A General Class of C^1 Smooth Rational Splines
Application to Construction of Exact Ellipses and Ellipsoids

Speleers, Hendrik; Toshniwal, Deepesh

DOI
10.1016/j.cad.2020.102982

Publication date
2021

Document Version
Final published version

Published in
CAD Computer Aided Design

Citation (APA)
Speleers, H., & Toshniwal, D. (2021). A General Class of C^1 Smooth Rational Splines: Application to Construction of Exact Ellipses and Ellipsoids. *CAD Computer Aided Design*, *132*, Article 102982. https://doi.org/10.1016/j.cad.2020.102982

Computer-Aided Design 132 (2021) 102982

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

A General Class of C^1 Smooth Rational Splines: Application to Construction of Exact Ellipses and Ellipsoids ☆

Hendrik Speleers [a], Deepesh Toshniwal [b,*]

[a] Department of Mathematics, University of Rome Tor Vergata, Italy
[b] Delft Institute of Applied Mathematics, Delft University of Technology, The Netherlands

ARTICLE INFO

Article history:
Received 18 May 2020
Received in revised form 23 November 2020
Accepted 30 November 2020

Keywords:
Piecewise-NURBS representations
Smooth parameterizations
Exact ellipses and ellipsoids

ABSTRACT

In this paper, we describe a general class of C^1 smooth rational splines that enables, in particular, exact descriptions of ellipses and ellipsoids — some of the most important primitives for CAD and CAE. The univariate rational splines are assembled by transforming multiple sets of NURBS basis functions via so-called design-through-analysis compatible extraction matrices; different sets of NURBS are allowed to have different polynomial degrees and weight functions. Tensor products of the univariate splines yield multivariate splines. In the bivariate setting, we describe how similar design-through-analysis compatible transformations of the tensor-product splines enable the construction of smooth surfaces containing one or two polar singularities. The material is self-contained, and is presented such that all tools can be easily implemented by CAD or CAE practitioners within existing software that support NURBS. To this end, we explicitly present the matrices (a) that describe our splines in terms of NURBS, and (b) that help refine the splines by performing (local) degree elevation and knot insertion. Finally, all C^1 spline constructions yield spline basis functions that are locally supported and form a convex partition of unity.

1. Introduction

Multivariate splines are used extensively for computer-aided design (CAD) and, more recently, for computer-aided engineering (CAE). Smoothness of such splines is a particularly valuable trait. When the aim is to create a (freeform) geometric model for a smooth object, it helps if the splines used for the task are smooth themselves. For instance, this circumvents situations where small displacements to control points may produce 'non-smooth' features such as C^0 kinks, loss of curvature continuity, etc. Similarly, when the aim is to numerically approximate the solution to high-order partial differential equations (PDEs) using isogeometric analysis (IGA) — a generalization of classical finite element analysis [1] — high smoothness of the approximating spaces can be beneficial. For instance, it can allow us to directly discretize the PDEs without any auxiliary variables, thus yielding simpler and more efficient implementations.

In this paper, we discuss a general class of C^1 smooth rational splines that allow for the construction of C^1 smooth curves and surfaces. These are an extension of classical C^1 non-uniform rational B-splines (NURBS) as they enjoy the flexibility of choosing locally unrelated weight functions as well as the option of local degree elevation — they can be roughly regarded as piecewise-NURBS. At the same time, they maintain intuitive control-point-based design. Moreover, they enable simple (low-degree) and smooth descriptions of some of the most important primitives for CAD and CAE (but also for computer vision, graphics and robotics): closed, real, non-degenerate quadrics — that is, ellipses in two dimensions and ellipsoids in three dimensions.

The ideas we present here build upon those from [2], in multiple directions, and their presentation is motivated by our primary objectives: *self-contained, explicit, NURBS-compatible descriptions that can be easily and efficiently implemented within existing CAD software*. The most important novel contributions are the following.

- We describe the usage of classical univariate NURBS to assemble C^1 *rational multi-degree spline basis functions* using an extraction matrix. The general framework was explained in [2], but we provide here a simplified exposition of the construction and a formal proof of the properties; see Remark 2.3. We mainly stick to parametric smoothness, but a construction centered around the notion of geometric smoothness can be formulated as well; see Remark 2.5.
- We describe efficient refinement of the C^1 splines leveraging classical NURBS refinement. The novelty here relies in an explicit and simple construction of the refinement matrices.

☆ This paper has been recommended for acceptance by Myung-Soo Kim.

* Corresponding author.
E-mail addresses: speleers@mat.uniroma2.it (H. Speleers), d.toshniwal@tudelft.nl (D. Toshniwal).

https://doi.org/10.1016/j.cad.2020.102982

Table 1
An overview of the explicit C^1 descriptions of quadrics presented in this paper. The table also compares the number of degrees of freedom (DOFs) needed by our C^1 representation compared to those needed by an equivalent C^{-1} NURBS representation.

Quadric	Polynomial degree	# rational pieces	# DOFs	# DOFs for C^{-1} NURBS	Section
Ellipse (special case: circle)	Uniform; 2	4	4	12	2.4.1
	Uniform; 3	2		8	2.4.2
	Non-uniform; (3, 2, 2)	3		10	2.4.3
Ellipsoid (special case: sphere)	Uniform; (2, 2)	8	6	72	3.4.1
	Uniform; (2, 3)	4		48	3.4.2
	Uniform; (3, 3)	2		32	3.4.3

- We describe how tensor-product bivariate C^1 rational splines can be used to build C^1 smooth geometries that may contain one or two polar singularities; the C^1 smooth splines describing the geometries are called *polar splines*. As above, the idea is based on building an extraction matrix.
- We describe efficient refinement of polar splines. In particular, we provide an explicit and simple construction of the refinement matrices.
- We provide explicit descriptions of ellipses and ellipsoids built using low-degree C^1 splines, and we detail their extraction in terms of NURBS so that they can be readily implemented and used in CAD or CAE software. Table 1 summarizes the descriptions included in this paper.

1.1. Extraction matrices

At the core of our approach is the notion of the so-called *design-through-analysis (DTA) compatible extraction matrix.*[1] Roughly speaking, such matrix helps us assemble 'simple splines' into 'more general splines.' Examples are the Bézier extraction matrix introduced to assemble Bernstein polynomials into B-/T-splines [3,4]; the multi-degree extraction matrix for assembling elements of extended Tchebycheff spaces into generalized Tchebycheffian B-splines [5–7]; and the unstructured spline extraction matrices for assembling tensor-product splines into splines on unstructured quadrilateral meshes [2,8,9].

Here, we apply the concept of extraction in the following context. We start from multiple sets of (univariate or bivariate) NURBS basis functions defined on adjacent domains, and collect all of these functions in the set $\{b_j : j = 1, \ldots, m\}$. Then, we assemble them into more general C^1 rational (polar) splines using a matrix $\boldsymbol{C}$ (with entries C_{ij}), called the *extraction matrix*. Denote this new set of splines by $\{N_i : i = 1, \ldots, n\}$, where $n < m$. These are defined as follows,

$$N_i = \sum_{j=1}^{m} C_{ij} b_j, \quad i = 1, \ldots, n. \tag{1}$$

We are particularly interested in matrices $\boldsymbol{C}$ such that the functions N_i

- satisfy certain smoothness constraints that may or may not be satisfied by the b_j, and
- possess the properties of non-negativity, locality, linear independence and partition of unity that the b_j already possess.

Such extraction matrices are called *DTA-compatible*.

[1] Our notion of *DTA-compatible extraction matrix* has been called *IGA-suitable extraction matrix* in [2]. The name reflects the fact that both design and analysis may profit from the extraction operation.

Definition 1.1 (*DTA-compatible Extraction*)**.** An extraction matrix $\boldsymbol{C}$ is called DTA-compatible if

(a) $\boldsymbol{C}$ is a full-rank matrix,
(b) each column of $\boldsymbol{C}$ sums to 1,
(c) each entry in $\boldsymbol{C}$ is non-negative, and
(d) $\boldsymbol{C}$ imparts locality to the functions N_i through sparsity.

It is easy to see that the action of a DTA-compatible extraction matrix 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), we have

$$\sum_{i=1}^{n} N_i = \sum_{i=1}^{n} \sum_{j=1}^{m} C_{ij} b_j = \sum_{j=1}^{m} b_j \sum_{i=1}^{n} C_{ij} = 1,$$

as the b_j form a partition of unity. Since $\boldsymbol{C}$ 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.

1.2. Related literature

As mentioned in the previous section, the construction of smooth univariate splines by joining simpler pieces has been recently explored in [2,5,6] for polynomial multi-degree splines, and in [7] for generalized Tchebycheffian splines. These approaches have conceptual similarities with the notion of beta-splines [10]. The main differences are that the former approaches do not rely on symbolic computations while the latter does, and the former approaches consider parametric continuity while the latter studies geometric continuity. The use of smooth univariate rational splines for construction of circles has been previously explored in [11–13]. It is known that a circle cannot be represented by a single (symmetric) periodic C^1 quadratic NURBS curve [14, Section 7.5] nor a C^2 cubic NURBS curve [15, Section 13.7]. However, it is possible to find C^k smooth descriptions using NURBS of degree $2(k + 1)$, which is shown to be the minimal degree in [11]. On the other hand, [12,13] presented a C^1 piecewise quadratic NURBS description of the circle and used it for IGA. Our rational multi-degree splines form a flexible extension of the latter framework, and allow for a variety of exact descriptions of circles using low (multi-)degrees, as indicated in Table 1.

In two dimensions, closed quadrics or, more generally, smooth closed surfaces of genus zero can be built using tensor-product splines by introducing polar singularities. For such polar surfaces, subdivision schemes producing C^1 surfaces [16,17] and C^2 surfaces [18,19] have been previously worked out. The corresponding limit surfaces consist of an infinite sequence of surface rings where the faces shrink to a point in the limit. A more CAD-friendly finite construction was developed in [20]; this approach 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. The 'shape' basis does not enjoy non-negativity and does not form a partition of unity, and extensions of it to higher smoothness leads

to degrees of freedom that control non-intuitive shape parameters. Similar recent constructions for obtaining C^1 polar spline caps can be found in [21]. Curvature continuous polar NURBS surfaces were discussed in [22], and [23] presented a construction of polar caps using periodic B-spline surfaces with G^k continuity for arbitrary k. On the CAE side, a standard circular serendipity-type element for IGA was proposed in [12], and C^k smooth basis functions over singular parameterizations of triangular domains were constructed in [24]. A design-through-analysis friendly construction of C^k smooth polar surfaces was recently proposed in [2], and the current work builds further upon this construction.

A completely different approach for dealing with curves and surfaces is the use of implicit representations [25]. Such representations enjoy nice geometric properties (especially for simple shapes). For instance, they allow for a straightforward point membership classification. On the other hand, explicit smooth B-spline representations are more convenient for direct geometric modeling and (local) modification.

1.3. Outline

In Section 2, we present the construction and refinement of C^1 smooth rational multi-degree spline curves via explicitly defined extraction and refinement matrices. The construction and refinement of C^1 smooth polar surfaces using tensor products of the univariate splines is detailed in Section 3. The explicit descriptions of ellipses and ellipsoids using the univariate and bivariate C^1 splines are reported in Sections 2.4 and 3.4, respectively. Finally, we conclude the paper in Section 4. It should be mentioned that the text is written such that the theoretical sections — Sections 2.1–2.3 and 3.1–3.3 — can be skipped by readers interested only in implementing explicit descriptions of smooth quadrics.

2. Piecewise-rational curves

In this section, we focus on a multi-degree extension of univariate NURBS splines. The multi-degree spline space is defined as a collection of classical NURBS spaces (with possibly different polynomial degrees and weight functions) glued together C^1 smoothly. For such space we present a construction of a set of basis functions, with similar properties to classical NURBS. After discussing some preliminary material on NURBS in Section 2.1, we elaborate how these basis functions can be computed through a DTA-compatible extraction matrix in Section 2.2. A more general but also more complex algorithmic construction has been detailed in [2, Section 2] and further explored in [5,6] for polynomial multi-degree splines. Then, in Section 2.3, we give an explicit procedure how to compute a refined representation of a given curve. Finally, in Section 2.4, we illustrate how this tool can be used to describe arbitrary ellipses in a C^1 smooth fashion using low-degree piecewise-rational curve representations suited for integrated design and analysis.

2.1. Preliminaries on NURBS

We start by defining notation for NURBS basis functions, and introduce some classical relations that can be found, e.g., in [14, 26].

Given a basic interval $I := [x_1, x_2] \subset \mathbb{R}$, let us denote with $\boldsymbol{\xi}$ an *open knot vector* of degree $p \in \mathbb{N}$ and length $n+p+1 \in \mathbb{N}$, i.e.,

$$\begin{aligned} \boldsymbol{\xi} &:= [\xi_1, \xi_2, \dots, \xi_{n+p+1}], \quad \xi_{i+1} \geq \xi_i, \\ &\xi_1 = \cdots = \xi_{p+1} = x_1 < \xi_{p+2}, \\ &\xi_{n+1} = \cdots = \xi_{n+p+1} = x_2 > \xi_n. \end{aligned} \tag{2}$$

The number of times a knot value ξ_i is duplicated in the knot vector is called the knot's *multiplicity*. The multiplicity of ξ_i is denoted with m_i, and we assume that $1 \leq m_i \leq p-1$. The corresponding set of B-splines $\{b_{j,p} : j = 1, \dots, n\}$ are defined using the recursive relation,

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

starting from

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

and under the convention that fractions with zero denominator have value zero. With the above definition, all the B-splines take the value zero at the end point x_2. Therefore, in order to avoid asymmetry over the interval I, it is common to assume the B-splines to be left continuous at x_2. We will follow suit.

Let us denote with $\boldsymbol{w}$ a *weight vector* of length n, i.e.,

$$\boldsymbol{w} := [w_1, w_2, \dots, w_n], \quad w_i > 0. \tag{3}$$

The corresponding set of NURBS $\{b^w_{j,p} : j = 1, \dots, n\}$ are defined by

$$b^w_{j,p}(x) := \frac{w_j b_{j,p}(x)}{\sum_{i=1}^n w_i b_{i,p}(x)}.$$

Each $b^w_{j,p}$ is non-negative on I and is locally supported on $[\xi_j, \xi_{j+p+1}]$. Moreover, the functions $b^w_{j,p}$ are linearly independent and form a partition of unity. They satisfy the following end-point conditions:

$$\begin{aligned} &b^w_{1,p}(x_1) = 1, \quad b^w_{j,p}(x_1) = 0, \quad j = 2, \dots, n, \\ &b^w_{n,p}(x_2) = 1, \quad b^w_{j,p}(x_2) = 0, \quad j = 1, \dots, n-1. \end{aligned}$$

The NURBS space corresponding to $\boldsymbol{\xi}$ and $\boldsymbol{w}$ is denoted with $\mathscr{R}[\boldsymbol{\xi}, \boldsymbol{w}]$ and is defined as the span of $\{b^w_{j,p} : j = 1, \dots, n\}$. This is a space of piecewise-rational functions of degree p with smoothness C^{p-m_i} at knot ξ_i and its dimension is n. The assumption on the multiplicity will ensure us global C^1 smoothness. Note that when $w_1 = \cdots = w_n$, the members of this space are piecewise-polynomial.

Remark 2.1. The structure of $\boldsymbol{\xi}$ in Eq. (2) is such that p, m_i, n and I are embedded in it. Therefore, we will assume that once a knot vector $\boldsymbol{\xi}$ is known, so are the degree, smoothness, and dimension of the corresponding NURBS space $\mathscr{R}[\boldsymbol{\xi}, \boldsymbol{w}]$.

We identify a function $f \in \mathscr{R}[\boldsymbol{\xi}, \boldsymbol{w}]$ with the vector of its coefficients $[f_1, \dots, f_n]$,

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

Only the first (last) $k+1$ basis functions contribute towards the kth order derivative at the left (right) end point of I. In particular, we have

$$\begin{aligned} &f(x_1) = f_1, \quad \frac{df}{dx}(x_1) = \frac{p}{\xi_{p+2}-x_1} \frac{w_2}{w_1}(f_2 - f_1), \\ &f(x_2) = f_n, \quad \frac{df}{dx}(x_2) = \frac{p}{x_2-\xi_n} \frac{w_{n-1}}{w_n}(f_n - f_{n-1}). \end{aligned} \tag{4}$$

A NURBS curve embedded in $\mathbb{R}^d$, $d \geq 2$, can be constructed as

$$\boldsymbol{f}(x) = \sum_{j=1}^n \boldsymbol{f}_j b^w_{j,p}(x),$$

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

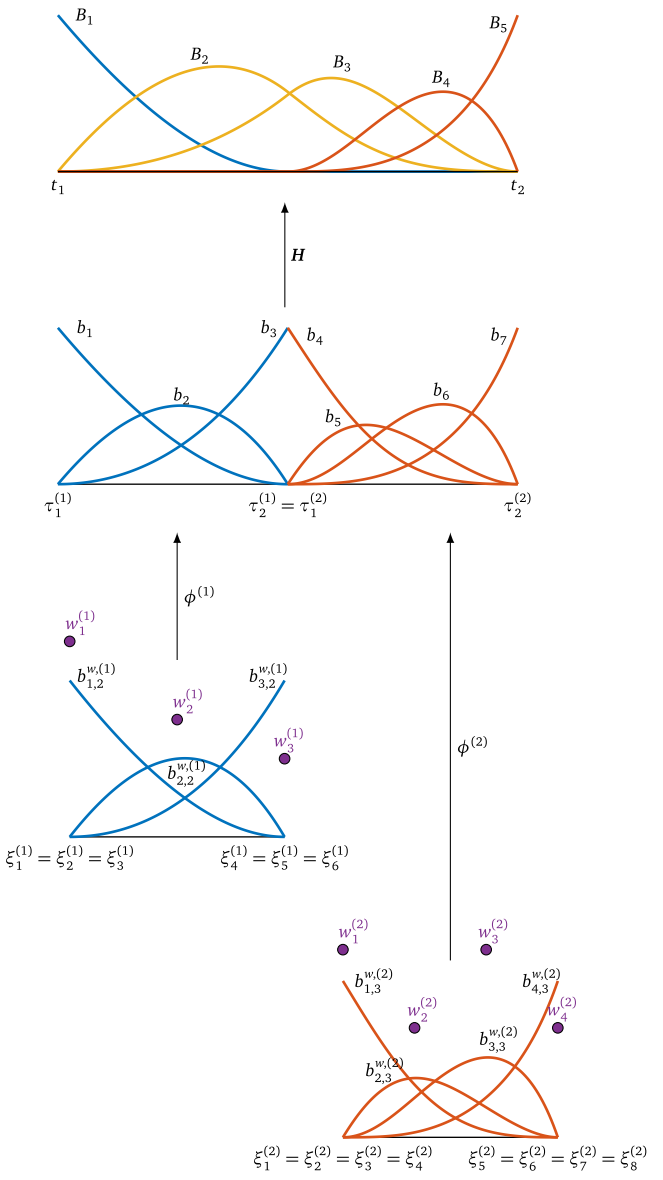


Fig. 1. A visual illustration of the notation and the construction of rational multi-degree B-splines as described in Section 2.2. Here, the quadratic (blue) and cubic (red) NURBS shown at the bottom are used to build the C^1 multi-degree B-splines shown at the top. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.2. Rational multi-degree B-splines

Consider m open knot vectors $\boldsymbol{\xi}^{(i)}$ of degree $p^{(i)}$, $i = 1, \ldots, m$, defined as in Eq. (2). We denote the left and right end points of the interval $I^{(i)}$ associated to $\boldsymbol{\xi}^{(i)}$ with $x_1^{(i)}$ and $x_2^{(i)}$, respectively. The collection $\boldsymbol{\Xi} := (\boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(m)})$ is called an *m-segment knot vector configuration*. The multi-degree spline spaces will be constructed by considering spline spaces over the knot vectors $\boldsymbol{\xi}^{(i)}$, which are glued together with certain smoothness requirements at the end points $x_2^{(i)}$ and $x_1^{(i+1)}$ for $i \in \{1, 2, \ldots, m-1\}$. The equivalence class at the points $x_2^{(i)}$ and $x_1^{(i+1)}$ is called the *ith segment join*. We define the mapping $\phi^{(i)}$ for each segment $i = 1, \ldots, m$,

$$\phi^{(i)}(x) := x - x_1^{(i)} + \tau_1^{(1)} + \sum_{\ell=1}^{i-1} (x_2^{(\ell)} - x_1^{(\ell)}), \tag{5}$$

for an arbitrarily chosen origin $\tau_1^{(1)} \in \mathbb{R}$. Then, $\Omega^{(i)} := [\tau_1^{(i)}, \tau_2^{(i)}] := \phi^{(i)}([x_1^{(i)}, x_2^{(i)}]) \subset \mathbb{R}$, and we construct the composed interval

$$\Omega := [t_1, t_2] := \Omega^{(1)} \cup \cdots \cup \Omega^{(m)}.$$

Note that $\tau_2^{(i)} = \tau_1^{(i+1)}$, $i \in \{1, \ldots, m-1\}$ and $t_1 = \tau_1^{(1)}$. Moreover, let $\boldsymbol{W} := (\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(m)})$ be a sequence of weight vectors defined as in Eq. (3). We refer the reader to Fig. 1 for a visual illustration of the notation of the above concepts, in case $m = 2$ and $p^{(1)} = 2$, $p^{(2)} = 3$.

The space of rational multi-degree splines is defined as

$$\mathscr{R}[\boldsymbol{\Xi}, \boldsymbol{W}] := \left\{ f \in C^1(\Omega) : f \circ \phi^{(i)} \in \mathscr{R}[\boldsymbol{\xi}^{(i)}, \boldsymbol{w}^{(i)}],\ 1 \le i \le m \right\},$$

and the periodic space of rational multi-degree splines as

$$\mathscr{R}^{per}[\boldsymbol{\Xi}, \boldsymbol{W}] := \left\{ f \in \mathscr{R}[\boldsymbol{\Xi}, \boldsymbol{W}] : f(t_1) = f(t_2), \ \frac{df}{dt}(t_1) = \frac{df}{dt}(t_2) \right\}.$$

The elements of $\mathscr{R}[\boldsymbol{\Xi}, \boldsymbol{W}]$ and $\mathscr{R}^{per}[\boldsymbol{\Xi}, \boldsymbol{W}]$ are piecewise-NURBS functions such that the pieces meet with C^1 continuity at each segment join. It is clear that classical NURBS spaces are a special case of the rational multi-degree spline spaces.

In the following, we build a suitable basis for the spaces $\mathscr{R}[\boldsymbol{\Xi}, \boldsymbol{W}]$ and $\mathscr{R}^{per}[\boldsymbol{\Xi}, \boldsymbol{W}]$. On the ith knot vector $\boldsymbol{\xi}^{(i)}$, we have $n^{(i)}$ NURBS $b_{j,p^{(i)}}^{w,(i)}$ of degree $p^{(i)}$ that span the spline space $\mathscr{R}[\boldsymbol{\xi}^{(i)}, \boldsymbol{w}^{(i)}]$. In the first step, we map these basis functions from $I^{(i)}$ to $\Omega^{(i)}$ using $\phi^{(i)}$ in Eq. (5), and extend them on the entire interval Ω by defining them to be zero outside $\Omega^{(i)}$. More precisely, specifying the cumulative local dimensions μ_i for $i = 0, \ldots, m$,

$$\mu_0 := 0, \quad \mu_i := \sum_{\ell=1}^{i} n^{(\ell)} = \mu_{i-1} + n^{(i)}, \quad i > 0,$$

we define for $i = 1, \ldots, m$ and $j = 1, \ldots, n^{(i)}$,

$$b_{\mu_{i-1}+j}(t) := \begin{cases} b_{j,p^{(i)}}^{w,(i)}(x), & \text{if } [\tau_1^{(i)}, \tau_2^{(i)}) \ni t = \phi^{(i)}(x), \\ b_{j,p^{(i)}}^{w,(i)}(x_2^{(i)}), & \text{if } i = m \text{ and } t = t_2, \\ 0, & \text{otherwise.} \end{cases}$$

For the sake of simplicity, we dropped the reference to the (local) degree and weight in the notation. From the properties of NURBS, it is clear that the functions $b_1, \ldots, b_{\mu_m}$ are linearly independent and form a non-negative partition of unity on Ω. We arrange these basis functions in a column vector $\boldsymbol{b}$ of length μ_m. We refer the reader again to Fig. 1 for a visual illustration of the notation of the above concepts.

Now, we construct extraction matrices $\boldsymbol{H}$ and $\boldsymbol{H}^{per}$ such that the functions in $\{B_i : i = 1, \ldots, n\}$ and $\{B_i^{per} : i = 1, \ldots, n^{per}\}$, defined by

$$\boldsymbol{B} := \boldsymbol{H}\boldsymbol{b}, \quad \boldsymbol{B}^{per} := \boldsymbol{H}^{per}\boldsymbol{b}, \tag{6}$$

span $\mathscr{R}[\boldsymbol{\Xi}, \boldsymbol{W}]$ and $\mathscr{R}^{per}[\boldsymbol{\Xi}, \boldsymbol{W}]$, respectively. The key here, and the reason our approach can be efficiently implemented by design, is that these extraction matrices can be explicitly specified. To this end, we define counters η_i for $i = 0, \ldots, m$,

$$\eta_0 := 0, \quad \eta_i := \sum_{\ell=1}^{i} (n^{(\ell)} - 2) = \mu_i - 2i, \quad i > 0,$$

and parameters $\alpha^{(i)}$ and $\beta^{(i)}$ for $i = 1, \ldots, m-1$,

$$\alpha^{(i)} := \frac{p^{(i)}}{x_2^{(i)} - \xi_{n^{(i)}}^{(i)}} \frac{w_{n^{(i)}-1}^{(i)}}{w_{n^{(i)}}^{(i)}} > 0,$$

$$\beta^{(i)} := \frac{p^{(i+1)}}{\xi_{p^{(i+1)}+2}^{(i+1)} - x_1^{(i+1)}} \frac{w_2^{(i+1)}}{w_1^{(i+1)}} > 0.$$

In the periodic setting, $\alpha^{(m)}$ and $\beta^{(m)}$ are computed using the above equations by identifying the index $i+1$ with 1. Recall Eq. (4) to see the motivation behind the definition of the above parameters. Then, we define a common sparse matrix $\boldsymbol{H}^c$ of size $\eta_m \times (\mu_m - 2)$, whose non-zero entries H^c_{ij} are identified as follows: for $i = 1, \ldots, m$ and $j = 1, \ldots, n^{(i)} - 2$,

$$H^c_{\eta_{i-1}+j,\mu_{i-1}+j} := 1, \tag{7}$$

and for $i = 1, \ldots, m-1$,

$$\begin{aligned} H^c_{\eta_i,\mu_i-1} &:= H^c_{\eta_i,\mu_i} := \frac{\alpha^{(i)}}{\alpha^{(i)}+\beta^{(i)}}, \\ H^c_{\eta_i+1,\mu_i-1} &:= H^c_{\eta_i+1,\mu_i} := \frac{\beta^{(i)}}{\alpha^{(i)}+\beta^{(i)}}. \end{aligned} \tag{8}$$

The desired extraction matrices in Eq. (6) are then specified as follows:

$$\boldsymbol{H} := \left[\begin{array}{c|c|c} 1 & \boldsymbol{0} & 0 \\ \hline \boldsymbol{0} & \boldsymbol{H}^c & \boldsymbol{0} \\ \hline 0 & \boldsymbol{0} & 1 \end{array}\right],$$

$$\boldsymbol{H}^{per} := \left[\begin{array}{c|c|c} \frac{\beta^{(m)}}{\alpha^{(m)}+\beta^{(m)}} & & \frac{\beta^{(m)}}{\alpha^{(m)}+\beta^{(m)}} \\ \boldsymbol{0} & \boldsymbol{H}^c & \boldsymbol{0} \\ \frac{\alpha^{(m)}}{\alpha^{(m)}+\beta^{(m)}} & & \frac{\alpha^{(m)}}{\alpha^{(m)}+\beta^{(m)}} \end{array}\right]. \tag{9}$$

The number of rows in the two matrices are denoted with $n := \eta_m + 2$ and $n^{per} := \eta_m$, respectively. The sparse and simple structure of both matrices means that it is easy to verify that both have full rank. Indeed, this conclusion can be directly deduced from the full rank of $\boldsymbol{H}^c$, which in turn is implied by Eq. (7). Moreover, their entries are non-negative, and the column sum is equal to one. Hence, we conclude that these matrices are DTA-compatible.

How these matrices help us build C^1 splines can be understood by taking into account Eq. (4) at any end point $\tau_2^{(i)}$. Let us discuss the non-periodic setting and fix $i \in \{1, \ldots, m-1\}$; the argument can be directly applied to the periodic setting as well. Only four functions $(b_{\mu_i-1}, b_{\mu_i}, b_{\mu_i+1}, b_{\mu_i+2})$ are not C^1 at $\tau_2^{(i)}$. More precisely, only these four functions have non-vanishing values and first derivatives here. In view of (4), a spline f given by

$$f(t) = \sum_{j=1}^{4} f_j b_{\mu_i-2+j}(t)$$

will be C^1 at $\tau_2^{(i)}$ if

$$f_2 = f_3, \quad \alpha^{(i)}(f_2 - f_1) = \beta^{(i)}(f_4 - f_3). \tag{10}$$

We can verify that the entries of $\boldsymbol{H}$ satisfy exactly such relations. Indeed, for some j, the matrix $\boldsymbol{H}$ defines two new functions B_j and B_{j+1} such that

$$\begin{bmatrix} B_j \\ B_{j+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \frac{\alpha^{(i)}}{\alpha^{(i)}+\beta^{(i)}} & \frac{\alpha^{(i)}}{\alpha^{(i)}+\beta^{(i)}} & 0 \\ 0 & \frac{\beta^{(i)}}{\alpha^{(i)}+\beta^{(i)}} & \frac{\beta^{(i)}}{\alpha^{(i)}+\beta^{(i)}} & 1 \end{bmatrix}}_{=:\bar{\boldsymbol{H}}^{(i)}} \begin{bmatrix} b_{\mu_i-1} \\ b_{\mu_i} \\ b_{\mu_i+1} \\ b_{\mu_i+2} \end{bmatrix} + \begin{bmatrix} \bar{B}_j^{(i)} \\ \bar{B}_{j+1}^{(i)} \end{bmatrix},$$

where $\bar{B}_j^{(i)}$ and $\bar{B}_{j+1}^{(i)}$ are at least C^1 at $\tau_2^{(i)}$. When setting $[f_1, f_2, f_3, f_4]$ equal to the first or the second row of $\bar{\boldsymbol{H}}^{(i)}$, we see that Eq. (10) is satisfied. The following result follows from the above discussion.

Theorem 2.2. *The C^1 smooth piecewise-rational functions in the sets $\{B_i : i = 1, \ldots, n\}$ and $\{B_i^{per} : i = 1, \ldots, n^{per}\}$ are linearly independent, locally supported, and form a convex partition of unity on Ω.*

Remark 2.3. In [2, Section 2.3.5] it was observed that the C^1 smooth piecewise-rational basis functions enjoy the properties described in Theorem 2.2. However, a formal proof was missing. It was also pointed out that the property of non-negativity is in general not present in case of C^2 or higher smoothness. On the other hand, this is possible when restricting to polynomial pieces [6].

Remark 2.4. With the aim of designing quadric curves, also called conics, it is natural to choose local NURBS spaces of the same degree p and defined on the same uniform knot vector $\boldsymbol{\xi}$. Moreover, it is common to set $w_1^{(i)} = w_{n^{(i)}}^{(i)} = 1$. Under these circumstances, the ratios in Eq. (8) read as

$$\begin{aligned} \frac{\alpha^{(i)}}{\alpha^{(i)}+\beta^{(i)}} &= \frac{w_{n^{(i)}-1}^{(i)}}{w_{n^{(i)}-1}^{(i)} + w_2^{(i+1)}}, \\ \frac{\beta^{(i)}}{\alpha^{(i)}+\beta^{(i)}} &= \frac{w_2^{(i+1)}}{w_{n^{(i)}-1}^{(i)} + w_2^{(i+1)}}. \end{aligned}$$

Finally, if there is additional symmetry in the choice of weights, so $w_{n^{(i)}-1}^{(i)} = w_2^{(i+1)}$, we simply get

$$\frac{\alpha^{(i)}}{\alpha^{(i)}+\beta^{(i)}} = \frac{1}{2}, \quad \frac{\beta^{(i)}}{\alpha^{(i)}+\beta^{(i)}} = \frac{1}{2}.$$

Once we have computed a DTA-compatible extraction matrix $\boldsymbol{H}$ (or $\boldsymbol{H}^{per}$), given n control points $\boldsymbol{f}_i \in \mathbb{R}^d$, $d \geq 2$, we can construct a piecewise-rational curve $\boldsymbol{f}$ embedded in $\mathbb{R}^d$,

$$\boldsymbol{f}(t) = \sum_{i=1}^{n} \boldsymbol{f}_i B_i(t).$$

For a fixed curve, the transpose of $\boldsymbol{H}$ (or $\boldsymbol{H}^{per}$) defines the relationship between control points of the b_j (discontinuous at the segment joins) and control points of the smooth B_i. More precisely, if

$$\sum_{j=1}^{\mu_m} \boldsymbol{g}_j b_j(t) = \boldsymbol{f}(t) = \sum_{i=1}^{n} \boldsymbol{f}_i B_i(t),$$

then

$$\boldsymbol{g}_j = \sum_{i=1}^{n} H_{ij} \boldsymbol{f}_i. \tag{11}$$

Remark 2.5. When dealing with curves, the proposed piecewise-NURBS framework can also be formulated in the context of geometric continuity [27]. In such case, the C^1 smoothness condition at the segment join in (10) is replaced by the G^1 smoothness condition

$$f_2 = f_3, \quad \alpha^{(i)}(f_2 - f_1) = \gamma^{(i)}\beta^{(i)}(f_4 - f_3),$$

for a given geometric shape parameter $\gamma^{(i)} > 0$, resulting in the matrix

$$\bar{\boldsymbol{H}}^{(i)} = \begin{bmatrix} 1 & \frac{\alpha^{(i)}}{\alpha^{(i)}+\gamma^{(i)}\beta^{(i)}} & \frac{\alpha^{(i)}}{\alpha^{(i)}+\gamma^{(i)}\beta^{(i)}} & 0 \\ 0 & \frac{\gamma^{(i)}\beta^{(i)}}{\alpha^{(i)}+\gamma^{(i)}\beta^{(i)}} & \frac{\gamma^{(i)}\beta^{(i)}}{\alpha^{(i)}+\gamma^{(i)}\beta^{(i)}} & 1 \end{bmatrix}.$$

It is clear that this matrix is still DTA-compatible.

2.3. Refinement of piecewise-rational curves

The rational spline spaces defined in the previous section can be refined in a multitude of ways. We could reduce the smoothness at segment joins, raise the polynomial degrees of local NURBS spaces, and/or insert new knots in local NURBS

spaces [2, Section 2.4.3]. A combination of these possibilities could be judiciously employed to achieve spline spaces that provide higher resolution or approximation power exactly where needed. In this section, we present an explicit construction of refined representations of a given piecewise-rational curve.

Before delving into the details of the refinement procedure, we first define two matrices $\boldsymbol{G}$ and $\boldsymbol{G}^{per}$ that can be regarded as right inverses of the extraction matrices $\boldsymbol{H}$ and $\boldsymbol{H}^{per}$, respectively. Looking at the structure of the matrix $\boldsymbol{H}^c$ specified in Eqs. (7)–(8), we can define a sparse matrix $\boldsymbol{G}^c$ of size $(\mu_m - 2) \times \eta_m$, whose non-zero entries G^c_{ij} are identified as follows: for $i = 1, \ldots, m$ and $j = 1, \ldots, n^{(i)} - 2$,

$$G^c_{\mu_{i-1}+j,\eta_{i-1}+j} := 1.$$

From its construction it is clear that the product $\boldsymbol{H}^c\boldsymbol{G}^c$ is equal to the identity matrix. Similarly, keeping in mind Eq. (9), the matrices

$$\boldsymbol{G} := \begin{bmatrix} 1 & \boldsymbol{0} & 0 \\ \hline \boldsymbol{0} & \boldsymbol{G}^c & \boldsymbol{0} \\ \hline 0 & \boldsymbol{0} & 1 \end{bmatrix}, \quad \boldsymbol{G}^{per} := \begin{bmatrix} \boldsymbol{0} \\ \hline \boldsymbol{G}^c \\ \hline \boldsymbol{0} \end{bmatrix} \tag{12}$$

give rise to products $\boldsymbol{H}\boldsymbol{G}$ and $\boldsymbol{H}^{per}\boldsymbol{G}^{per}$ that are equal to identity matrices.

Now, let $\mathscr{R}$ be a given spline space and let us denote the target refined space with $\tilde{\mathscr{R}}$. For simplicity of notation, we drop the superscript *per* in case of periodicity. Then, we consider the two unique representations of a curve $\boldsymbol{f}$ with coordinate functions in $\mathscr{R} \subset \tilde{\mathscr{R}}$,

$$\sum_{i=1}^{\tilde{n}} \tilde{\boldsymbol{f}}_i \tilde{B}_i(t) = \boldsymbol{f}(t) = \sum_{i=1}^{n} \boldsymbol{f}_i B_i(t). \tag{13}$$

Let us collect the control points in the row vectors $\tilde{\boldsymbol{F}} := [\tilde{\boldsymbol{f}}_1, \ldots, \tilde{\boldsymbol{f}}_{\tilde{n}}]$ and $\boldsymbol{F} := [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n]$. We now seek the refinement matrix $\boldsymbol{R}$ of size $n \times \tilde{n}$ that helps us compute $\tilde{\boldsymbol{F}}$ from $\boldsymbol{F}$, i.e.,

$$\tilde{\boldsymbol{F}} = \boldsymbol{F}\boldsymbol{R}.$$

Assume that $\boldsymbol{H}$ and $\tilde{\boldsymbol{H}}$ are the extraction matrices corresponding to the spaces $\mathscr{R}$ and $\tilde{\mathscr{R}}$, respectively. Incorporating these matrices in the representations in Eq. (13) results in

$$\tilde{\boldsymbol{F}}\tilde{\boldsymbol{H}}\tilde{\boldsymbol{b}} = \boldsymbol{F}\boldsymbol{H}\boldsymbol{b},$$

where the column vectors $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ collect the local NURBS basis functions. Let the matrix $\boldsymbol{S}$ be such that $\boldsymbol{b} = \boldsymbol{S}\tilde{\boldsymbol{b}}$; this matrix can be computed with standard NURBS refinement techniques. Then, we have

$$\tilde{\boldsymbol{F}}\tilde{\boldsymbol{H}}\tilde{\boldsymbol{b}} = \boldsymbol{F}\boldsymbol{H}\boldsymbol{S}\tilde{\boldsymbol{b}}.$$

This implies that we can compute $\boldsymbol{R}$ by solving the following (overdetermined) linear system with a unique solution,

$$\boldsymbol{R}\tilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{S}.$$

After multiplication of both sides of this system with the matrix $\tilde{\boldsymbol{G}}$ (corresponding to $\tilde{\boldsymbol{H}}$) as defined in Eq. (12), we arrive at

$$\boldsymbol{R} = \boldsymbol{H}\boldsymbol{S}\tilde{\boldsymbol{G}}. \tag{14}$$

Note that the application of $\tilde{\boldsymbol{G}}$ in Eq. (14) means that a subset of $\tilde{n}$ columns of $\boldsymbol{H}\boldsymbol{S}$ are selected to form $\boldsymbol{R}$.

Remark 2.6. The definition of $\tilde{\boldsymbol{G}}$ is done for the sake of simplicity of computation of $\boldsymbol{R}$ in Eq. (14), but is not unique. Any matrix that is a right inverse of $\tilde{\boldsymbol{H}}$ would be a valid choice as well, such as the standard Moore–Penrose right inverse $\tilde{\boldsymbol{H}}^T(\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^T)^{-1}$.

2.4. Circles and ellipses

We now present the general construction of ellipses (and as a special case also circles) using the C^1 rational splines introduced thus far. We present three approaches for doing so using splines of low(est) degree, i.e., C^1 splines of quadratic degree, cubic degree and mixed quadratic/cubic multi-degree. All approaches will construct four C^1 piecewise-NURBS functions B_i and associated control points $\boldsymbol{f}_i$, $i = 1, \ldots, 4$, such that the curve $\boldsymbol{f}$,

$$\boldsymbol{f}(t) := (f_x(t), f_y(t)) := \sum_{i=1}^{4} \boldsymbol{f}_i B_i(t),$$

describes the exact ellipse centered at $(0, 0)$ and with axis lengths (a_x, a_y),

$$\left(\frac{f_x}{a_x}\right)^2 + \left(\frac{f_y}{a_y}\right)^2 = 1. \tag{15}$$

Since the splines B_i form a partition of unity, these ellipses can be affinely transformed by directly applying the transformation to the control points $\boldsymbol{f}_i$. Subdivided or higher-degree representations can be easily obtained by refining the representations provided here (see Section 2.3).

To visually illustrate the smoothness of $\boldsymbol{f}$, we will also show the curve $\tilde{\boldsymbol{f}}$ obtained by perturbing one of the control points. Since all B_i are smooth, the perturbed curve will also be smooth. For uniformity throughout the examples, we will choose the control points of the perturbed curve as

$$\tilde{\boldsymbol{f}}_i := \begin{cases} \boldsymbol{f}_i + (0, a_y), & i = 1, \\ \boldsymbol{f}_i, & i = 2, 3, 4. \end{cases} \tag{16}$$

2.4.1. C^1 description of degree 2

Here we present a C^1 quadratic description of the ellipse in Eq. (15) using 4 rational pieces. Consider the domain $\Omega = [0, 4]$ in the periodic setting. Choose $m = 4$ and

$$\xi^{(i)} = [0, 0, 0, 1, 1, 1], \quad \boldsymbol{w}^{(i)} = \left[1, \tfrac{\sqrt{2}}{2}, 1\right], \quad i = 1, \ldots, 4.$$

We can define four C^1 quadratic piecewise-NURBS functions B_i on Ω using the extraction matrix

$$\boldsymbol{H}^{per} = \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} \end{bmatrix}; \tag{17}$$

see Eq. (9) taking into account Remark 2.4. These spline basis functions are shown in Fig. 2(a, top row). Finally, we can build an ellipse centered at $(0, 0)$ and with axis lengths (a_x, a_y) by combining the splines B_i with the control points $\boldsymbol{f}_i$ defined as

$$\boldsymbol{f}_1 = (a_x, a_y) = -\boldsymbol{f}_3, \quad \boldsymbol{f}_2 = (a_x, -a_y) = -\boldsymbol{f}_4.$$

Remark 2.7. To verify that the curve $\boldsymbol{f}$ satisfies Eq. (15), we can proceed as follows. The simplest approach is to numerically evaluate $\boldsymbol{f}(t)$ at all t and plug the result in that equation. Alternatively, this verification can also be performed analytically by looking at the explicit expressions of the rational pieces that form $\boldsymbol{f}$. For instance, consider the first quadratic rational piece, $\boldsymbol{g}^{(1)}$, that is a part of $\boldsymbol{f}$. As discussed in Eq. (11), we can get the control points of this piece, denoted with $\boldsymbol{g}^{(1)}_j$, $j \in \{1, 2, 3\}$, by applying the transpose of (a submatrix of) $\boldsymbol{H}^{per}$ from Eq. (17) to a vector containing the points $\boldsymbol{f}_i$, i.e.,

$$\boldsymbol{g}^{(1)}_j = \sum_{i=1}^{4} H^{per}_{ij} \boldsymbol{f}_i.$$

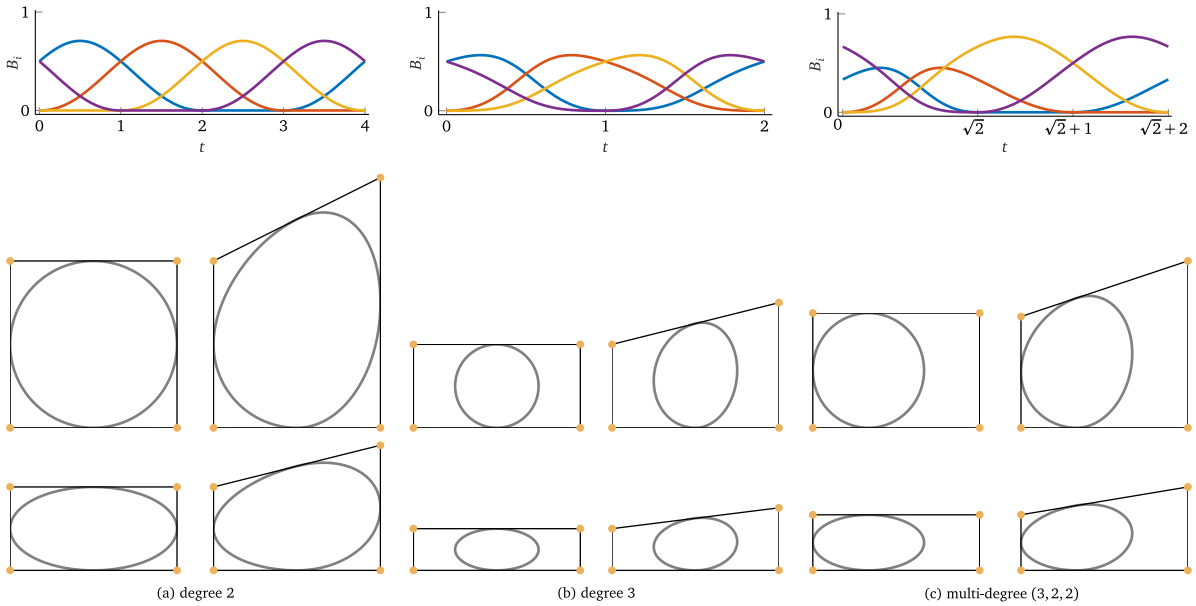


Fig. 2. Different C^1 smooth descriptions of unit circles and ellipses with axis lengths $(1, \frac{1}{2})$. The conics in figure boxes (a)–(c) are made up of C^1 splines of degree 2, degree 3 and multi-degree $(3, 2, 2)$, respectively. In each box, the middle and bottom rows of figures show the C^1 circles and ellipses built using the C^1 B-spline functions shown at the top; the t-axis markers correspond to the breakpoint locations. Furthermore, in each of the bottom two rows of figures, the curve on the left shows the exact conic, and the curve on the right is obtained by raising one of the control points of the exact conic; see Examples 2.8, 2.9 and 2.11 for details. (Even though all circles and ellipses have the same respective dimensions, they are scaled differently only to accommodate their control nets into the figure.)

This yields the control points

$$\boldsymbol{g}_1^{(1)} = (0, a_y), \quad \boldsymbol{g}_2^{(1)} = (a_x, a_y), \quad \boldsymbol{g}_3^{(1)} = (a_x, 0).$$

Combining the above control points with the NURBS basis defined on the first segment,

$$b_{1,2}^w(t) = \frac{(1-t)^2}{w(t)}, \quad b_{2,2}^w(t) = \frac{\sqrt{2}t(1-t)}{w(t)}, \quad b_{3,2}^w(t) = \frac{t^2}{w(t)},$$

where $w(t) := (1-t)^2 + \sqrt{2}t(1-t) + t^2$, some simple algebra shows that $\boldsymbol{g}^{(1)}$ indeed satisfies Eq. (15). Verifications for the other pieces of $\boldsymbol{f}$ can be similarly done.

Example 2.8. Choosing $a_x = a_y = 1$, we obtain a circle of radius 1, as shown in Fig. 2(a, middle row). This C^1 quadratic description is equivalent to the one used in [12]. The choice $a_x = 2a_y = 1$ yields an ellipse with axis lengths $(1, \frac{1}{2})$, as shown in Fig. 2(a, bottom row). The perturbed versions of these conics, with the control points chosen as in Eq. (16), are shown as well and they remain clearly smooth. ♦

2.4.2. C^1 description of degree 3

Here we present a C^1 cubic representation of the ellipse in Eq. (15) that uses only 2 rational pieces. Consider the domain $\Omega = [0, 2]$ in the periodic setting. Choose $m = 2$ and

$$\boldsymbol{\xi}^{(i)} = [0, 0, 0, 0, 1, 1, 1, 1], \quad \boldsymbol{w}^{(i)} = \left[1, \tfrac{1}{3}, \tfrac{1}{3}, 1\right], \quad i = 1, 2.$$

We can define four C^1 cubic piecewise-NURBS functions B_i on Ω using the extraction matrix

$$\boldsymbol{H}^{per} = \begin{bmatrix} \frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{2} \end{bmatrix}. \tag{18}$$

These basis functions are shown in Fig. 2(b, top row). Choosing the associated control points $\boldsymbol{f}_i$ as

$$\boldsymbol{f}_1 = (2a_x, a_y) = -\boldsymbol{f}_3, \quad \boldsymbol{f}_2 = (2a_x, -a_y) = -\boldsymbol{f}_4,$$

we get a C^1 cubic description of an ellipse centered at $(0, 0)$ with axis lengths (a_x, a_y). This can be verified in the vein of Remark 2.7.

Example 2.9. Choosing $a_x = a_y = 1$, we obtain a circle of radius 1, as shown in Fig. 2(b, middle row). This C^1 cubic description is equivalent to the one used in [2]. The choice $a_x = 2a_y = 1$ yields an ellipse with axis lengths $(1, \frac{1}{2})$, as shown in Fig. 2(b, bottom row). The perturbed versions of these conics, with the control points chosen as in Eq. (16), are shown as well and they remain clearly smooth. It can be observed that, compared to the description from Section 2.4.1, the control points here are at a greater distance from the curve. This is completely analogous to the behavior of classical NURBS. ♦

2.4.3. C^1 description of multi-degree $(3, 2, 2)$

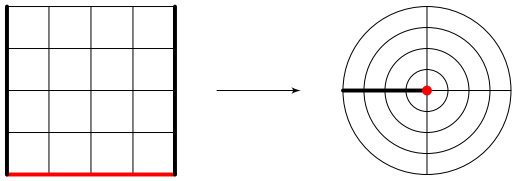
Many, many different C^1 low-degree representations of the circle can be cooked up. Instead of attempting the impossible task of presenting them all, we present a single example that uses our splines in a more general setting than the above two descriptions. Consider the domain $\Omega = [0, \sqrt{2} + 2]$ in the periodic setting. Choose $m = 3$ and

$$\boldsymbol{\xi}^{(1)} = [0, 0, 0, 0, \sqrt{2}, \sqrt{2}, \sqrt{2}, \sqrt{2}], \quad \boldsymbol{w}^{(1)} = \left[1, \tfrac{1}{3}, \tfrac{1}{3}, 1\right],$$
$$\boldsymbol{\xi}^{(2)} = \boldsymbol{\xi}^{(3)} = [0, 0, 0, 1, 1, 1], \quad \boldsymbol{w}^{(2)} = \boldsymbol{w}^{(3)} = \left[1, \tfrac{\sqrt{2}}{2}, 1\right].$$

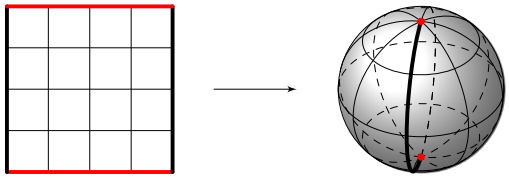
Remark 2.10. Opting for geometric continuity (see Remark 2.5) would have allowed the choice of Ω with integral length, but the present setting is sufficient for illustrative purposes.

Then, we can define four C^1 multi-degree piecewise-NURBS functions B_i on Ω using the extraction matrix

$$\boldsymbol{H}^{per} = \begin{bmatrix} \frac{1}{3} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 1 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{3} & \frac{2}{3} & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{2}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & \frac{2}{3} \end{bmatrix}.$$



(a) Single polar singularity



(b) Double polar singularity

Fig. 3. A single edge or a pair of opposite edges of a tensor-product spline patch can be collapsed for creating geometries with polar singularities. The collapsed edges here are shown in red, and the black edges are identified with each other to enforce periodicity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

These basis functions are shown in Fig. 2(c, top row). Choosing the associated control points $\boldsymbol{f}_i$ as

$$\boldsymbol{f}_1 = (2a_x, a_y), \quad \boldsymbol{f}_2 = (2a_x, -a_y),$$
$$\boldsymbol{f}_3 = (-a_x, -a_y), \quad \boldsymbol{f}_4 = (-a_x, a_y),$$

we get a C^1 multi-degree description of an ellipse centered at $(0, 0)$ with axis lengths (a_x, a_y). This can be verified in the vein of Remark 2.7.

Example 2.11. Choosing $a_x = a_y = 1$, we obtain a circle of radius 1, as shown in Fig. 2(c, middle row). The choice $a_x = 2a_y = 1$ yields an ellipse with axis lengths $(1, \frac{1}{2})$, as shown in Fig. 2(c, bottom row). The perturbed versions of these conics, with the control points chosen as in Eq. (16), are shown as well and they remain clearly smooth. Once again, compared to the description from Section 2.4.1, the control points here lie at a greater distance from the cubic portion of the curve. ♦

3. Piecewise-rational polar surfaces

In this section, we describe how to construct C^1 smooth representations for *polar surfaces* containing single or double polar singularities (e.g., hemispheres and spheres, respectively). Such surfaces can be obtained by starting from a bivariate tensor-product (piecewise-NURBS) spline patch and collapsing one or two of its edges, respectively, as illustrated in Fig. 3. Each of such edge collapses creates a *polar point* and can be achieved by coalescing the control points related to basis functions with non-zero values on the edge. In general, however, this control-point coalescing will introduce kinks at the poles and the surface representation will not be smooth. To achieve overall smoothness, additional conditions need to be satisfied by the control points [2, Section 3]. In Section 3.1, we derive C^1 smoothness conditions at a polar point, and they enable us to build *smooth polar splines* as linear combinations of bivariate tensor-product splines in Section 3.2. Then, in Section 3.3, we give an explicit procedure how to compute a refined representation of a given polar surface. Finally, in Section 3.4, we present explicit descriptions of arbitrary ellipsoids using C^1 smooth low-degree polar spline representations suited for integrated design and analysis.

3.1. Smoothness conditions at the polar points

A polar surface will be smooth at a polar point if it can be locally (re)parameterized in a smooth way. Such parameterizations can be specified in a constructive manner and we elaborate upon it in this section. The resulting conditions will help us build smooth polar B-splines in the next section.

As shown in Fig. 3, we first describe the initial setup — a tensor-product spline space on a rectangular domain. We start from two univariate C^1 rational spline spaces $\mathscr{R}^s, \mathscr{R}^t$ defined on the univariate domains $\Omega^s := [s_1, s_2]$ and $\Omega^t := [t_1, t_2]$, respectively; the superscripts of s and t are meant to indicate the symbols used for the respective coordinates. Using a Cartesian product, we build the rectangular domain $\Omega := \Omega^s \times \Omega^t$, and on Ω we define the tensor-product spline space $\mathscr{R} := \mathscr{R}^s \otimes \mathscr{R}^t$. Without loss of generality, we assume that $s_1 = t_1 = 0$. This tensor-product spline space is spanned by tensor-product B-spline basis functions B_{ij}, $i = 1, \ldots, n^s$; $j = 1, \ldots, n^t$. Here, n^s and n^t denote the respective dimensions of the chosen univariate spline spaces; the basis functions spanning these spaces are denoted with B_i^s and B_j^t. Then, the tensor-product basis function B_{ij} is simply the product $B_i^s \otimes B_j^t$. The functions B_{ij} are assumed to be periodic in s and non-periodic in t.

Now, let us use the functions B_{ij} to map the domain Ω to a polar surface using edge-collapse. Then, the smoothness conditions at a collapsed edge will only involve those B_{ij} that have non-zero first derivatives there. Observe that, if $n^t \geq 4$, then any B_{ij} with non-zero first derivatives at the bottom edge of Ω will have zero first derivatives at its top edge, and vice versa. The upshot is that, when we are collapsing both the bottom and top edges of Ω into two polar points, as in Fig. 3(b), the smoothness conditions at those points are independent of each other and can be resolved separately for $n^t \geq 4$.

3.1.1. Single polar point

In light of the above discussion, in the following we first focus on the case of a single collapsed edge, i.e., the one shown in Fig. 3(a). We derive smoothness conditions that will help us build smooth polar spline functions (and, using them, smooth polar surfaces). This is done by explicitly specifying the parameterization with respect to which the spline functions are deemed smooth. First, we construct a planar disk-like domain Ω^{pol}, called the *polar parametric domain*, via a suitable polar map $\boldsymbol{F}$; see Fig. 3(a). Next, for an arbitrary C^1 spline $f \in \mathscr{R}$, we define $f^{pol} : \Omega^{pol} \to \mathbb{R}$ to be $f \circ \boldsymbol{F}^{-1}$. In general, f^{pol} will be multivalued at the pole. Finally, we derive the required smoothness conditions by asking for f^{pol} to be C^1 smooth at the polar point.

We start by building the map $\boldsymbol{F}$. Assign the control point $\boldsymbol{F}_{ij} := (\rho_j \cos(\theta_i), \rho_j \sin(\theta_i)) \in \mathbb{R}^2$ to the basis function B_{ij}, where

$$\rho_j := \frac{j-1}{n^t - 1} \in [0, 1], \tag{19}$$

and

$$\theta_i := 2\pi + \frac{(1-2i)\pi}{n^s} \in [0, 2\pi]. \tag{20}$$

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 the disk-like domain Ω^{pol} with the aid of the map $\boldsymbol{F}$ from Ω to Ω^{pol},

$$\Omega \ni (s, t) \mapsto \boldsymbol{F}(s, t) = (u, v) \in \Omega^{pol},$$

defined as

$$\boldsymbol{F}(s, t) := (F_u(s, t), F_v(s, t)) := \sum_{i=1}^{n^s} \sum_{j=1}^{n^t} \boldsymbol{F}_{ij} B_{ij}(s, t). \tag{21}$$

Note that the above construction will not necessarily yield an exactly circular domain Ω^{pol}; its shape will depend on the choice of $\mathscr{R}$. This domain will serve as the reference element for the smoothness of polar configurations, i.e., we will define polar splines such that they are C^1 smooth functions over Ω^{pol}. It is clear that for all $s \in \Omega^s$,

$$\boldsymbol{F}(s,0) = (0,0), \tag{22}$$

where $(0,0) \in \Omega^{pol}$ is the polar point. Note that this implies

$$\left.\frac{\partial F_u}{\partial s}\right|_{t=0} \equiv 0 \equiv \left.\frac{\partial F_v}{\partial s}\right|_{t=0}. \tag{23}$$

Let B_{ij}^{pol} be the image of B_{ij} under the polar map $\boldsymbol{F} : \Omega \mapsto \Omega^{pol}$ in Eq. (21) so that

$$B_{ij}^{pol}(u,v) = B_{ij}^{pol}(\boldsymbol{F}(s,t)) = B_{ij}(s,t).$$

Then, for given coefficients f_{ij}, a polar spline function f^{pol} over Ω^{pol} can be constructed as

$$f^{pol}(u,v) = \sum_{i=1}^{n^s}\sum_{j=1}^{n^t} f_{ij} B_{ij}^{pol}(u,v).$$

We can pull f^{pol} back to Ω as follows,

$$\begin{aligned} f(s,t) := f^{pol}(\boldsymbol{F}(s,t)) &= \sum_{i=1}^{n^s}\sum_{j=1}^{n^t} f_{ij} B_{ij}^{pol}(\boldsymbol{F}(s,t)) \\ &= \sum_{i=1}^{n^s}\sum_{j=1}^{n^t} f_{ij} B_{ij}(s,t). \end{aligned} \tag{24}$$

Moreover, by using the chain rule we can also relate the partial derivatives of f and f^{pol}:

$$\begin{aligned} \frac{\partial f}{\partial s}(s,t) &= \frac{\partial f^{pol}}{\partial u}(u,v)\frac{\partial F_u}{\partial s}(s,t) + \frac{\partial f^{pol}}{\partial v}(u,v)\frac{\partial F_v}{\partial s}(s,t), \\ \frac{\partial f}{\partial t}(s,t) &= \frac{\partial f^{pol}}{\partial u}(u,v)\frac{\partial F_u}{\partial t}(s,t) + \frac{\partial f^{pol}}{\partial v}(u,v)\frac{\partial F_v}{\partial t}(s,t). \end{aligned}$$

For f^{pol} to be C^1 smooth at the polar point, there must exist real values α, β, γ such that

$$\lim_{(u,v)\to(0,0)} \left[f^{pol},\ \frac{\partial f^{pol}}{\partial u},\ \frac{\partial f^{pol}}{\partial v}\right](u,v) = [\alpha,\ \beta,\ \gamma]. \tag{25}$$

In view of (22) and (23) this means for all $s \in \Omega^s$,

$$f(s,0) = \alpha, \quad \frac{\partial f}{\partial t}(s,0) = \beta\frac{\partial F_u}{\partial t}(s,0) + \gamma\frac{\partial F_v}{\partial t}(s,0).$$

In particular, since only B_{ij}, $j \le 2$, have non-zero values and derivatives when $t = 0$, the above condition translates to the following requirement for all $s \in \Omega^s$,

$$\begin{aligned} &\sum_{i=1}^{n^s} f_{i1} B_{i1}(s,0) = \alpha, \\ &\sum_{i=1}^{n^s}\sum_{j=1}^{2} \left[f_{ij} - \boldsymbol{F}_{ij}\cdot(\beta,\gamma)\right]\frac{\partial B_{ij}}{\partial t}(s,0) = 0. \end{aligned} \tag{26}$$

3.1.2. Double polar point

Eq. (26) shows the required smoothness conditions when the bottom edge of Ω is being collapsed. Next, if we also want to collapse the top edge of Ω, we can repeat the previous argument with minor changes. We would, of course, need to choose a map $\hat{\boldsymbol{F}}$ that collapses the edge $\Omega^s \times \{t_2\}$ instead. One way of achieving this could be by choosing the control points $\hat{\boldsymbol{F}}_{ij} := (\hat{\rho}_j \cos(\hat{\theta}_i), \hat{\rho}_j \sin(\hat{\theta}_i)) \in \mathbb{R}^2$, where

$$\hat{\rho}_j := 1 - \rho_j, \quad \hat{\theta}_i := 2\pi - \theta_i.$$

Then, we can follow the same argument as in Section 3.1.1. Asking for C^1 smoothness of $\hat{f}^{pol}$ is equivalent to asking that there exist real values $\hat{\alpha}, \hat{\beta}, \hat{\gamma}$ such that for all $s \in \Omega^s$,

$$\begin{aligned} &\sum_{i=1}^{n^s} f_{i,n^t} B_{i,n^t}(s,t_2) = \hat{\alpha}, \\ &\sum_{i=1}^{n^s}\sum_{j=n^t-1}^{n^t} \left[f_{ij} - \hat{\boldsymbol{F}}_{ij}\cdot(\hat{\beta},\hat{\gamma})\right]\frac{\partial B_{ij}}{\partial t}(s,t_2) = 0. \end{aligned} \tag{27}$$

Note once again that the smoothness at the polar point corresponding to $t = 0$ is imposed with respect to the parameterization $\boldsymbol{F}(\Omega)$, while that at the polar point corresponding to $t = t_2$ is imposed with respect to the parameterization $\hat{\boldsymbol{F}}(\Omega)$. The corresponding smoothness conditions in Eqs. (26) and (27) involve different coefficients f_{ij} for $n^t \ge 4$, and so can be resolved separately.

Remark 3.1. The choices of $\theta_i, \rho_j, \hat{\theta}_i, \hat{\rho}_j$ are such that the maps $\boldsymbol{F}$ and $\hat{\boldsymbol{F}}$ preserve the orientation of the parametric domain Ω.

3.2. Rational polar B-splines at the polar points

We now elaborate how the derived C^1 smoothness constraints at a polar point will enable the computation of a DTA-compatible extraction matrix. This matrix represents a linear map to a set of polar spline basis functions that are C^1 smooth on the polar parametric domain.

3.2.1. Single polar point

As before, we start by considering the case of a single collapsed edge, i.e., the one shown in Fig. 3(a). Let us arrange the set of basis functions $\{B_{ij}^{pol} : i = 1,\dots,n^s; j = 1,\dots,n^t\}$ in a vector $\boldsymbol{B}$, where B_{ij}^{pol} occupies the $(i+(j-1)n^s)$th entry. Our goal is to construct C^1 smooth polar basis functions on the polar parametric domain Ω^{pol} as suitable linear combinations of the functions B_{ij}^{pol}. In other words, we are looking for an extraction matrix $\boldsymbol{E}$ such that the polar spline basis functions in $\{N_l^{pol} : l = 1,\dots,n\}$ defined by the following relation,

$$\boldsymbol{N} := \boldsymbol{E}\boldsymbol{B}, \tag{28}$$

are C^1 at the polar point. For fixed j, the set $\{B_{ij}^{pol} : i = 1,\dots,n^s\}$ is called the $(j-1)$th polar ring of basis functions. When $j > 2$, all basis functions in the $(j-1)$th ring already satisfy the C^1 continuity conditions at the polar point (their derivatives are identically zero there), so they can be included without modifications in the set of polar basis functions being created. The others will be substituted by three smooth polar basis functions. This dictates that $\boldsymbol{E}$ will be a matrix, with $n := n^s(n^t-2)+3$ rows and $n^s n^t$ columns, taking the following sparse block-diagonal form:

$$\boldsymbol{E} := \begin{bmatrix} \bar{\boldsymbol{E}} & \\ & \boldsymbol{I} \end{bmatrix}, \tag{29}$$

where $\boldsymbol{I}$ is the identity matrix of size $n^s(n^t-2)\times n^s(n^t-2)$ and $\bar{\boldsymbol{E}}$ is a matrix of size $3\times 2n^s$. The entry of $\bar{\boldsymbol{E}}$ corresponding to its lth row and $(i+(j-1)n^s)$th column is denoted with $\bar{E}_{l,(ij)}$. We can then rewrite Eq. (28) as follows for $l = 1, 2, 3$,

$$N_l^{pol}(u,v) = \sum_{i=1}^{n^s}\sum_{j=1}^{2} \bar{E}_{l,(ij)} B_{ij}^{pol}(u,v).$$

We can pull these back to Ω using Eq. (24) to obtain the equivalent representation for $l = 1, 2, 3$,

$$N_l(s,t) = \sum_{i=1}^{n^s} \sum_{j=1}^{2} \bar{E}_{l,(ij)} B_{ij}(s,t). \tag{30}$$

We will enforce C^1 continuity at the polar point by requiring the basis functions N_l^{pol} to satisfy a linearly independent Hermite data set at the polar point, in the spirit of Eq. (25). To this end, we will use three *source basis functions* $\{T_l : l = 1, 2, 3\}$, that provide us with the appropriate Hermite data. Given a non-degenerate triangle $\triangle$ with vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$, let $(\lambda_1, \lambda_2, \lambda_3)$ be the unique barycentric coordinates of point (u, v) with respect to $\triangle$ such that

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

Then, we define

$$T_l(u, v) := \lambda_l, \quad l = 1, 2, 3.$$

These functions can be interpreted as triangular Bernstein polynomials of degree 1. They are non-negative on the domain triangle $\triangle$. Moreover, they are linearly independent, form a partition of unity, and span the space of bivariate polynomials of total degree less than or equal to 1. Then, we require that N_l in Eq. (30) is a spline function f such that it satisfies the continuity constraints in Eq. (26), with

$$\alpha = T_l(0,0), \quad \beta = \frac{\partial T_l}{\partial u}(0,0), \quad \gamma = \frac{\partial T_l}{\partial v}(0,0),$$

for $l = 1, 2, 3$. In the interest of standardization, we choose the triangle $\triangle$ as equilateral with vertices

$$\boldsymbol{v}_1 = (2\rho_2, 0), \quad \boldsymbol{v}_2 = (-\rho_2, \sqrt{3}\rho_2), \quad \boldsymbol{v}_3 = (-\rho_2, -\sqrt{3}\rho_2);$$

recall the definition of ρ_2 from Eq. (19). After some calculations, we deduce that

$$\bar{E}_{1,(i1)} = \bar{E}_{2,(i1)} = \bar{E}_{3,(i1)} = \frac{1}{3},$$

and

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

This relation says that $\left(\bar{E}_{1,(i2)}, \bar{E}_{2,(i2)}, \bar{E}_{3,(i2)}\right)$ are simply the barycentric coordinates of the control point $\boldsymbol{F}_{i2} := (\rho_2 \cos(\theta_i), \rho_2 \sin(\theta_i))$ with respect to $\triangle$. It is easily checked that $\triangle$ encloses the circle centered at $(0,0)$ with a radius of ρ_2, and hence $\left(\bar{E}_{1,(i2)}, \bar{E}_{2,(i2)}, \bar{E}_{3,(i2)}\right)$ are guaranteed to be non-negative. In summary, $\bar{\boldsymbol{E}}$ is specified as

$$\bar{\boldsymbol{E}} := \begin{bmatrix} \frac{1}{3} & \cdots & \frac{1}{3} & \bar{E}_{1,(12)} & \cdots & \bar{E}_{1,(i2)} & \cdots & \bar{E}_{1,(n^s 2)} \\ \frac{1}{3} & \cdots & \frac{1}{3} & \bar{E}_{2,(12)} & \cdots & \bar{E}_{2,(i2)} & \cdots & \bar{E}_{2,(n^s 2)} \\ \frac{1}{3} & \cdots & \frac{1}{3} & \bar{E}_{3,(12)} & \cdots & \bar{E}_{3,(i2)} & \cdots & \bar{E}_{3,(n^s 2)} \end{bmatrix}. \tag{31}$$

This matrix has full rank and the column sum is equal to one, thus confirming that $\boldsymbol{E}$ is DTA-compatible. The following result follows from the above discussion.

Theorem 3.2. *The C^1 smooth polar spline functions in the set $\{N_l^{pol} : l = 1, \ldots, n\}$ are linearly independent, locally supported, and form a convex partition of unity on Ω^{pol}.*

Remark 3.3. As long as $\triangle$ is chosen to be a triangle enclosing the first polar ring of control points $\boldsymbol{F}_{ij}$ for a given configuration, we are guaranteed non-negative extraction coefficients. It is only in the interest of standardization that we have chosen to fix $\triangle$ as an equilateral triangle with a fixed pattern of vertices.

Given n control points $\boldsymbol{f}_l \in \mathbb{R}^d$, $d \geq 3$, we can construct a C^1 polar surface $\boldsymbol{f}$ embedded in $\mathbb{R}^d$,

$$\boldsymbol{f}(u, v) = \sum_{l=1}^{n} \boldsymbol{f}_l N_l^{pol}(u, v),$$

or, equivalently, after pulling back to Ω,

$$\boldsymbol{f}(s,t) = \sum_{l=1}^{n} \boldsymbol{f}_l N_l(s,t). \tag{32}$$

The behavior of $\boldsymbol{f}$ at the polar point is going to be fully specified by the first three control points $\boldsymbol{f}_1, \boldsymbol{f}_2$ and $\boldsymbol{f}_3$. These control points can be thought of as forming a *control triangle*. For a fixed surface, the transpose of the extraction matrix $\boldsymbol{E}$ defines the relationship between control points of the B_{ij} and control points of the smooth N_l. More precisely, if

$$\sum_{i=1}^{n^s} \sum_{j=1}^{n^t} \boldsymbol{g}_{ij} B_{ij}(s,t) = \boldsymbol{f}(s,t) = \sum_{l=1}^{n} \boldsymbol{f}_l N_l(s,t),$$

then

$$\boldsymbol{g}_{ij} = \sum_{l=1}^{n} E_{l,(ij)} \boldsymbol{f}_l. \tag{33}$$

In particular, the transpose of the extraction matrix $\bar{\boldsymbol{E}}$ in Eq. (31) essentially computes the control points of the zeroth and the first polar rings of basis functions B_{ij} as convex combinations of $\boldsymbol{f}_1, \boldsymbol{f}_2$ and $\boldsymbol{f}_3$. In other words, it forces the zeroth and the first polar rings of the control points $\boldsymbol{g}_{ij}$ to be coplanar. The plane in which they lie is the one passing through $\boldsymbol{f}_1, \boldsymbol{f}_2$ and $\boldsymbol{f}_3$, and hence the control triangle must be tangent to the surface $\boldsymbol{f}$ at the pole.

In light of Theorem 3.2 we can immediately conclude the following properties for the pulled-back functions N_l, $l = 1, \ldots, n$.

Corollary 3.4. *The spline functions in the set $\{N_l : l = 1, \ldots, n\}$ are linearly independent, locally supported, and form a convex partition of unity on Ω. Moreover, any polar surface $\boldsymbol{f}$ as in Eq. (32) with non-collinear control points $\boldsymbol{f}_1, \boldsymbol{f}_2$ and $\boldsymbol{f}_3$ will have a well-defined tangent plane at the pole.*

Remark 3.5. According to Section 2.2, the univariate sets of basis functions $\{B_i^s : i = 1, \ldots, n^s\}$ and $\{B_j^t : j = 1, \ldots, n^t\}$ are built from local NURBS basis functions through extraction matrices $\boldsymbol{H}^s$ and $\boldsymbol{H}^t$, respectively. Hence, by combining the matrices $\boldsymbol{E}$, $\boldsymbol{H}^s$ and $\boldsymbol{H}^t$, the spline basis functions in the set $\{N_l : l = 1, \ldots, n\}$ can be directly expressed in terms of local tensor-product NURBS basis functions.

3.2.2. Double polar point

When dealing with double polar surfaces, the spline construction can be obtained by collapsing a pair of two opposite edges as illustrated in Fig. 3(b). As explained in Section 3.1, the smoothness treatment of the two poles can be done separately for $n^t \geq 4$. In this case, each pole leads to a local extraction matrix by applying the same procedure as in Section 3.2.1 and the combined global extraction matrix takes the following sparse block-diagonal form:

$$\boldsymbol{E} := \begin{bmatrix} \bar{\boldsymbol{E}}^{(1)} & & \\ & \boldsymbol{I} & \\ & & \bar{\boldsymbol{E}}^{(2)} \end{bmatrix}, \tag{34}$$

where $\boldsymbol{I}$ is the identity matrix of size $n^s(n^t - 4) \times n^s(n^t - 4)$ and $\bar{\boldsymbol{E}}^{(i)}$, $i = 1, 2$, are matrices of size $3 \times 2n^s$. By choosing the two

polar parameterizations $\boldsymbol{F}(\Omega)$ and $\hat{\boldsymbol{F}}(\Omega)$ specified in Section 3.1, it is easily verified that one can set

$$\bar{\boldsymbol{E}}^{(1)} := \bar{\boldsymbol{E}}, \quad \bar{\boldsymbol{E}}^{(2)} := \boldsymbol{J}_3 \bar{\boldsymbol{E}} \boldsymbol{J}_{2n^s}, \tag{35}$$

where $\bar{\boldsymbol{E}}$ is the matrix defined in Eq. (31) and $\boldsymbol{J}_k$ is the exchange matrix of size $k \times k$, i.e., an anti-diagonal matrix of the form

$$\boldsymbol{J}_k := \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}.$$

The extraction matrix $\boldsymbol{E}$ can then be used to compute the set of spline functions $\{N_l : l = 1, \ldots, n\}$ in terms of the tensor-product functions $\{B_{ij} : i = 1, \ldots, n^s; j = 1, \ldots, n^t\}$. Similar to the single-pole result in Corollary 3.4, these spline functions have the following properties.

Corollary 3.6. *The spline functions in the set $\{N_l : l = 1, \ldots, n\}$ are linearly independent, locally supported, and form a convex partition of unity on Ω. Moreover, any polar surface $\boldsymbol{f}$ as in Eq. (32) with non-collinear control points $\boldsymbol{f}_1, \boldsymbol{f}_2$ and $\boldsymbol{f}_3$ and with non-collinear control points $\boldsymbol{f}_{n-2}, \boldsymbol{f}_{n-1}$ and $\boldsymbol{f}_n$ will have a well-defined tangent plane at both poles.*

3.3. Refinement of piecewise-rational polar surfaces

A polar spline surface can be refined in a manner similar to the one discussed in Section 2.3. We begin with the following observation. Consider the 3×3 matrix

$$\bar{\boldsymbol{M}} := \begin{bmatrix} \frac{1}{3} & 0 & \frac{1}{3} \\ -\frac{1}{6} & \frac{\sqrt{3}}{6} & \frac{1}{3} \\ -\frac{1}{6} & -\frac{\sqrt{3}}{6} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0 & \cos(\theta_\iota) & \cos(\theta_\kappa) \\ 0 & \sin(\theta_\iota) & \sin(\theta_\kappa) \\ 1 & 1 & 1 \end{bmatrix},$$

for some angles θ_ι and $\theta_\kappa \notin \{\theta_\iota, \theta_\iota + \pi\}$ selected from the set in Eq. (20). This is a submatrix of $\bar{\boldsymbol{E}}$, defined in Eq. (31), consisting of three linearly independent columns. Its inverse is given by

$$\begin{aligned} \bar{\boldsymbol{M}}^{-1} &= \begin{bmatrix} 0 & \cos(\theta_\iota) & \cos(\theta_\kappa) \\ 0 & \sin(\theta_\iota) & \sin(\theta_\kappa) \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{3} & 0 & \frac{1}{3} \\ -\frac{1}{6} & \frac{\sqrt{3}}{6} & \frac{1}{3} \\ -\frac{1}{6} & -\frac{\sqrt{3}}{6} & \frac{1}{3} \end{bmatrix}^{-1} \\ &= \frac{1}{\sin(\theta_\kappa - \theta_\iota)} \bar{\boldsymbol{L}} \begin{bmatrix} 2 & -1 & -1 \\ 0 & \sqrt{3} & -\sqrt{3} \\ 1 & 1 & 1 \end{bmatrix}, \end{aligned}$$

where

$$\bar{\boldsymbol{L}} := \begin{bmatrix} \sin(\theta_\iota) - \sin(\theta_\kappa) & \cos(\theta_\kappa) - \cos(\theta_\iota) & \sin(\theta_\kappa - \theta_\iota) \\ \sin(\theta_\kappa) & -\cos(\theta_\kappa) & 0 \\ -\sin(\theta_\iota) & \cos(\theta_\iota) & 0 \end{bmatrix}.$$

Then, we define a sparse matrix $\bar{\boldsymbol{D}}$ of size $2n^s \times 3$, whose non-zero entries $\bar{D}_{ij}$ are identified as follows: for $j = 1, 2, 3$,

$$\bar{D}_{1j} := \bar{M}^{-1}_{1j}, \quad \bar{D}_{n^s+\iota,j} := \bar{M}^{-1}_{2j}, \quad \bar{D}_{n^s+\kappa,j} := \bar{M}^{-1}_{2j}.$$

From its construction it is clear that the product $\bar{\boldsymbol{E}}\bar{\boldsymbol{D}}$ is equal to the identity matrix. Note that the product $(\boldsymbol{J}_3 \bar{\boldsymbol{E}} \boldsymbol{J}_{2n^s})(\boldsymbol{J}_{2n^s} \bar{\boldsymbol{D}} \boldsymbol{J}_3)$ is also equal to the identity matrix. Hence, keeping in mind the definition of $\boldsymbol{E}$ in Eqs. (34)–(35), the matrix

$$\boldsymbol{D} := \begin{bmatrix} \bar{\boldsymbol{D}} & & \\ & \boldsymbol{I} & \\ & & \boldsymbol{J}_{2n^s} \bar{\boldsymbol{D}} \boldsymbol{J}_3 \end{bmatrix} \tag{36}$$

gives rise to a product $\boldsymbol{E}\boldsymbol{D}$ that is equal to the identity matrix. A similar matrix $\boldsymbol{D}$ can be found (with only two diagonal blocks) for the matrix $\boldsymbol{E}$ defined in Eq. (29).

Any refinement matrix of polar splines can then be built using the following procedure. First, we compute the control points of the tensor-product basis functions B_{ij} as in Eq. (33). Then, we refine the tensor-product control points using a tensor product of univariate refinement matrices (see Section 2.3). Denote this matrix with $\boldsymbol{S}$, and denote the polar spline extraction matrices before and after refinement with $\boldsymbol{E}$ and $\tilde{\boldsymbol{E}}$, respectively. Then, control points of the refined polar spline basis functions can be obtained by applying a matrix $\boldsymbol{R}$ to the original set of polar control points, where $\boldsymbol{R}$ is computed by solving the following (overdetermined) linear system with a unique solution,

$$\boldsymbol{R}\tilde{\boldsymbol{E}} = \boldsymbol{E}\boldsymbol{S}.$$

After multiplication of both sides of this system with the matrix $\tilde{\boldsymbol{D}}$ (corresponding to $\tilde{\boldsymbol{E}}$) as defined in Eq. (36), we arrive at

$$\boldsymbol{R} = \boldsymbol{E}\boldsymbol{S}\tilde{\boldsymbol{D}}. \tag{37}$$

Remark 3.7. Similar to Remark 2.6, there is some flexibility in the definition of $\tilde{\boldsymbol{D}}$ for the computation of $\boldsymbol{R}$ in Eq. (37), as long as it is a right inverse of $\tilde{\boldsymbol{E}}$. The current choice is again done for the sake of simplicity. For the computation of $\bar{\boldsymbol{M}}^{-1}$, from a numerical point of view, it is advised to select the angles θ_ι and θ_κ from the set in Eq. (20) such that their difference is as close as possible to $\pm\pi/2$. For instance, taking $\iota = \kappa + \lfloor n^s/4 + 1/2 \rfloor$ gives

$$\theta_\kappa - \theta_\iota = \frac{2\pi}{n^s} \left\lfloor \frac{n^s}{4} + \frac{1}{2} \right\rfloor,$$

and then $\det(\bar{\boldsymbol{M}}) = \sin(\theta_\kappa - \theta_\iota) \geq 0.86$ for $n^s \geq 3$.

3.4. Ellipsoids and spheres

Let us now present explicit descriptions of ellipsoids (and as a special case also spheres) built using the C^1 polar spline framework discussed thus far. Once again, in the interest of providing the simplest possible representations, we focus on smooth descriptions of low(est) degree only. More precisely, we present three explicit descriptions of arbitrary ellipsoids using smooth polar splines. The first one uses 8 rational pieces of bi-degree $(2, 2)$; the second one uses 4 rational pieces of bi-degree $(2, 3)$; and the last one uses 2 rational pieces of bi-degree $(3, 3)$. All approaches will define six C^1 polar spline functions N_l and associated control points $\boldsymbol{f}_l$, $l = 1, \ldots, 6$, such that the bivariate surface $\boldsymbol{f}$,

$$\boldsymbol{f}(s, t) := (f_x(s, t), f_y(s, t), f_z(s, t)) := \sum_{l=1}^{6} \boldsymbol{f}_l N_l(s, t),$$

describes the exact ellipsoid centered at $(0, 0, 0)$ and with axis lengths (a_x, a_y, a_z),

$$\left(\frac{f_x}{a_x}\right)^2 + \left(\frac{f_y}{a_y}\right)^2 + \left(\frac{f_z}{a_z}\right)^2 = 1. \tag{38}$$

To this end, we will make repeated use of the identity matrix $\boldsymbol{I}_k$ of size $k \times k$ and the exchange matrix $\boldsymbol{J}_k$ of size $k \times k$.

To visually illustrate the smoothness of $\boldsymbol{f}$, we will also show the surface $\tilde{\boldsymbol{f}}$ obtained by perturbing one of the control points. This surface will also be smooth at the poles. For uniformity throughout the examples, we will choose the control points of the perturbed surface as

$$\tilde{\boldsymbol{f}}_l := \begin{cases} \boldsymbol{f}_l + (0, 0, 4a_x), & l = 3, \\ \boldsymbol{f}_l, & l \neq 3. \end{cases} \tag{39}$$

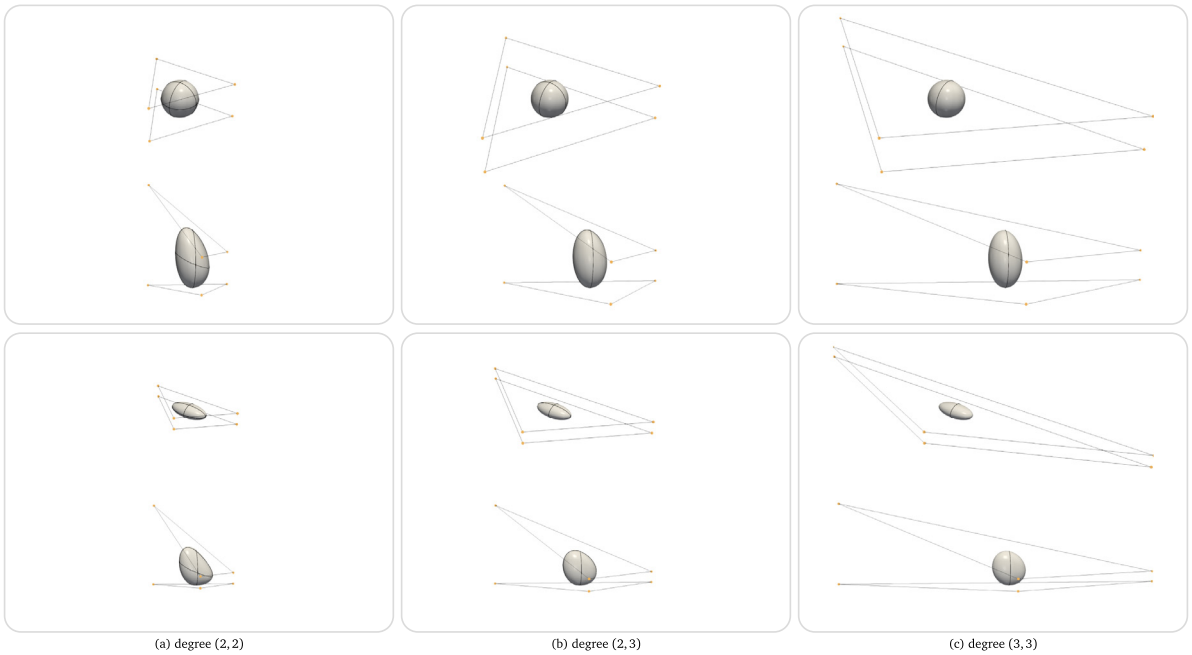


Fig. 4. Different C^1 smooth descriptions of unit spheres (top row boxes) and ellipsoids with axis lengths $(1, \frac{1}{2}, \frac{1}{3})$ (bottom row boxes). The quadrics in figures (a)–(c) are made up of 8, 4 and 2 rational pieces of bi-degrees $(2, 2)$, $(2, 3)$ and $(3, 3)$, respectively. In particular, the top figure in each box shows the exact quadric, while the bottom figure shows a deformed quadric obtained by raising one of the control points of the exact quadric; see Examples 3.8, 3.9 and 3.10 for details.

3.4.1. C^1 description of degree $(2, 2)$

For the first approach, we choose $\Omega^s = [0, 4]$ and $\Omega^t = [0, 2]$, and build the univariate rational spline spaces $\mathscr{R}^s$ (periodic) and $\mathscr{R}^t$ on them using the following sets of parameters:

$$\mathscr{R}^s : \left.\begin{aligned} \boldsymbol{\xi}^{(i)} &= [0, 0, 0, 1, 1, 1] \\ \boldsymbol{w}^{(i)} &= \left[1, \tfrac{\sqrt{2}}{2}, 1\right] \end{aligned}\right\} \; i = 1, \ldots, 4;$$

$$\mathscr{R}^t : \left.\begin{aligned} \boldsymbol{\xi}^{(i)} &= [0, 0, 0, 1, 1, 1] \\ \boldsymbol{w}^{(i)} &= \left[1, \tfrac{\sqrt{2}}{2}, 1\right] \end{aligned}\right\} \; i = 1, 2.$$

The corresponding piecewise-NURBS extraction operators are $\boldsymbol{H}^s = \boldsymbol{H}^{per}$ defined in Eq. (17) and $\boldsymbol{H}^t$ given by

$$\boldsymbol{H}^t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

respectively. The full tensor-product extraction matrix is obtained as

$$\boldsymbol{H} = \boldsymbol{H}^s \otimes \boldsymbol{H}^t. \tag{40}$$

The next ingredient in the C^1 smooth polar construction is the polar extraction operator, and it can be computed, using above formulas, to be

$$\boldsymbol{E} = \begin{bmatrix} \bar{\boldsymbol{E}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_3 \bar{\boldsymbol{E}} \boldsymbol{J}_8 \end{bmatrix}, \tag{41}$$

where the matrix $\bar{\boldsymbol{E}}$ is given by

$$\bar{\boldsymbol{E}} = \frac{1}{3} + \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & -e_+ & -e_- & e_+ & e_- \\ 0 & 0 & 0 & 0 & e_- & e_+ & -e_- & -e_+ \end{bmatrix},$$

and

$$e_+ = \frac{\sqrt{3}+1}{6}, \quad e_- = \frac{\sqrt{3}-1}{6};$$

see Eqs. (31) and (34)–(35). Here the compact notation of adding a scalar to a matrix means the entrywise operation of adding the scalar to each entry of the matrix. The matrix $\boldsymbol{E}$ maps a total of 16 tensor-product C^{-1} piecewise-NURBS B_j of degree $(2, 2)$ to a total of 6 C^1 polar splines N_l. Equivalently, $\boldsymbol{EH}$ maps a total of 72 NURBS b_j of degree $(2, 2)$ to a total of 6 C^1 polar splines N_l. These relations are encapsulated in the following equation,

$$N_l(s, t) = \sum_{j=1}^{16} E_{lj} B_j(s, t) = \sum_{j=1}^{16} E_{lj} \sum_{k=1}^{72} H_{jk} b_k(s, t).$$

Finally, the required ellipsoid in Eq. (38) is obtained by defining the associated 6 control points as

$$\boldsymbol{f}_1 = \left(0, 2\sqrt{2}a_y, a_z\right), \quad \boldsymbol{f}_2 = \left(-\sqrt{6}a_x, -\sqrt{2}a_y, a_z\right),$$
$$\boldsymbol{f}_3 = \left(\sqrt{6}a_x, -\sqrt{2}a_y, a_z\right), \quad \boldsymbol{f}_4 = \left(-\sqrt{6}a_x, -\sqrt{2}a_y, -a_z\right),$$
$$\boldsymbol{f}_5 = \left(\sqrt{6}a_x, -\sqrt{2}a_y, -a_z\right), \quad \boldsymbol{f}_6 = \left(0, 2\sqrt{2}a_y, -a_z\right).$$

Example 3.8. The box at the top in Fig. 4(a) shows a bi-degree $(2, 2)$ unit sphere built by choosing $a_x = a_y = a_z = 1$, while the box at the bottom shows a bi-degree $(2, 2)$ ellipsoid with axis lengths $(1, \frac{1}{2}, \frac{1}{3})$ built by choosing $a_x = 2a_y = 3a_z = 1$. These descriptions use only 8 rational pieces. In each box, the figure at the top shows the exact quadric, while the figure at the bottom shows the deformed quadric obtained by perturbing the control points as per Eq. (39). The exact and deformed surfaces are all C^1 smooth at the poles. ♦

3.4.2. C^1 description of degree $(2, 3)$

For the second approach, we choose $\Omega^s = [0, 4]$ and $\Omega^t = [0, 1]$, and build the univariate rational spline spaces $\mathscr{R}^s$ (peri-

odic) and $\mathscr{R}^t$ on them using the following sets of parameters:

$$\left.\begin{array}{l}\mathscr{R}^s:\ \boldsymbol{\xi}^{(i)} = [0,0,0,1,1,1]\\ \qquad\ \boldsymbol{w}^{(i)} = \left[1, \frac{\sqrt{2}}{2}, 1\right]\end{array}\right\}\ i = 1,\ldots,4;$$

$$\mathscr{R}^t:\ \begin{array}{l}\boldsymbol{\xi}^{(1)} = [0,0,0,0,1,1,1,1]\\ \boldsymbol{w}^{(1)} = \left[1, \frac{1}{3}, \frac{1}{3}, 1\right]\end{array}.$$

The corresponding piecewise-NURBS extraction operators are $\boldsymbol{H}^s = \boldsymbol{H}^{per}$ defined in Eq. (17) and $\boldsymbol{H}^t = \boldsymbol{I}_4$, respectively. The full tensor-product extraction matrix $\boldsymbol{H}$ is obtained as in Eq. (40). Moreover, the polar extraction operator $\boldsymbol{E}$ is equal to the matrix in Eq. (41). The latter matrix maps a total of 16 tensor-product C^{-1} piecewise-NURBS B_j of degree $(2,3)$ to a total of 6 C^1 polar splines N_l. Equivalently, $\boldsymbol{EH}$ maps a total of 48 tensor-product NURBS b_j of degree $(2,3)$ to a total of 6 C^1 polar splines N_l. These relations are encapsulated in the following equation,

$$N_l(s,t) = \sum_{j=1}^{16} E_{lj} B_j(s,t) = \sum_{j=1}^{16} E_{lj} \sum_{k=1}^{48} H_{jk} b_k(s,t).$$

Finally, the required ellipsoid in Eq. (38) is obtained by defining the associated 6 control points as

$$\boldsymbol{f}_1 = \left(0, 4\sqrt{2}a_y, a_z\right),\quad \boldsymbol{f}_2 = \left(-2\sqrt{6}a_x, -2\sqrt{2}a_y, a_z\right),$$
$$\boldsymbol{f}_3 = \left(2\sqrt{6}a_x, -2\sqrt{2}a_y, a_z\right),\quad \boldsymbol{f}_4 = \left(-2\sqrt{6}a_x, -2\sqrt{2}a_y, -a_z\right),$$
$$\boldsymbol{f}_5 = \left(2\sqrt{6}a_x, -2\sqrt{2}a_y, -a_z\right),\quad \boldsymbol{f}_6 = \left(0, 4\sqrt{2}a_y, -a_z\right).$$

Note that, since we are using a higher-degree representation compared to Section 3.4.1, the control points move farther away from the spline surface, mimicking the behavior of classical NURBS.

Example 3.9. The box at the top in Fig. 4(b) shows a bi-degree $(2,3)$ unit sphere built by choosing $a_x = a_y = a_z = 1$, while the box at the bottom shows a bi-degree $(2,3)$ ellipsoid with axis lengths $(1, \frac{1}{2}, \frac{1}{3})$ built by choosing $a_x = 2a_y = 3a_z = 1$. These descriptions use 4 rational pieces. In each box, the figure at the top shows the exact quadric, while the figure at the bottom shows the deformed quadric obtained by perturbing the control points as per Eq. (39). The exact and deformed surfaces are all C^1 smooth at the poles. ♦

3.4.3. C^1 description of degree $(3,3)$

Finally, for the third approach, we choose $\Omega^s = [0,2]$ and $\Omega^t = [0,1]$, and build the univariate rational spline spaces $\mathscr{R}^s$ (periodic) and $\mathscr{R}^t$ on them using the following sets of parameters:

$$\left.\begin{array}{l}\mathscr{R}^s:\ \boldsymbol{\xi}^{(i)} = [0,0,0,0,1,1,1,1]\\ \qquad\ \boldsymbol{w}^{(i)} = \left[1, \frac{1}{3}, \frac{1}{3}, 1\right]\end{array}\right\}\ i = 1,2;$$

$$\mathscr{R}^t:\ \begin{array}{l}\boldsymbol{\xi}^{(1)} = [0,0,0,0,1,1,1,1]\\ \boldsymbol{w}^{(1)} = \left[1, \frac{1}{3}, \frac{1}{3}, 1\right]\end{array}.$$

The corresponding piecewise-NURBS extraction operators are $\boldsymbol{H}^s = \boldsymbol{H}^{per}$ defined in Eq. (18) and $\boldsymbol{H}^t = \boldsymbol{I}_4$, respectively. The full tensor-product extraction matrix $\boldsymbol{H}$ is obtained as in Eq. (40). Moreover, the polar extraction operator $\boldsymbol{E}$ is equal to the matrix in Eq. (41). The latter matrix maps a total of 16 tensor-product C^{-1} piecewise-NURBS B_j of degree $(3,3)$ to a total of 6 C^1 polar splines N_l. Equivalently, $\boldsymbol{EH}$ maps a total of 32 tensor-product NURBS b_j of degree $(3,3)$ to a total of 6 C^1 polar splines N_l. These relations are encapsulated in the following equation,

$$N_l(s,t) = \sum_{j=1}^{16} E_{lj} B_j(s,t) = \sum_{j=1}^{16} E_{lj} \sum_{k=1}^{32} H_{jk} b_k(s,t).$$

Finally, the required ellipsoid in Eq. (38) is obtained by defining the associated 6 control points as

$$\boldsymbol{f}_1 = \left(0, 4\sqrt{2}a_y, a_z\right),\quad \boldsymbol{f}_2 = \left(-4\sqrt{6}a_x, -2\sqrt{2}a_y, a_z\right),$$
$$\boldsymbol{f}_3 = \left(4\sqrt{6}a_x, -2\sqrt{2}a_y, a_z\right),\quad \boldsymbol{f}_4 = \left(-4\sqrt{6}a_x, -2\sqrt{2}a_y, -a_z\right),$$
$$\boldsymbol{f}_5 = \left(4\sqrt{6}a_x, -2\sqrt{2}a_y, -a_z\right),\quad \boldsymbol{f}_6 = \left(0, 4\sqrt{2}a_y, -a_z\right).$$

Observe again that, since we are using a higher-degree representation compared to Sections 3.4.1 and 3.4.2 , the control points move even farther away from the spline surface, mimicking the behavior of classical NURBS.

Example 3.10. The box at the top in Fig. 4(c) shows a bi-degree $(3,3)$ unit sphere built by choosing $a_x = a_y = a_z = 1$, while the box at the bottom shows a bi-degree $(3,3)$ ellipsoid with axis lengths $(1, \frac{1}{2}, \frac{1}{3})$ built by choosing $a_x = 2a_y = 3a_z = 1$. These descriptions use only 2 rational pieces. In each box, the figure at the top shows the exact quadric, while the figure at the bottom shows the deformed quadric obtained by perturbing the control points as per Eq. (39). The exact and deformed surfaces are all C^1 smooth at the poles. ♦

Remark 3.11. The examples presented here have focused on the simplest possible C^1 descriptions of quadrics, namely descriptions that either use lowest-degree splines — bi-degree $(2,2)$ — or the smallest number of polynomial pieces — two. Unsurprisingly, these simplest descriptions can lead to large control triangles since each control point influences a large portion of the spline surface. Nevertheless, localized control is easily attained upon refinement (see Section 3.3) and, in particular, refinement also leads to much smaller control triangles that offer much finer geometric control. The surface shown in Fig. 5 illustrates this point. This surface has been obtained by refining and modifying the control points of the bi-degree $(2,2)$ sphere from Fig. 4(a). The smaller control triangle is visible near the top of the figure.

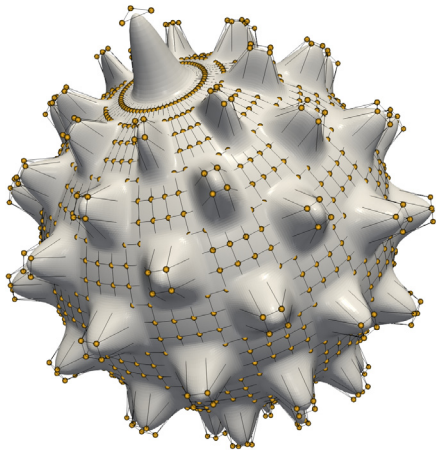


Fig. 5. A smoothly deformed sphere built from Fig. 4(a) by refining the surface and then modifying the control points.

4. Conclusions

We have presented a general class of C^1 smooth rational splines that allow for the construction and refinement of C^1

smooth curves and (polar) surfaces. They are built by gluing together multiple sets of NURBS basis functions with C^1 smoothness using a DTA-compatible extraction matrix. The main features of the splines we have built are the following:

- all standard properties of NURBS, including support for intuitive control-point-based design,
- (local) degree elevation and knot insertion based on classical NURBS refinement,
- low-degree C^1 descriptions of exact ellipses and ellipsoids, and
- compatibility with CAD or CAE software through the explicit representation in terms of NURBS.

In particular, with regard to the last two bullets above, we believe that the explicit, NURBS-compatible C^1 descriptions of ellipses and ellipsoids provided herein will be of use to geometric modelers [21] and computational scientists [9] alike. For instance, the exact C^1 (re)parameterizations at polar points may make the design of algorithms more stable and efficient; it may also avoid the need for special treatment of polar points.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

H. Speleers was partially supported by the Beyond Borders Program of the University of Rome Tor Vergata, Italy through the project ASTRID (CUP E84I19002250005) and by the MIUR Excellence Department Project awarded to the Department of Mathematics, University of Rome Tor Vergata, Italy (CUP E83C18000100006). He is a member of Gruppo Nazionale per il Calcolo Scientifico, Istituto Nazionale di Alta Matematica.

References

[1] Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Engrg 2005;194:4135–95.
[2] Toshniwal D, Speleers H, Hiemstra RR, Hughes TJR. Multi-degree smooth polar splines: A framework for geometric modeling and isogeometric analysis. Comput Methods Appl Mech Engrg 2017;316:1005–61.
[3] Borden MJ, Scott MA, Evans JA, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of NURBS. Internat J Numer Methods Engrg 2011;87:15–47.
[4] Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of T-splines. Internat J Numer Methods Engrg 2011;88:126–56.
[5] Speleers H. Algorithm 999: Computation of multi-degree B-splines. ACM Trans Math Software 2019;45:43.
[6] Toshniwal D, Speleers H, Hiemstra RR, Manni C, Hughes TJR. Multi-degree B-splines: Algorithmic computation and properties. Comput Aided Geom Design 2020;76:101792.
[7] Hiemstra RR, Hughes TJR, Manni C, Speleers H, Toshniwal D. A Tchebycheffian extension of multi-degree B-splines: Algorithmic computation and properties. SIAM J Numer Anal 2020;58:1138–63.
[8] Toshniwal D, Speleers H, Hughes TJR. Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: Geometric design and isogeometric analysis considerations. Comput Methods Appl Mech Engrg 2017;327:411–58.
[9] Toshniwal D, Hughes TJR. Isogeometric discrete differential forms: Non-uniform degrees, Bézier extraction, polar splines and flows on surfaces. Comput Methods Appl Mech Engrg 2020. (in press).
[10] Barsky BA. Computer graphics and geometric modeling using beta-splines. Springer-Verlag; 1988.
[11] Bangert C, Prautzsch H. Circle and sphere as rational splines. Neural Parallel Sci Comput 1997;5:153–62.
[12] Lu J. Circular element: Isogeometric elements of smooth boundary. Comput Methods Appl Mech Engrg 2009;198:2391–402.
[13] Lu J, Zhou X. Cylindrical element: Isogeometric model of continuum rod. Comput Methods Appl Mech Engrg 2011;200:233–41.
[14] Piegl L, Tiller W. The NURBS book. 2nd ed. Springer-Verlag; 2012.
[15] Farin G. Curves and surfaces for CAGD: A practical guide. 5th ed. Morgan Kaufmann; 2002.
[16] Karčiauskas K, Peters J. Bicubic polar subdivision. ACM Trans Graph 2007;26:14.
[17] Myles A. Curvature-continuous bicubic subdivision surfaces for polar configurations (Ph.D. thesis), University of Florida; 2008.
[18] Karčiauskas K, Myles A, Peters J. A C^2 polar jet subdivision. In: Polthier K, Sheffer A, editors. Proceedings of the fourth eurographics symposium on geometry processing. Eurographics Association; 2006, p. 173–80.
[19] Myles A, Peters J. Bi-3 C^2 polar subdivision. ACM Trans Graph 2009;28:48.
[20] Myles A, Peters J. C^2 splines covering polar configurations. Comput Aided Des 2011;43:1322–9.
[21] Karčiauskas K, Peters J. Smooth polar caps for locally quad-dominant meshes. Comput Aided Geom Design 2020;81:101908.
[22] Shi K-L, Yong J-H, Tang L, Sun J-G, Paul J-C. Polar NURBS surface with curvature continuity. Comput Graph Forum 2013;32:363–70.
[23] Shi K-L, Yong J-H, Sun J-G, Paul J-C. G^n filling orbicular N-sided holes using periodic B-spline surfaces. Sci China Inf Sci 2011;54:1383–94.
[24] Takacs T. Construction of smooth isogeometric function spaces on singularly parameterized domains. In: Boissonnat J-D, et al., editors. Curves and surfaces. LNCS, vol. 9213, Springer; 2015, p. 433–51.
[25] Gomes A, Voiculescu I, Jorge J, Wyvill B, Galbraith C. Implicit curves and surfaces: Mathematics, data structures and algorithms. Springer-Verlag; 2009.
[26] Rogers DF. An introduction to NURBS: With historical perspective. Morgan Kaufmann; 2001.
[27] Dyn N, Micchelli CA. Piecewise polynomial spaces and geometric continuity of curves. Numer Math 1989;54:319–37.