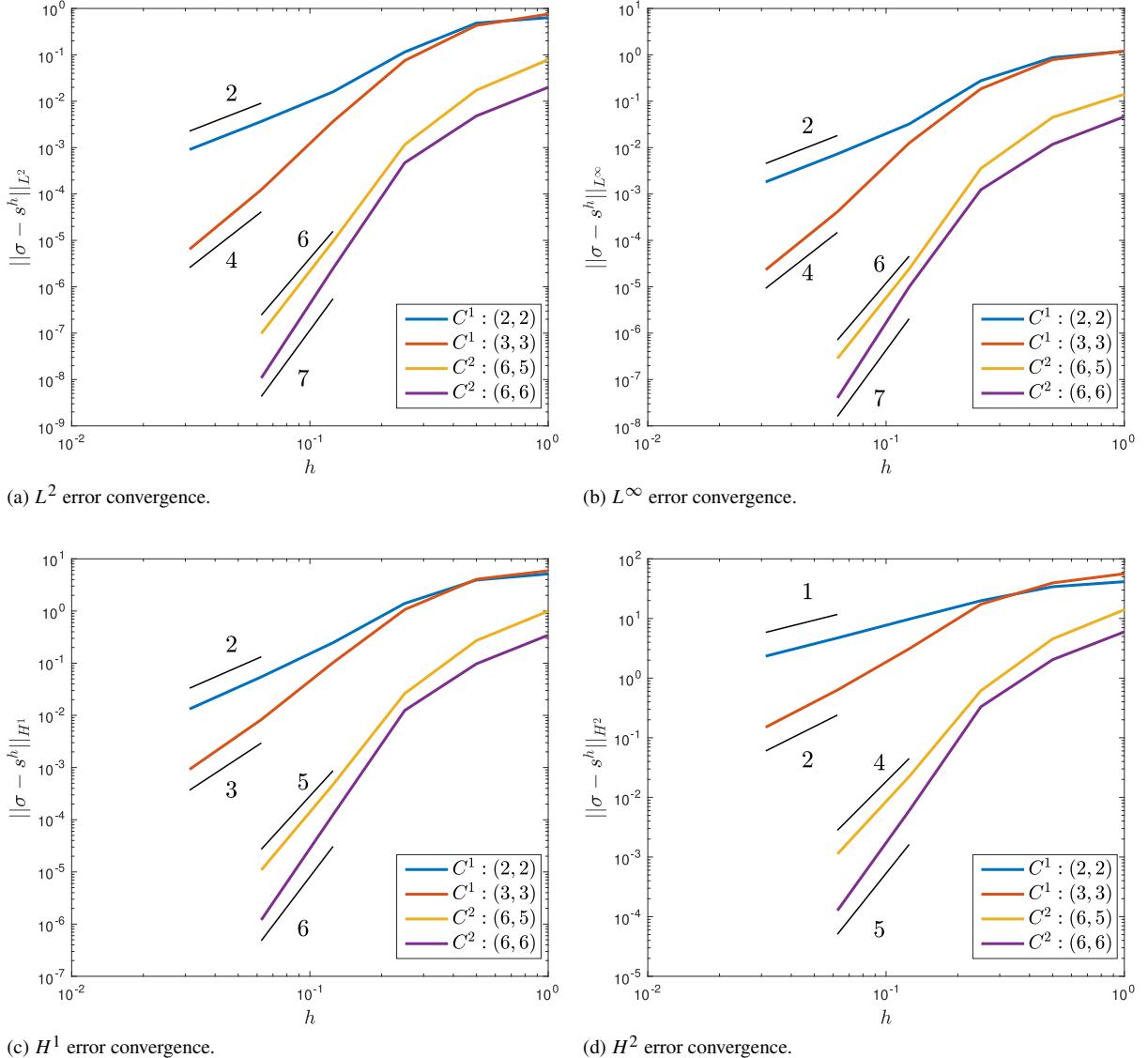


Fig. 25. Errors measured in L^2 , L^∞ , H^1 and H^2 norms for the biharmonic problem outlined in Example 4.4.

help describe smooth conic sections. A simple algorithm for their computation in terms of linear combinations of NURBS was presented. It was shown that this curve representation allowed for non-uniformity in polynomial degree as well. The second drawback of current polar spline configurations stems from the singularity introduced at the origin of the polar patch, leading to a discontinuity there. By specific linear combinations of C^{-1} basis functions, encoded in a geometry and analysis suitable extraction operator, we obtained a set of globally C^k smooth basis functions. Furthermore, explicit constructions for $k \in \{0, 1, 2\}$ were presented.

We showed that the proposed basis has all the properties required for application in geometric modeling, such as partition of unity, non-negativity and local support. Furthermore, we proved linear independence and numerically verified full resolving power, that is best possible rates of convergence, making it suitable in engineering analysis as well. Several examples have shown the versatility and efficacy of the proposed polar spline technology in design and analysis. In particular, the examples illustrated free-form design of multi-degree polar splines with a suitable minimum number of degrees of freedom in a typical control-net structure, smooth hole-filling of splines in polar coordinates, and numerical solution of high-order partial differential equations on polar domains with best possible rates of convergence in the L^2 , H^1 , H^2 norms (and also L^∞ norm).

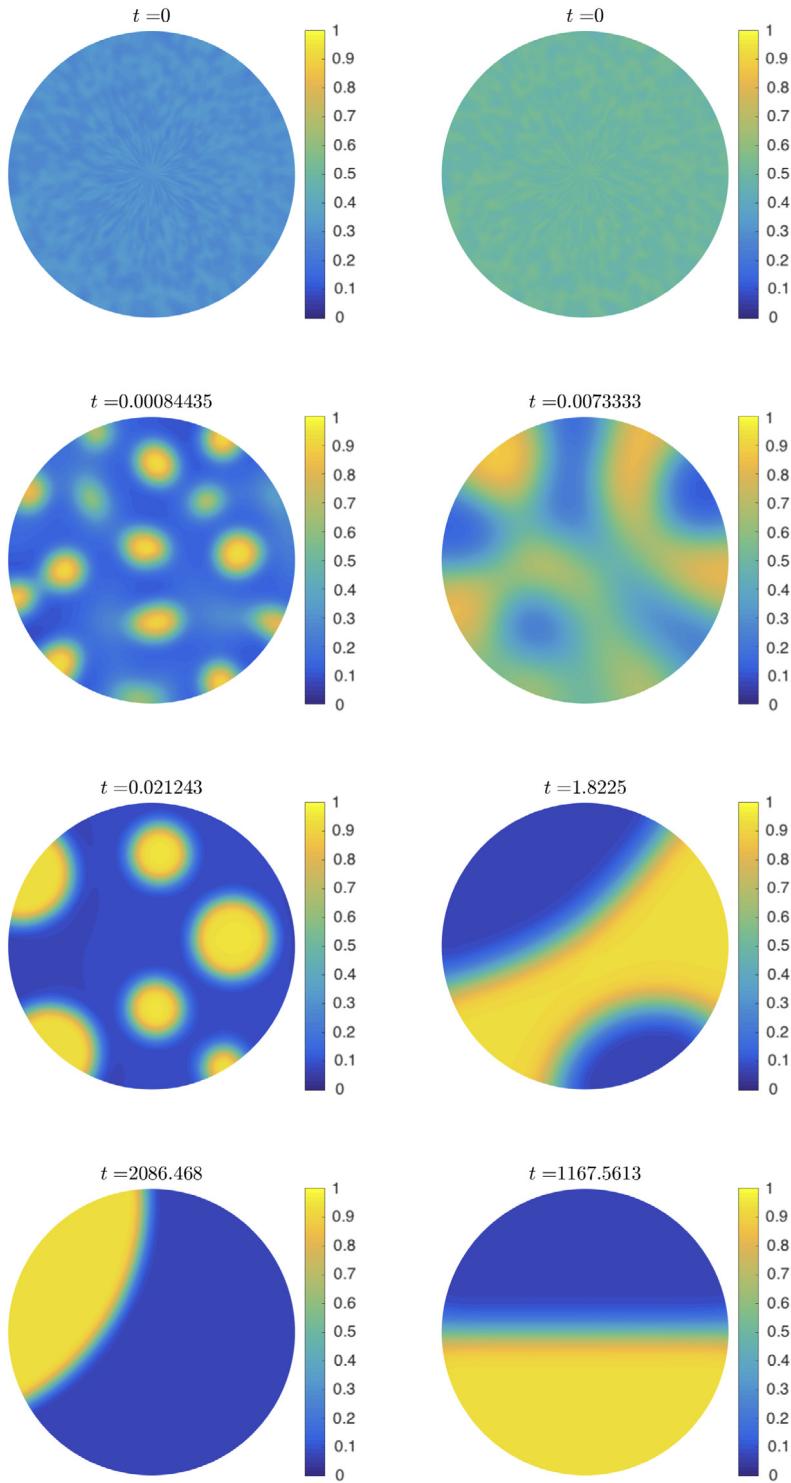


Fig. 26. Solutions to the Cahn–Hilliard problem with volume fractions of 0.3 and 0.5 (left and right, respectively) described in Example 4.5. The formulations are taken from [48,49], and the mesh chosen corresponded to the non-dimensional parameters 753.08 and 161.45 (left and right, respectively).

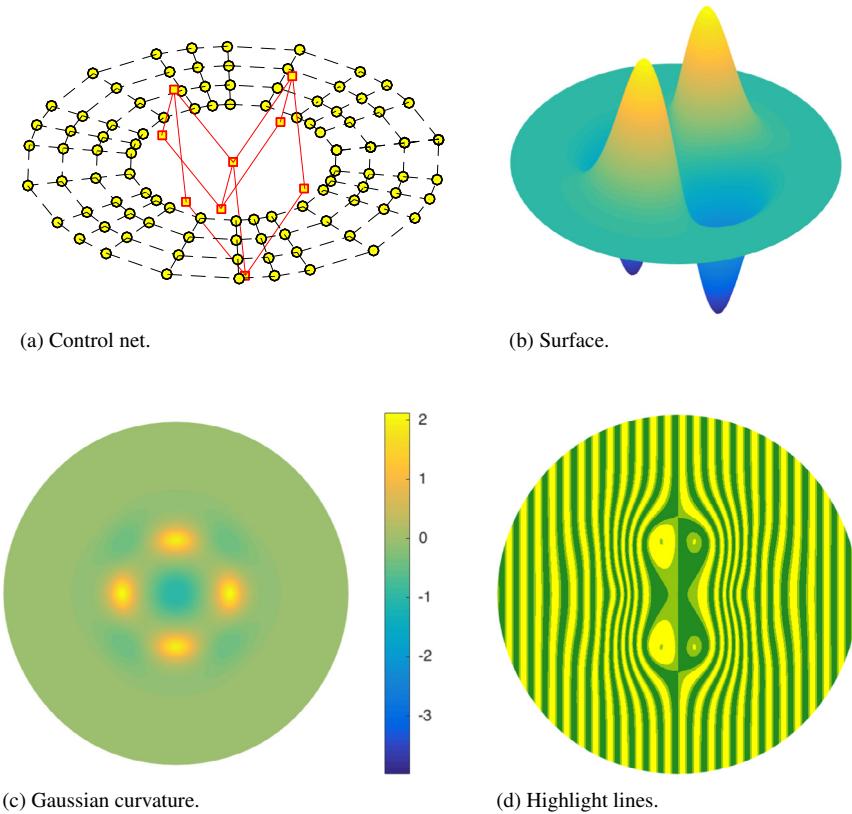


Fig. 27. Process of smoothness imposition at the polar point can be varied to endow favorable properties onto the polar spline basis functions. For instance, in a purely design-oriented application, imposition of smoothness could be done using a tensor-product Bézier patch. Compromising on linear independence, this would allow the designer to use a more conventional control net to manipulate the surface around the polar point. An example of a C^2 polar surface exploiting this flexibility is shown here; bi-quadratic tensor-product Bernstein polynomials were used to generate the Hermite data used in Eq. (3.17) yielding 9 new C^2 basis functions. The control points associated to these have been connected using red lines; the connectivity is the same as for bi-quadratic tensor-product Bézier elements. Figs. 27(c) and 27(d) confirm curvature continuity of the surface. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We see several directions for future research. First, the approach taken to generalize univariate NURBS to non-uniform degree could further be explored to generate bases for a more general class of splines, such as trigonometric, exponential and Tchebycheffian splines [17]. Second, univariate multi-degree splines may be generalized to the multivariate setting by means of a local tensor-product construction. This would allow non-uniformity in the polynomial degree that is local to a Bézier element instead of propagating through an entire patch as is the case in a global tensor-product construction. Third, we would like to extend the treatment of the polar spline point to extraordinary points in T-splines. Local control of polynomial degree around the one ring of elements about an extraordinary point may resolve the current problem of suboptimal approximation power there. Furthermore, it would be interesting to study other configurations to enforce continuity at the polar point whilst maintaining partition of unity, non-negativity and local support. For instance, in purely design-based applications, one could compromise on linear independence to work with a more conventional arrangement of the control points at the center, as shown in Fig. 27. Finally, the proposed polar splines enable smooth parametrization of surfaces in polar coordinates and could serve as an excellent basis with applications in, e.g., the boundary element method and shell and membrane analysis.

Acknowledgments

T.J.R. Hughes was supported by grants from the Office of Naval Research (N00014-08-1-0992) and SINTEF (UTA10-000374), with the University of Texas at Austin. H. Speleers was supported by the MIUR “Futuro in Ricerca

2013” Programme through the project DREAMS (RBFR13FBI3), and by the “Uncovering Excellence” Programme of the University of Rome “Tor Vergata” through the project DEXTEROUS.

Appendix A. Bernstein basis functions and derivatives on \mathcal{T}_k

- $k = 0$: For $l = 1$,

$$T_l(0, 0) = 1.$$

- $k = 1$: For $l = 1, 2, 3$,

$$\begin{aligned} T_l(0, 0) &= \frac{1}{3}, \\ \frac{\partial T_l}{\partial u}(0, 0) &= \frac{2i_1 - i_2 - i_3}{3\tau_1}, \quad \frac{\partial T_l}{\partial v}(0, 0) = \frac{i_2 - i_3}{\sqrt{3}\tau_1}. \end{aligned}$$

- $k = 2$: For $l = 1, 2, 3, 4, 5, 6$,

$$\begin{aligned} T_l(0, 0) &= \binom{2}{i_1 \ i_2 \ i_3} \frac{1}{9}, \\ \frac{\partial T_l}{\partial u}(0, 0) &= \binom{2}{i_1 \ i_2 \ i_3} \frac{2i_1 - i_2 - i_3}{9\tau_2}, \quad \frac{\partial T_l}{\partial v}(0, 0) = \binom{2}{i_1 \ i_2 \ i_3} \frac{i_2 - i_3}{3\sqrt{3}\tau_2}, \\ \frac{\partial^2 T_l}{\partial u^2}(0, 0) &= \binom{2}{i_1 \ i_2 \ i_3} \frac{2 + 3i_1(3i_1 - 5)}{9(\tau_2)^2}, \\ \frac{\partial^2 T_l}{\partial u \partial v}(0, 0) &= \binom{2}{i_1 \ i_2 \ i_3} \frac{(1 - 3i_1)(i_3 - i_2)}{3\sqrt{3}(\tau_2)^2}, \\ \frac{\partial^2 T_l}{\partial v^2}(0, 0) &= \binom{2}{i_1 \ i_2 \ i_3} \frac{(i_3 - i_2)^2 - (i_2 + i_3)}{3(\tau_2)^2}. \end{aligned}$$

Appendix B. Pulling back derivatives

This appendix briefly outlines how to relate derivatives of $s(u, v)$ and $\hat{s}(\xi, \eta)$ from Eq. (3.5), considering the map $\mathbf{F}(\xi, \eta) = (F_u(\xi, \eta), F_v(\xi, \eta))$ in Eq. (3.1). Let us denote the derivatives of \mathbf{F} with respect to ξ and η as follows,

$$\mathbf{F}_{m_1 m_2} := (F_{u,m_1 m_2}, F_{v,m_1 m_2}) := \left(\frac{\partial^{m_1+m_2} F_u}{\partial \xi^{m_1} \partial \eta^{m_2}}, \frac{\partial^{m_1+m_2} F_v}{\partial \xi^{m_1} \partial \eta^{m_2}} \right).$$

Given $m \in \mathbb{N}$ and a multi-index $\mathbf{i} := (i_1, \dots, i_m) \in (\mathbb{N} \cup \{0\})^m$, we define specific differential operators to a function $f(u, v)$,

$$\begin{aligned} D_{\xi, \mathbf{i}}^m f &:= \left(F_{u,m0} \frac{\partial}{\partial u} + F_{v,m0} \frac{\partial}{\partial v} \right)^{i_m} \cdots \left(F_{u,10} \frac{\partial}{\partial u} + F_{v,10} \frac{\partial}{\partial v} \right)^{i_1} f, \quad D_{\xi, 0}^0 f := f, \\ D_{\eta, \mathbf{i}}^m f &:= \left(F_{u,0m} \frac{\partial}{\partial u} + F_{v,0m} \frac{\partial}{\partial v} \right)^{i_m} \cdots \left(F_{u,01} \frac{\partial}{\partial u} + F_{v,01} \frac{\partial}{\partial v} \right)^{i_1} f, \quad D_{\eta, 0}^0 f := f. \end{aligned} \tag{B.1}$$

Given $m_1, m_2 \in \mathbb{N}$ and a multi-index $\mathbf{i} := (i_1, \dots, i_{m_2}) \in (\mathbb{N} \cup \{0\})^{m_2}$, we also define,

$$D_{\eta, \mathbf{i}}^{m_1, m_2} f := \frac{\partial^{m_1} D_{\eta, \mathbf{i}}^{m_2}}{\partial \xi^{m_1}} f, \quad D_{\eta, 0}^{m_1, 0} f := 0, \quad D_{\eta, \mathbf{i}}^{0, m_2} f := D_{\eta, \mathbf{i}}^{m_2} f, \quad D_{\eta, 0}^{0, 0} f := f. \tag{B.2}$$

We denote with ξ_m the following set of solutions of a Diophantine equation,

$$\xi_m := \left\{ \mathbf{i} := (i_1, \dots, i_m) \in (\mathbb{N} \cup \{0\})^m : \sum_{l=1}^m l i_l = m \right\}, \quad m \geq 1,$$

and $\zeta_0 := \{0\}$. Then, with the help of [50], we can express the derivatives of \hat{s} in terms of the derivatives of s using the following chain rule for higher-order derivatives,

$$\frac{\partial^{m_1+m_2} \hat{s}}{\partial \xi^{m_1} \partial \eta^{m_2}} = \sum_{l=0}^{m_1} \binom{m_1}{l} \sum_{i \in \zeta_{m_1-l}} \sum_{j \in \zeta_{m_2}} C_i C_j D_{\xi, i}^{m_1-l} D_{\eta, j}^{l, m_2} s, \quad (\text{B.3})$$

where $C_i := \frac{m!}{i_1! \cdots i_m! (1!)^{i_1} \cdots (m!)^{i_m}}$ for $i \in \zeta_m$, and $C_0 := 1$.

References

- [1] K. Karčiauskas, J. Peters, Finite curvature continuous polar patchworks, in: E.R. Hancock, et al. (Eds.), Mathematics of Surfaces XIII, in: LNCS, vol. 5654, Springer-Verlag, 2009, pp. 222–234.
- [2] A. Myles, K. Karčiauskas, J. Peters, Pairs of bi-cubic surface constructions supporting polar connectivity, *Comput. Aided Geom. Design* 25 (2008) 621–630.
- [3] A. Myles, J. Peters, C^2 splines covering polar configurations, *Comput. Aided Des.* 43 (2011) 1322–1329.
- [4] K.-L. Shi, J.-H. Yong, J.-G. Sun, J.-C. Paul, G^n filling orbicular n -sided holes using periodic B-spline surfaces, *Sci. China Inf. Sci.* 54 (2011) 1383–1394.
- [5] K.-L. Shi, J.-H. Yong, L. Tang, J.-G. Sun, J.-C. Paul, Polar NURBS surface with curvature continuity, *Comput. Graph. Forum* 32 (2013) 363–370.
- [6] Y. Zhang, Y. Bazilevs, S. Goswami, C.L. Bajaj, T.J.R. Hughes, Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow, *Comput. Methods Appl. Mech. Engrg.* 196 (2007) 2943–2959.
- [7] A. Karimi, M. Navidbakhsh, R. Razaghi, M. Haghpanahi, A computational fluid–structure interaction model for plaque vulnerability assessment in atherosclerotic human coronary arteries, *J. Appl. Phys.* 115 (2014) 144702.
- [8] J. Lu, X. Zhou, Cylindrical element: Isogeometric model of continuum rod, *Comput. Methods Appl. Mech. Engrg.* 200 (2011) 233–241.
- [9] A. Bartezzaghi, L. Dedè, A. Quarteroni, Isogeometric analysis of high order partial differential equations on surfaces, *Comput. Methods Appl. Mech. Engrg.* 295 (2015) 446–469.
- [10] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley & Sons, 2009.
- [11] T. Takacs, B. Jüttler, Existence of stiffness matrix integrals for singularly parameterized domains in isogeometric analysis, *Comput. Methods Appl. Mech. Engrg.* 200 (2011) 3568–3582.
- [12] T. Takacs, B. Jüttler, H^2 regularity properties of singular parameterizations in isogeometric analysis, *Graph. Models* 74 (2012) 361–372.
- [13] J. Lu, Circular element: Isogeometric elements of smooth boundary, *Comput. Methods Appl. Mech. Engrg.* 198 (2009) 2391–2402.
- [14] M.J. Borden, M.A. Scott, J.A. Evans, T.J.R. Hughes, Isogeometric finite element data structures based on Bézier extraction of NURBS, *Internat. J. Numer. Methods Engrg.* 87 (2011) 15–47.
- [15] L. Piegl, W. Tiller, *The NURBS Book*, Springer-Verlag, 2012.
- [16] C. de Boor, *A Practical Guide to Splines*, Revised Edition, Springer-Verlag, 2001.
- [17] L.L. Schumaker, *Spline Functions: Basic Theory*, Third Edition, Cambridge University Press, 2007.
- [18] T.W. Sederberg, J. Zheng, X. Song, Knot intervals and multi-degree splines, *Comput. Aided Geom. Design* 20 (2003) 455–468.
- [19] W. Shen, G. Wang, Changeable degree spline basis functions, *J. Comput. Appl. Math.* 234 (2010) 2516–2529.
- [20] W. Shen, G. Wang, A basis of multi-degree splines, *Comput. Aided Geom. Design* 27 (2010) 23–35.
- [21] W. Shen, G. Wang, P. Yin, Explicit representations of changeable degree spline basis functions, *J. Comput. Appl. Math.* 238 (2013) 39–50.
- [22] X. Li, Z.J. Huang, Z. Liu, A geometric approach for multi-degree splines, *J. Comput. Sci. Tech.* 27 (2012) 841–850.
- [23] B.A. Barsky, *Computer Graphics and Geometric Modeling Using Beta-splines*, Springer-Verlag, 1988.
- [24] T.N.T. Goodman, Properties of β -splines, *J. Approx. Theory* 44 (1985) 132–153.
- [25] G. Seroussi, B.A. Barsky, An explicit derivation of discretely shaped beta-spline basis functions of arbitrary order, in: T. Lyche, L.L. Schumaker (Eds.), *Mathematical Methods in Computer-Aided Geometric Design II*, Academic Press, 1992, pp. 567–584.
- [26] A. Collin, G. Sangalli, T. Takacs, Analysis-suitable G^1 multi-patch parametrizations for C^1 isogeometric spaces, *Comput. Aided Geom. Design* 47 (2016) 93–113.
- [27] M. Kapl, F. Buchegger, M. Bercovier, B. Jüttler, Isogeometric analysis with geometrically continuous functions on planar multi-patch geometries, *Comput. Methods Appl. Mech. Engrg.* 316 (2017) 209–234.
- [28] H. Prautzsch, Freeform splines, *Comput. Aided Geom. Design* 14 (1997) 201–206.
- [29] U. Reif, TURBS—Topologically unrestricted rational B-splines, *Constr. Approx.* 14 (1998) 47–77.
- [30] A. Levin, Modified subdivision surfaces with continuous curvature, *ACM Trans. Graph.* 25 (2006) 1035–1040.
- [31] K. Karčiauskas, J. Peters, Bicubic polar subdivision, *ACM Trans. Graph.* 26 (2007) Article No. 14.
- [32] K. Karčiauskas, A. Myles, J. Peters, A C^2 polar jet subdivision, in: K. Polthier, A. Sheffer (Eds.), *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, Eurographics Association, 2006, pp. 173–180.
- [33] A. Myles, J. Peters, Bi-3 C^2 polar subdivision, *ACM Trans. Graph.* 28 (2009) Article No. 48.
- [34] J.C. Navau, N.P. Garcia, Modeling surfaces from meshes of arbitrary topology, *Comput. Aided Geom. Design* 17 (2000) 643–671.
- [35] L. Ying, D. Zorin, A simple manifold-based construction of surfaces of arbitrary smoothness, *ACM Trans. Graph.* 23 (2004) 271–275.
- [36] J.A. Gregory, J. Zhou, Filling polygonal holes with bicubic patches, *Comput. Aided Geom. Design* 11 (1994) 391–410.
- [37] L.A. Piegl, W. Tiller, Filling n -sided regions with NURBS patches, *Vis. Comput.* 15 (1999) 77–89.

- [38] A. Levin, Filling N -sided holes using combined subdivision schemes, in: P.-J. Laurent, et al. (Eds.), *Curve and Surface Design: Saint-Malo 1999*, Vanderbilt University Press, 2000, pp. 221–228.
- [39] K.-L. Shi, J.-H. Yong, J.-G. Sun, J.-C. Paul, G^n blending multiple surfaces in polar coordinates, *Comput. Aided Des.* 42 (2010) 479–494.
- [40] T. Nguyen, J. Peters, Refinable C^1 spline elements for irregular quad layout, *Comput. Aided Geom. Design* 43 (2016) 123–130.
- [41] T. Takacs, Construction of smooth isogeometric function spaces on singularly parameterized domains, in: J.-D. Boissonnat, et al. (Eds.), *Curves and Surfaces*, in: LNCS, vol. 9213, Springer, 2015, pp. 433–451.
- [42] T. Nguyen, K. Karčiauskas, J. Peters, A comparative study of several classical, discrete differential and isogeometric methods for solving Poisson's equation on the disk, *Axioms* 3 (2014) 280–300.
- [43] W. Tiller, Knot-removal algorithms for NURBS curves and surfaces, *Comput. Aided Des.* 24 (1992) 445–453.
- [44] M.J. Lai, L.L. Schumaker, *Spline Functions on Triangulations*, Cambridge University Press, 2007.
- [45] H. Speleers, Construction of normalized B-splines for a family of smooth spline spaces over Powell–Sabin triangulations, *Constr. Approx.* 37 (2013) 41–72.
- [46] H. Speleers, P. Dierckx, S. Vandewalle, Weight control for modelling with NURPS surfaces, *Comput. Aided Geom. Design* 24 (2007) 179–186.
- [47] G. Strang, G.J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall Englewood Cliffs, 1973.
- [48] H. Gómez, V.M. Calo, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of the Cahn–Hilliard phase-field model, *Comput. Methods Appl. Mech. Engrg.* 197 (2008) 4333–4352.
- [49] J. Liu, L. Dedè, J.A. Evans, M.J. Borden, T.J.R. Hughes, Isogeometric analysis of the advective Cahn–Hilliard equation: Spinodal decomposition under shear flow, *J. Comput. Phys.* 242 (2013) 321–350.
- [50] H.N. Huang, S. Marcantognini, N. Young, Chain rules for higher derivatives, *Math. Intelligencer* 28 (2006) 61–69.