A Cox-de Boor-type recurrence relation for C1 multi-degree splines

A Cox-de Boor-type recurrence relation for C^1 multi-degree splines

Carolina Vittoria Beccari, Giulio Casciola

Department of Mathematics, University of Bologna, P.zza di Porta S.Donato 5, 40126 Bologna, Italy

https://doi.org/10.1016/j.cagd.2019.101784

A Cox-de Boor-type recurrence relation for C^1 multi-degree splines

Carolina Vittoria Beccari[a,*], Giulio Casciola[a]

[a]*Department of Mathematics, University of Bologna, Piazza di Porta San Donato 5, 40126 Bologna, Italy*

Abstract

Multi-degree splines are piecewise functions comprised of polynomial segments of different degrees. A subclass of such splines, that we refer to as C^1 MD-splines, is featured by arbitrary continuity between pieces of same degree and at most C^1 continuity between pieces of different degrees. For these spline spaces a B-spline basis can be defined by means of an integral recurrence relation, as an instance of the more general construction in [1]. In this paper, we provide efficient formulas for evaluating C^1 MD-splines and their derivatives, akin to the classical B-spline recurrence relations. Furthermore we derive algorithms for geometric design, including knot insertion and local degree elevation. Finally we demonstrate the utility of these splines, not only for geometric modeling, but also for graphical applications, discussing in particular the advantages for modeling and storing vector images.

Keywords: Multi-Degree spline, B-spline basis recurrence relation, B-spline derivative, Geometric modeling, Vector graphics
2010 MSC: 65D07, 65D17, 41A15, 68W25

1. Introduction

Multi-degree splines (MD-splines, for short) are piecewise polynomial functions where each piece can have different degree. In this way, they require less control points to represent a shape with respect to using conventional splines (piecewise polynomial functions having pieces of the same degree). Their study was initiated in [2–4] and has been a subject of several recent researches, with applications ranging from geometric design [1, 5–10] to isogeometric analysis [11].

In a recent paper [1] multi-degree spline spaces are defined starting from a sequence of intervals, each associated with a degree, and from a sequence of continuities, so as to replicate the setup of conventional splines. Consequently, the dimension of a space, and therefore the number of control points, is unambiguously determined. Splines in these spaces have continuity C^{d-1} between pieces of same degree d and C^k between pieces of degrees d_i and d_{i+1}, with $k := \min\{d_i, d_{i+1}\}$. Moreover continuity can locally be reduced by means of multiple knots, in a similar way as for conventional splines. The general framework in [1] comprises as special cases some previously appeared instances of MD-splines, such as those in [6, 7]. Furthermore, the considered MD-spline spaces possess B-spline-type bases that can be generated by an integral recurrence relation. All relations of this kind, however, are of little practical use for evaluating a spline, due to the computational complexity of the integral formulation. To overcome this limitation, the same work proposes another method, much more viable from a computational point of view, which consists in determining each B-spline basis function as the solution of a small linear system. For splines in the same spaces, an alternative evaluation algorithm, reported in [11, 12], is based on a global extraction operator that allows to efficiently represent each element of the multi-degree B-spline basis as a linear combination of local B-splines of different degrees.

The evaluation of MD-splines through a Cox-de Boor-type recurrence relation has previously been studied in [8]. The input of the algorithm consists of a sequence of knot intervals, a sequence of degrees and a control polygon. Moreover, it is required that knot intervals be associated with either controls points or control polygon edges according

*Corresponding author.
Email addresses: carolina.beccari2@unibo.it (Carolina Vittoria Beccari), giulio.casciola@unibo.it (Giulio Casciola)

to a *degree assignment rule*. The resulting splines are C^1 continuous in between intervals having different associated degrees and C^{d-1} in case of same degree d. However, for a given input configuration, the scheme might generate a curve that does not belong to any of the spaces studied in the aforementioned works. In addition, the construction in [8] does not allow to prescribe as input a specific continuity, lower than the maximum one, in correspondence of a breakpoint.

In this paper, we focus on the subclass of the splines in [1] where at most C^1 continuity is allowed between pieces of different degrees, referred to as C^1 MD-splines. Conventional splines with non-uniform knots and arbitrary multiplicities are special instances of these splines. In fact, since the restriction to have at most C^1 continuity occurs only at the join of pieces having different degrees, the considered splines can be interpreted as conventional splines with local modifications, corresponding to the intervals where the degree changes. As we will see, this particular setup makes it possible to develop simple, stable and efficient algorithms that generalize the classical ones. In particular, a Cox de-Boor type evaluation algorithm for C^1 MD-splines was already suggested in [1] without proof. Here we reformulate that recurrence relation in a more concise and computationally efficient way and prove its correctness. In addition we provide a recurrence scheme for derivatives along with geometric design tools, such as knot insertion and removal and local degree elevation and reduction.

Working with C^1 continuity is the standard in relevant applications, such as artistic drawing and Web design. We demonstrate the utility of C^1 MD-splines by discussing their application in two different contexts. One is concerned with modeling and storing images in SVG (Scalable Vector Graphics) format, whereas the other with the design of non-uniform rational MD-splines reproducing circles and quadrics. To experience and appreciate the potential of C^1 MD-splines, we also provide a simple yet powerful Web-App based on the modeling tools proposed in this paper.

The remainder of the paper is organized as follows. Section 2 is devoted to recalling the basics on MD-splines and to presenting some preliminary material. Section 3 contains the Cox-de Boor-type recurrence relation for C^1 MD-splines and a recurrence relation for their derivatives. Efficient modeling algorithms for the considered splines are presented in Section 4, including knot insertion and degree elevation and the corresponding knot removal and degree reduction. Section 5 illustrates the construction of periodic MD-splines and finally Section 6 introduces the aforementioned applications of C^1 MD-splines. Section 7 closes the paper presenting the conclusions.

2. Multi-Degree spline spaces and B-spline bases

A multi-degree spline is a function defined on a real interval $[a, b]$ and composed of polynomial pieces of different degrees, defined on subintervals and joined at the endpoints with suitable degree of smoothness. In this section we recall the basics on such splines from [1] and we develop some preliminary material.
Let $[a, b]$ be a bounded and closed interval and $\mathcal{X} := \{x_i\}_{i=1}^q$ be a set of *breakpoints* such that $a \equiv x_0 < x_1 < \ldots < x_q < x_{q+1} \equiv b$. Let also $\mathbf{d} := \left(d_0, \ldots, d_q\right)$ be a vector of positive integers, where d_i is the degree of the polynomial defined on the interval $[x_i, x_{i+1}]$. Furthermore let us assume that two adjacent polynomial pieces join at a breakpoint x_i with continuity C^{k_i} where

$$0 \leqslant k_i \leqslant \begin{cases} \min(d_{i-1}, d_i), & \text{if} \quad d_{i-1} \neq d_i, \\ d_i - 1, & \text{if} \quad d_{i-1} = d_i, \end{cases}$$

in such a way that the vector of nonnegative integers $\mathcal{K} := \left(k_1, \ldots, k_q\right)$ contains information on the degree of smoothness at breakpoints. We can hence define a multi-degree spline space $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ as follows.

Definition 1 (Multi-degree spline space)**.** Given a partition $\mathcal{X} = \{x_i\}_{i=1}^q$ on the bounded and closed interval $[a, b]$, a vector of polynomial degrees $\mathbf{d}$ and avector $\mathcal{K}$ of degrees of smoothness, the corresponding space of *multi-degree splines* is defined as

$$\begin{aligned} S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K}) := \Big\{ f \,\Big|\, & \text{there exist } p_i \in \mathcal{P}_{d_i}, i = 0, \ldots, q, \text{ such that:} \\ & \text{i)} \;\; f(x) = p_i(x) \text{ for } x \in [x_i, x_{i+1}],\ i = 0, \ldots, q; \\ & \text{ii)} \;\; D^\ell p_{i-1}(x_i) = D^\ell p_i(x_i) \text{ for } \ell = 0, \ldots, k_i, \;\; i = 1, \ldots, q \Big\}, \end{aligned}$$

where $\mathcal{P}_{d_i}$ denotes the space of algebraic polynomials of degree d_i.

Note that if $d_i = d$, for all $i = 0, \ldots, q$, and a fixed positive integer d, then $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ is reduced to a conventional spline space.

The following result on the dimension of multi-degree spline spaces can be proved by standard arguments.

Proposition 1 (Dimension of a multi-degree spline space)**.** *The set $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ of multi-degree splines in Definition 1 is a function space of dimension*

$$K := \dim S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K}) = d_0 + 1 + \sum_{i=1}^{q} (d_i - k_i) \qquad \textit{or equivalently} \qquad K := \dim S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K}) = \sum_{i=1}^{q} (d_{i-1} - k_i) + d_q + 1.$$

The B-spline basis of a conventional spline space is classically defined on an extended partition, computed from the breakpoints and continuity conditions. In the multi-degree context, to construct an analogous basis, we will need to consider two extended partitions defined as follows.

Definition 2 (Extended partitions)**.** The set of knots $\mathbf{s} := \left\{s_j\right\}_{j=1}^{K}$ is called the *left extended partition* associated with an MD-spline space $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ if and only if:

i) $s_1 \leqslant s_2 \leqslant \ldots \leqslant s_K$;

ii) $s_{d_0+1} \equiv a$;

iii) $\{s_{d_0+2}, \ldots, s_K\} \equiv \{\underbrace{x_1, \ldots, x_1}_{d_1 - k_1 \text{ times}}, \ldots, \underbrace{x_q, \ldots, x_q}_{d_q - k_q \text{ times}}\}$.

Similarly, the set of knots $\mathbf{t} := \left\{t_j\right\}_{j=1}^{K}$ is called the *right extended partition* associated with $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ if and only if:

i) $t_1 \leqslant t_2 \leqslant \ldots \leqslant t_K$;

ii) $t_{K-d_q} \equiv b$;

iii) $\left\{t_1, \ldots, t_{K-d_q-1}\right\} \equiv \{\underbrace{x_1, \ldots, x_1}_{d_0 - k_1 \text{ times}}, \ldots, \underbrace{x_q, \ldots, x_q}_{d_{q-1} - k_q \text{ times}}\}$.

For simplicity and without loss of generality, we will limit our discussion to the case of *clamped partitions*, i.e. extended partitions with the two extreme breakpoints repeated $d_0 + 1$ times in $\mathbf{s}$, and $d_q + 1$ times in $\mathbf{t}$, that is $s_1 = \ldots = s_{d_0+1} = x_0$ and $t_{K-d_q} = \ldots = t_K = x_{q+1}$. Nevertheless the theory is readily generalizable to other partitions, including the periodic case which will be discussed in Section 5.

Multi-degree spline spaces posses a B-spline-type basis defined on the extended partitions $\mathbf{s}$ and $\mathbf{t}$ [1]. Hereinafter, we refer to the elements of a such basis as *MDB-spline basis functions* or simply *MDB-splines* and denote them by $\{N_{i,m}\}_{i=1}^{K}$, with $m := \max_{j=0,\ldots,q}\{d_j\}$. Moreover we define as usual the *support* of a function f to be the set $\text{supp}(f) = \text{closure}\{x : f(x) \neq 0\}$.

Each MDB-spline $N_{i,m}$ can be generated by an integral relation which requires to compute recursively the functions $N_{i,n}$, $n = 0, \ldots, m$, where $\text{supp}(N_{i,n}) = [s_i, t_{i-m+n}]$ and for $x \in [x_j, x_{j+1}) \subset [s_i, t_{i-m+n}]$ (with $s_i < t_{i-m+n}$):

$$N_{i,n}(x) := \begin{cases} 1, & x_j \leqslant x < x_{j+1}, & n = m - d_j, \\ \int_{-\infty}^{x} \left[\delta_{i,n-1} N_{i,n-1}(u) - \delta_{i+1,n-1} N_{i+1,n-1}(u)\right] du, & n > m - d_j, \\ 0, & & \textit{otherwise}, \end{cases} \tag{1}$$

with

$$\delta_{i,n} := \left(\int_{-\infty}^{+\infty} N_{i,n}(x) dx \right)^{-1}.$$

Note that undefined $N_{i,n}$ functions shall be intended to be identically zero and their integral at the right-hand side of (1) shall be computed as

$$\int_{-\infty}^{x} \delta_{i,n} N_{i,n}(u)\, du := \begin{cases} 0, & x < s_i, \\ 1, & x \geqslant s_i. \end{cases}$$

The above integral construction warrants that functions $N_{i,m}$ belong to the multi-degree spline space $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ and in addition yields the following properties, which highlight the analogy with respect to conventional B-spline bases.

Proposition 2 (Properties of the MDB-spline basis)**.** *The MDB-spline functions $\{N_{i,m}\}_{i=1}^{K}$ satisfy the following properties:*

i) Local Support: $N_{i,m}(x) = 0$, *for* $x \notin [s_i, t_i]$;

ii) Positivity: $N_{i,m}(x) > 0$, *for* $x \in (s_i, t_i)$;

iii) End Point: $N_{i,m}$ *vanishes exactly*

- $d_{ps_i} - \max\{j \geqslant 0 \mid s_i = s_{i+j}\}$ *times at* s_i,
- $d_{pt_i - 1} - \max\{j \geqslant 0 \mid t_{i-j} = t_i\}$ *times at* t_i,

where ps_i is the index of the breakpoint in $\mathcal{X}$ which coincides with the knot s_i and pt_i is the index of the breakpoint in $\mathcal{X}$ coinciding with t_i;

iv) Partition of unity: $\sum_{i=1}^{K} N_{i,m}(x) = 1$, *for all* $x \in [a, b]$.

Remark 1 (Uniqueness of the MDB-spline basis)**.** *Each MDB-spline $N_{i,m}$ is uniquely determined by properties i), iii), iv) together with the continuity conditions at the breakpoints contained in (s_i, t_i) (see [1] for further details).*

The existence of an MDB-spline basis $\{N_{i,m}\}_{i=1}^{K}$ with the above properties makes so that any function f in the spline space can be represented as

$$f(x) = \sum_{i=1}^{K} c_i\, N_{i,m}(x), \quad x \in [a, b],$$

and also, locally, as

$$f(x) = \sum_{i=\ell-d_j}^{\ell} c_i\, N_{i,m}(x), \quad x \in [x_j, x_{j+1}), \quad s_\ell \leqslant x_j < \min(s_{\ell+1}, b), \tag{2}$$

where we take $\min(s_{\ell+1}, b)$ to be b if $s_{\ell+1}$ is not defined or, equivalently, if $\ell + 1 > K$.

Throughout the paper we will deal with multi-degree spline spaces where functions can be either C^0 or C^1 at the breakpoints separating intervals of different degrees. These spaces are instances of the following definition for $k = 0, 1$.

Definition 3 (C^k multi-degree spline space)**.** A function space $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ as in Definition 1 is said to be a C^k *MD-spline space* if its functions are at most C^k continuous at the joins between pieces of different degrees. Therefore it is defined by

$$S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K}), \quad \text{with} \quad \mathcal{K} = (k_0, k_1, \ldots, k_q) \quad \text{and} \quad 0 \leqslant k_i \leqslant k \quad \text{if} \quad d_{i-1} \neq d_i, \quad i = 1, \ldots, q.$$

The corresponding MDB-spline basis is referred to as a C^k *MDB-spline basis*.

It is immediate to observe that the integral relation (1), when applied in the context of a conventional spline space, generates the B-spline basis. Moreover, for our future purposes, it is convenient to characterize the functions produced by (1) when the degrees are the same just locally. To this aim we introduce the following definition and remark.

Definition 4 (Conventional MDB-splines)**.** A function f in a C^k MD-spline space $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ such that $\text{supp}(f) = [x_r, x_{r+j}]$ is said to be *conventional* when all breakpoint intervals included in its support have associated equal degree, namely when $d_\ell = d,$ for all $\ell = r, \dots, r + j - 1$.

Remark 2. *The following properties of the MDB-spline basis, that hold for locally uniform degrees, are an immediate consequence of the integral construction* (1)*:*

i) A conventional MDB-spline $N_{i,m}$, such that $d_\ell = d$, $\ell = ps_i, \dots, pt_{i-1}$, is a B-spline of a conventional spline space of degree d. In other words, there exists a conventional spline space (defined on the same breakpoint sequence) where a B-spline basis function is identical to $N_{i,m}$ on $[s_i, t_i] \equiv [x_{ps_i}, x_{pt_i}]$.

ii) A C^0 MDB-spline basis is a conventional B-spline basis piecewisely. In particular, let x_i and x_j be two breakpoints with associated C^0 continuity and let $d_\ell = d$, for $\ell = i, \dots, j-1$. Then the C^0 MDB-spline basis, restricted to $[x_i, x_j]$, is a conventional B-spline basis of degree d .

In the following we will implicitly assume that $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ is a C^1 MD-spline space where $k_i = 1$, if $d_i \neq d_{i+1}$, $i = 1, \dots, q$, in such a way that the functions $\{N_{i,m-1}\}_{i=2}^K$, generated at the last but one iteration of (1), be continuous and thus belong to a C^0 MD-spline space. The results developed in this paper are nevertheless valid with any continuity $k_i \in \{0, 1\}$. In fact, if a spline space presents C^0 continuity at any breakpoint x_i, we will simply split it into two spaces with all interior continuities greater than zero and continue separately in each of these MD-spline spaces. Under this assumption, it can easily be seen that the set $\{N_{i,m-1}\}_{i=2}^K$ is a C^0 MDB-spline basis. Moreover, the support of each $N_{i,m-1}$ is characterized by the following property, which holds more generally for any MDB-spline basis function of a C^0 MD-spline space.

Proposition 3. *Let $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ be a C^1 multi-degree spline space. Each C^0 MDB-spline $N_{i,m-1}$, $i = 2 \dots, K$, generated at the last but one iteration of the integral relation* (1)*, is either a conventional B-spline or it is nonzero on at most three consecutive breakpoint intervals.*

Proof. We proceed by contradiction and suppose that there exists a C^0 MDB-spline $N_{i,m-1}$ which is nonzero on four consecutive breakpoint intervals with associated different degrees. The support $[s_i, t_{i-1}]$ of $N_{i,m-1}$ depends on the continuity at the breakpoints in such a way that the higher the continuity, the larger the number of breakpoint intervals included in it. We will thus consider a local configuration which allows us to maximize the support, corresponding to a degree vector $\mathbf{d} = (d, d, d, 0)$, a continuity vector $\mathcal{K} = (d-1, d-1, 0)$, C^{d-1} continuity at s_i and C^0 continuity at t_{i-1}. The total number of degrees of freedom provided by the configuration is $3d + 4$, whereas the number of continuity conditions is $3d + 2$. This is in contradiction with a necessary condition for the existence and uniqueness of the MDB-spline basis functions, which requires to have as many degrees of freedom as the number of continuity conditions plus one (where the additional condition arises from the partition of unity property). Therefore there cannot exist a nonconventional $N_{i,m-1}$ which is nonzero on four consecutive breakpoint intervals. The same count of constraints versus degrees of freedom shows that functions $N_{i,m-1}$ having support on three breakpoint intervals are uniquely determined. A function of this kind can be derived by considering degrees $\mathbf{d} = (d, 0, d)$, continuities $\mathcal{K} = (0, 0)$ and C^{d-1} continuity at the endpoints of the support, s_i and t_{i-1}. In fact, the latter configuration fulfills all the conditions for the existence and uniqueness of MDB-spline basis functions recalled in Remark 1. □

Remark 3. *Proposition 3 entails that the support of a nonconventional $N_{i,m-1}$ can include either two or three breakpoint intervals (if the support consists of one breakpoint interval only, then $N_{i,m-1}$ is necessarily conventional). The same arguments in the proof of Proposition 3, together with the endpoint property of MDB-splines, allow us to further characterize nonconventional $N_{i,m-1}$ functions. In particular, in the former case, supposed* $\text{supp}(N_{i,m-1}) = [x_{r-1}, x_{r+1}]$*, $N_{i,m-1}$ is the last Bernstein basis function of degree $d_{r-1} - 1$ relative to $[x_{r-1}, x_r]$ and the first Bernstein basis function of degree $d_r - 1$ relative to $[x_r, x_{r+1}]$. If instead* $\text{supp}(N_{i,m-1}) = [x_{r-1}, x_{r+2}]$*, then $N_{i,m-1}$ must necessarily have degree zero on $[x_r, x_{r+1}]$. Moreover it coincides with the last and first Bernstein basis function of degree $d_{r-1} - 1$ and $d_{r+1} - 1$ relative to the first and last breakpoint interval, respectively. The fact that $N_{i,m-1}$ can only have these two particular forms also entails that on a breakpoint interval there can be at most two nonconventional nonvanishing functions of level $m - 1$. All these observations will be relevant for proving the correctness of the Cox-de Boor-type recurrence relation presented in the following section.*

3. Recurrence schemes for C^1 MD-splines and their derivatives

In the reminder of the paper we restrict our attention to the class of C^1 MD-splines and accordingly, from now on, we indicate by $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ a space of such splines, rather than a generic one. We will show that the considered splines can be evaluated by efficient and stable recurrence schemes, similar to those available in the conventional setting.

3.1. A Cox-de Boor-type recurrence relation for C^1 MD-splines

The main result of this section is the following proposition, which provides a recurrence scheme for the evaluation of C^1 MD-splines. A proof will be given at the end of the section, since it requires some preliminary results.

Proposition 4 (Cox-de Boor-type recurrence relation)**.** *Let $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ be a C^1 MD-spline space on $[a, b]$. Let also $\widetilde{\mathcal{X}} := \{\tilde{x}_j\}_{j=1}^{q}$ together with $\tilde{x}_0$ and $\tilde{x}_{q+1}$ be obtained by scaling and translating $\mathcal{X}$, x_0 and x_{q+1} as follows:*

$$\tilde{x}_0 = 0 \qquad \text{and} \qquad \tilde{x}_j = \sum_{i=1}^{j} \frac{x_i - x_{i-1}}{d_{i-1}}, \quad j = 1, \dots, q+1.$$

In addition, let us consider the extended partitions $\tilde{\mathbf{s}} = \{\tilde{s}_j\}_{j=1}^{K}$ and $\tilde{\mathbf{t}} = \{\tilde{t}_j\}_{j=1}^{K}$ obtained from $\widetilde{\mathcal{X}}$ according to Definition 2 and let us set $m := \max_{j=0,\dots,q}\{d_j\}$. Then an MDB-spline $N_{i,m}$, $i = 1, \dots, K$, can be computed on the breakpoint interval $[x_j, x_{j+1})$ contained in its support $[s_i, t_i]$ by the following recurrence scheme:

$$\begin{aligned} N_{\ell,m-d_j}(x) &= \begin{cases} 1, & x_j \leqslant x < x_{j+1}, \quad \text{for} \quad \ell \quad \text{such that} \quad s_\ell \leqslant x_j < \min(s_{\ell+1}, b), \\ 0, & \text{otherwise}, \end{cases} \\ N_{i,n}(x) &= \phi_i^{n-1}(x) N_{i,n-1}(x) + \left(1 - \phi_{i+1}^{n-1}(x)\right) N_{i+1,n-1}(x), \quad n = m - d_j + 1, \dots, m, \end{aligned} \tag{3}$$

with

$$\phi_i^{n-1}(x) = \frac{\tilde{x} - \tilde{s}_i}{\tilde{t}_{i-m+n-1} - \tilde{s}_i}, \tag{4}$$

and where $\tilde{x}$ is the scaled and translated version of $x \in [x_j, x_{j+1})$ according to:

$$\tilde{x} = \tilde{x}_j + \frac{x - x_j}{d_j}. \tag{5}$$

Note that undefined $N_{i,n}$ functions in (3) shall be regarded as the zero function, and $0/0$ in (4) shall be intended as 0.

Remark 4. *The computation of each nonvanishing $N_{i,m}$ at $x \in [x_j, x_{j+1})$ requires precisely d_j steps of the recurrence scheme (3)–(4).*

Functions $N_{i,m}$ generated by (3)–(4) have by construction $\mathrm{supp}(N_{i,m}) = [s_i, t_i]$. In the following we will prove that they are a C^1 MDB-spline basis and in particular that (3)–(4) and (1) generate the same set of functions on the extended partitions $\mathbf{s}$ and $\mathbf{t}$. To this end, we observe below that the proposed Cox-de Boor-type recurrence relation generates all the C^1 MDB-splines that are of conventional type. This is a consequence of the fact that, when the degrees are locally (or globally) uniform, equations (3)–(4) are reduced to the classical Cox-de Boor recurrence scheme [13, 14], which we will also refer to as the *B-spline recurrence relation*.

Lemma 1. *Let $N_{i,m}$ be a function generated by the recurrence scheme in Proposition 4. Moreover let us assume that all the breakpoint intervals included in $\mathrm{supp}(N_{i,m}) = [s_i, t_i]$ have same associated degree, namely that $d_j = d$, for all $j = d_{ps_i}, \dots, d_{pt_i - 1}$. Then $N_{i,m}$ is an element of the C^1 MDB-spline basis.*

Proof. Under the given assumption, all the functions ϕ_j^{n-1}, for $n = m - d + 1, \dots, m$, involved in the computation of $N_{i,m}$, can be rewritten so as to take the same expression as in the standard B-spline recurrence relation, namely $\phi_j^{n-1}(x) = (x - s_i)/(t_{i-m+n-1} - s_i)$, for $x \in \mathrm{supp}(N_{j,n-1}) = [s_i, t_{i-m+n-1}]$. This is a simple consequence of the fact that the denominator of each ϕ_j^{n-1} corresponds to the support of the associated $N_{j,n-1}$ in (3) and of Remark 4. Thus $N_{i,m}$ is a B-spline of a conventional space and, in view of Remark 2-*i)*, is an element of the C^1 MDB-spline basis. □

The following fact is an immediate consequence of the above lemma and of Remark 4.

Remark 5. *When $d_j = d$, for all $j = 0, \dots, q$, the recurrence scheme (3)–(4) generates the B-spline basis of a conventional degree-d spline space.*

Let us now consider the special case of spaces of C^0 multi-degree splines. In this case, it is easy to understand that the MDB-spline basis can be obtained by locally applying the classical B-spline recurrence relation. The following result shows that the proposed Cox-de Boor-type recurrence scheme will also generate this basis.

Lemma 2. *If $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ is a C^0 MD-spline space on $[a, b]$, then the recurrence scheme in Proposition 4 generates its MDB-spline basis.*

Proof. Let x_i and x_j be two breakpoints with associated C^0 continuity and let $d_\ell = d$, for $\ell = i, \dots, j-1$. Then, there will be one knot of multiplicity d in $\mathbf{s}$ corresponding to x_i and one knot of multiplicity d in $\mathbf{t}$ corresponding to x_j. In view of Remarks 5 and 2-*ii)*, in the restriction of the space to $[x_i, x_j]$ the scheme (3)–(4) generates a conventional B-spline basis of degree d, which is the C^0 MDB-spline basis restricted to $[x_i, x_j]$. Note that the B-spline recurrence relation, in a clamped partition, does not depend on the location of the first and last knot, that is on the location of the first knot in $\mathbf{s}$ and of the last knot in $\mathbf{t}$ and therefore those knots can be undefined. □

An important consequence of the above lemma is that, if $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ is a C^1 MD-spline space, then the last but one step of the recurrence scheme (3)–(4) generates a C^0 MDB-spline basis $\{N_{i,m-1}\}_{i=2}^{K}$. From this one can also see that the functions $\{N_{i,m}\}_{i=1}^{K}$ produced by (3) enjoy the partition of unity property. The latter two facts are among the key ingredients of the following proof.

Proof of Proposition 4. In the following we will show that the recurrence scheme (3)–(4) and the integral relation (1) generate the same set of functions $\{N_{i,m}\}_{i=1}^{K}$. Due to Lemmas 1 and 2, this is already known to be true for all functions $N_{i,m}$ of conventional type and for the set of functions $\{N_{i,m-1}\}_{i=2}^{K}$ generated at the last but one iteration. Our analysis can therefore be limited to nonconventional $N_{i,m}$ functions only.

The remainder of the proof is organized as follows. *Parts I and II* are devoted to legitimate the definition of functions ϕ_i^{m-1} in (4). In particular, in *Part I* we show that, whenever a function $N_{i,m-1}$ at the right-hand side of (3) is nonconventional, the corresponding ϕ_i^{m-1} shall be computed as

$$\phi_i^{m-1}(x) = \begin{cases} \dfrac{1}{\delta}\dfrac{x - x_{ps_i}}{d_{ps_i}}, & x \in [x_{ps_i}, x_{ps_i+1}], \\ \dfrac{1}{\delta}\left(\dfrac{x_{ps_i+1} - x_{ps_i}}{d_{ps_i}} + \dfrac{x - x_{ps_i+1}}{d_{ps_i+1}}\right), & x \in [x_{ps_i+1}, x_{ps_i+2}], \\ \dfrac{1}{\delta}\left(\dfrac{x_{ps_i+1} - x_{ps_i}}{d_{ps_i}} + \dfrac{x_{ps_i+2} - x_{ps_i+1}}{d_{ps_i+1}} + \dfrac{x - x_{ps_i+2}}{d_{ps_i+2}}\right), & x \in [x_{ps_i+2}, x_{pt_{i-1}}], \end{cases} \qquad \delta := \sum_{j=ps_i}^{pt_{i-1}-1} \frac{x_{j+1} - x_j}{d_j}. \tag{6}$$

The fact that each ϕ_i^{m-1} considered above needs to be defined on at most three breakpoint intervals is a consequence of Proposition 3 and guarantees that formula (6) covers all possible scenarios. We remind in passing that any conventional $N_{i,m-1}$ may instead have arbitrary support width and that the associated ϕ_i^{m-1} is defined as in the standard B-spline recurrence relation, namely as $\phi_i^{m-1}(x) = (x - x_{ps_i})/(x_{pt_{i-1}} - x_{ps_i})$, $x \in [x_{ps_i}, x_{pt_{i-1}}]$ (see Lemma 1). We also recall that formula (6) was originally presented in [1], even though without proof. In *Part II* we show that (6) and (4) are equivalent expressions of the functions ϕ_i^{m-1}. Finally, in *Part III* we prove that the (nonvanishing) nonconventional $N_{i,m}$ functions generated by (3)–(4) and (1) are the same on any breakpoint interval $[x_r, x_{r+1}]$.

Part 1. We shall define each function ϕ_i^{m-1} on the support of the corresponding nonconventional $N_{i,m-1}$ at the right-hand side of (3). Due to Proposition 3, in particular, $\mathrm{supp}(N_{i,m-1}) = [s_i, t_{i-1}]$ may include either two or three breakpoint intervals.

On account of Remark 3, if $\mathrm{supp}(N_{i,m-1})$ includes two breakpoint intervals, then $[s_i, t_{i-1}] \equiv [x_{ps_i}, x_{ps_i+2}]$ and $k_{ps_i+1} = 1$ (Figure 1(a) illustrates this situation in full generality). Since $N_{i+1,m-1}(x) = 0$, for $x \leqslant x_{ps_i+1} \leqslant s_{i+1}$, from (3) we obtain

$$\phi_i^{m-1}(x) = \frac{N_{i,m}(x)}{N_{i,m-1}(x)}, \quad x \in [x_{ps_i}, x_{ps_i+1}]. \tag{7}$$

Function $N_{i,m}$ has degree d_{ps_i} on $[x_{ps_i}, x_{ps_i+1}]$ and vanishes d_{ps_i} times at s_i, whereas $N_{i,m-1}$ has degree $d_{ps_i} - 1$ and vanishes $d_{ps_i} - 1$ times at the same point. There follows that ϕ_i^{m-1} is a linear function and that $\phi_i^{m-1}(s_i) = 0$. Similarly, recalling that $N_{i-1,m-1}(x) = 0$, for $x \geqslant x_{ps_i+1} \geqslant t_{i-2}$, and using (3) with $N_{i-1,m}$ at the left-hand side we obtain

$$1 - \phi_i^{m-1}(x) = \frac{N_{i-1,m}(x)}{N_{i,m-1}(x)}, \quad x \in [x_{ps_i+1}, x_{ps_i+2}]. \tag{8}$$

In $[x_{ps_i+1}, x_{ps_i+2}]$ the functions $N_{i-1,m}$ and $N_{i,m-1}$ have degrees d_{ps_i+1} and $d_{ps_i+1} - 1$, respectively; in addition, they have a zero of multiplicity $d_{ps_i+1} + 1$ and d_{ps_i+1}, respectively, at t_{i-1}. Therefore ϕ_i^{m-1} is a linear function and $\phi_i^{m-1}(t_{i-1}) = 1$.

We will now show that ϕ_i^{m-1} is continuous at x_{ps_i+1} and compute its value. To this aim, note that $N_{i,m-1}$ is C^0 at x_{ps_i+1} and thus $N_{i,m-1}(x_{ps_i+1}) = 1$. Using the integral definition of the MDB-spline basis in (1), equation (7) yields

$$\phi_i^{m-1}(x_{ps_i+1}^-) = N_{i,m}(x_{ps_i+1}) = \frac{\int_{x_{ps_i}}^{x_{ps_i+1}} N_{i,m-1}(x)\,dx}{\int_{x_{ps_i}}^{x_{ps_i+2}} N_{i,m-1}(x)\,dx},$$

and equation (8) yields

$$\phi_i^{m-1}(x_{ps_i+1}^+) = 1 - N_{i-1,m}(x_{ps_i+1}) = 1 - 1 + \frac{\int_{x_{ps_i}}^{x_{ps_i+1}} N_{i,m-1}(x)\,dx}{\int_{x_{ps_i}}^{x_{ps_i+2}} N_{i,m-1}(x)\,dx}.$$

From the two above identities one can see that ϕ_i^{m-1} is continuous at x_{ps_i+1}. To compute $\phi_i^{m-1}(x_{ps_i+1})$ we shall recall that $N_{i,m-1}$ is nothing else than the last Bernstein basis function of degree $d_{ps_i} - 1$ relative to $[x_{ps_i}, x_{ps_i+1}]$ and the first Bernstein basis function of degree $d_{ps_i+1} - 1$ relative to $[x_{ps_i+1}, x_{ps_i+2}]$ (see Remark 3). This yields

$$\phi_i^{m-1}(x_{ps_i+1}) = \frac{x_{ps_i+1} - x_{ps_i}}{d_{ps_i}} \left(\frac{x_{ps_i+1} - x_{ps_i}}{d_{ps_i}} + \frac{x_{ps_i+2} - x_{ps_i+1}}{d_{ps_i+1}} \right)^{-1}.$$

The expression of ϕ_i^{m-1} on $[x_{ps_i}, x_{ps_i+2}]$ can thus explicitly be computed using the fact that ϕ_i^{m-1} is piecewise linear, continuous and interpolates known values at x_{ps_i}, x_{ps_i+1}, x_{ps_i+2}. This yields the first and second line of equation (6).

If instead $\mathrm{supp}(N_{i,m-1})$ includes three breakpoint intervals, then $[s_i, t_{i-1}] \equiv [x_{ps_i}, x_{ps_i+3}]$ and, again on account of Remark 3, we shall have $d_{ps_i+1} = 1$ and C^1 continuity at the breakpoints (this circumstance is illustrated with full generality in Figure 1(b)). Similarly as above, we can prove that ϕ_i^{m-1} is a linear and continuous function, that $\phi_i^{m-1}(x_{ps_i}) = 0$, $\phi_i^{m-1}(x_{ps_i+3}) = 1$ and we can compute the values of ϕ_i^{m-1} at x_{ps_i+1} and x_{ps_i+2}. Constructing ϕ_i^{m-1} by linear interpolation we hence obtain the first, second and third line of (6).

In view of *Part 3*, it is useful to observe that, due to the construction of functions ϕ_i^{m-1}, the scheme (3)–(4) will generate two pieces of functions

$$N_{i,m}(x),\ x \in [x_{ps_i}, x_{ps_i+1}] \qquad \text{and} \qquad N_{i-1,m}(x),\ x \in [x_{pt_{i-1}-1}, x_{pt_{i-1}}], \tag{9}$$

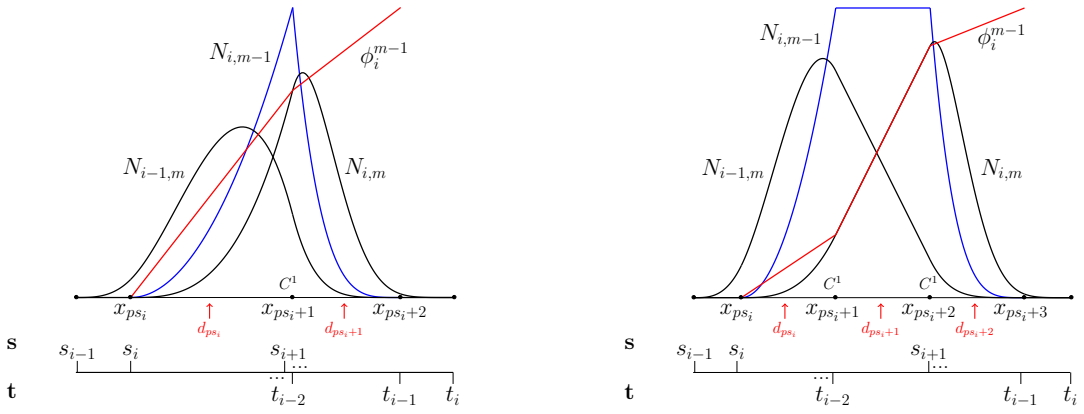
coinciding with those defined by the integral relation (1). Therefore, the above indicated pieces of functions $N_{i,m}$ and $N_{i-1,m}$ belong *by construction* to the C^1 MDB-spline basis. In particular, $N_{i,m}$ is the function which "starts" in x_{ps_i} with the maximum continuity $C^{d_{ps_i}-1}$, whereas $N_{i-1,m}$ is the function which "terminates" in $x_{pt_{i-1}}$ with the maximum continuity $C^{d_{pt_{i-2}}-1}$.

Part 2. We show that (4) and (6) are equivalent formulations to compute functions ϕ_i^{m-1} associated with nonconventional $N_{i,m-1}$. In particular, when $n = m$, equation (4) becomes

$$\phi_i^{m-1}(x) = \frac{\tilde{x} - \tilde{s}_i}{\tilde{t}_{i-1} - \tilde{s}_i}, \tag{10}$$

and rewriting δ in (6) in terms of the extended partitions $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{t}}$ we get

$$\delta = \sum_{j=ps_i}^{pt_{i-1}-1} \frac{x_{j+1} - x_j}{d_j} = \tilde{x}_{pt_{i-1}} - \tilde{x}_{ps_i} = \tilde{t}_{i-1} - \tilde{s}_i, \tag{11}$$



(a) (b)

Figure 1: The two possible types of nonconventional $N_{i,m-1}$ (in blue) that can be generated at the last but one level of the Cox-de Boor-type recurrence relation (see Remark 3) and the corresponding configurations to be considered in the proof of Proposition 4.

which yields the denominator of (10). As for the numerator $\tilde{x} - \tilde{s}_i$, we shall recall that $\tilde{s}_i = \tilde{x}_{ps_i}$ and address the following three cases, which correspond to each line of (6).

If $x \in [x_{ps_i}, x_{ps_i+1}]$, then $\tilde{x} = \tilde{x}_{ps_i} + (x - x_{ps_i})/d_{ps_i}$ and thus

$$\tilde{x} - \tilde{s}_i = \tilde{x}_{ps_i} + \frac{x - x_{ps_i}}{d_{ps_i}} - \tilde{x}_{ps_i} = \frac{x - x_{ps_i}}{d_{ps_i}}. \tag{12}$$

If $x \in [x_{ps_i+1}, x_{ps_i+2}]$, then $\tilde{x} = \tilde{x}_{ps_i+1} + (x - x_{ps_i+1})/d_{ps_i+1}$ and thus

$$\tilde{x} - \tilde{s}_i = \tilde{x}_{ps_i+1} + \frac{x - x_{ps_i+1}}{d_{ps_i+1}} - \tilde{x}_{ps_i} = \sum_{i=1}^{ps_i+1} \frac{x_i - x_{i-1}}{d_{i-1}} + \frac{x - x_{ps_i+1}}{d_{ps_i+1}} - \sum_{i=1}^{ps_i} \frac{x_i - x_{i-1}}{d_{i-1}} = \frac{x_{ps_i+1} - x_{ps_i}}{d_{ps_i}} + \frac{x - x_{ps_i+1}}{d_{ps_i+1}}. \tag{13}$$

Finally, if $x \in [x_{ps_i+2}, x_{pt_{i-1}}] \equiv [x_{ps_i+2}, x_{ps_i+3}]$, then $\tilde{x} = \tilde{x}_{ps_i+2} + (x - x_{ps_i+2})/d_{ps_i+2}$ and

$$\tilde{x} - \tilde{s}_i = \tilde{x}_{ps_i+2} + \frac{x - x_{ps_i+2}}{d_{ps_i+2}} - \tilde{x}_{ps_i} = \sum_{i=1}^{ps_i+2} \frac{x_i - x_{i-1}}{d_{i-1}} + \frac{x - x_{ps_i+2}}{d_{ps_i+2}} - \sum_{i=1}^{ps_i} \frac{x_i - x_{i-1}}{d_{i-1}} = \frac{x_{ps_i+1} - x_{ps_i}}{d_{ps_i}} + \frac{x_{ps_i+2} - x_{ps_i+1}}{d_{ps_i+1}} + \frac{x - x_{ps_i+2}}{d_{ps_i+2}}. \tag{14}$$

The equivalence of formulas (10) and (6) straightforwardly follows from the above equalities (11), (12), (13) and (14).

Part 3. In the following we consider a generic breakpoint interval $[x_r, x_{r+1}]$ and show that the recurrence scheme (3)–(4) generates the pieces of all the $d_r + 1$ MDB-splines that are nonvanishing on it. In particular, we will only study functions $N_{i,m}$ which (with a slight abuse of terminology) are nonconventional, since, due to Lemma 1, we can already conclude that all the others are correct.

The functions we are concerned with are generated by (3) when either one of $N_{i,m-1}$ or $N_{i+1,m-1}$, or both, are nonconventional. Hence, since on $[x_r, x_{r+1}]$ there can be at most two nonconventional functions at level $m - 1$ (see Remark 3), there will be at most four such functions at level m. We can confine ourselves to proving the correctness of all but one of them, as the remaining one can be handled by the partition of unity property of the set $\{N_{i,m}\}_{i=1}^{K}$, recalled earlier in this section.

As a consequence, if there is just one nonconventional $N_{i,m}$ on $[x_r, x_{r+1}]$, then there is nothing to prove – this happens when either one of $N_{i,m-1}$ or $N_{i+1,m-1}$ is nonconventional and its support does not include $[x_r, x_{r+1}]$.

If instead on $[x_r, x_{r+1}]$ there is one (nonvanishing) nonconventional function at level $m - 1$, or there are two of them with successive i-indices, at level m these will give rise to two or three nonconventional functions, respectively.

As pointed out at the end of *Part 1*, on any breakpoint interval as many pieces of nonconventional $N_{i,m}$ functions as the number of nonconventional $N_{i,m-1}$ turn out to be correct by construction. On this account, also in this case, we can conclude that all the functions $N_{i,m}$ into analysis will be correct.

Therefore, there only remains to be studied the case in which there are four level-m nonconventional functions on $[x_r, x_{r+1}]$, which may only occur when

$$d_{r-1} \neq d_r, \quad d_r \neq d_{r+1}, \quad d_r \geqslant 3, \quad k_r = k_{r+1} = 1. \tag{15}$$

The above configuration results in two nonconventional $N_{i,m-1}$ and $N_{j+1,m-1}$, such that $j > i$, and four corresponding nonconventional $N_{i-1,m}$, $N_{i,m}$, $N_{j,m}$, $N_{j+1,m}$. Of these, $N_{i-1,m}$ and $N_{j+1,m}$ have the maximum continuity C^{d_r-1} at x_{r+1} and x_r, respectively, and thus their correctness is ensured by the construction of ϕ_i^{m-1} and ϕ_{j+1}^{m-1} (see (9) and related discussion). Moreover, since at x_r there is a knot in $\mathbf{s}$ of multiplicity $\geqslant 2$, whereas at x_{r+1} there is a knot in $\mathbf{t}$ of multiplicity $\geqslant 2$, the functions $N_{i,m}$ and $N_{j,m}$ have a zero of multiplicity $d_r - 2$ at x_{r+1} and x_r, respectively. We will next focus on studying one of these two functions which, without loosing generality, we can suppose to be $N_{j,m}$. The verification of the correctness of $N_{j,m}$ will conclude the proof.

Under the assumed configuration (15), $N_{j,m}$ is a combination of a conventional $N_{j,m-1}$ and a nonconventional $N_{j+1,m-1}$. Hence $N_{j,m}$ is a continuous function and we will now show that it is C^1-continuous at x_{r+1}. In particular, $N_{j,m-1}$ and $N_{j+1,m-1}$ are respectively the last but one and last Bernstein basis functions of degree $d_r - 1$ relative to $[x_r, x_{r+1}]$. Moreover, $N_{j+1,m-1}$ is the first Bernstein basis function of degree $d_{r+1} - 1$ relative to $[x_{r+1}, x_{r+2}]$ (and $N_{j,m-1}$ is identically zero). As a consequence, $\phi_j^{m-1}(x) = \frac{x-x_r}{x_{r+1}-x_r}$, $x \in [x_r, x_{r+1}]$. In addition, constructiong ϕ_{j+1}^{m-1} as in *Part I*, we obtain[1]

$$\phi_{j+1}^{m-1}(x) = \begin{cases} \dfrac{x - x_r}{x_{r+1} - x_r} c_r, & x \in [x_r, x_{r+1}], \\ \dfrac{x_{r+2} - x}{x_{r+2} - x_{r+1}} c_r + \dfrac{x - x_{r+1}}{x_{r+2} - x_{r+1}} c_{r+1}, & x \in [x_{r+1}, x_{r+2}], \end{cases}$$

with

$$\begin{cases} c_r = \dfrac{\delta_r}{\delta_r + \delta_{r+1}} & \text{and} \quad c_{r+1} = 1, \qquad \text{if } \operatorname{supp}(N_{j+1,m-1}) = [x_r, x_{r+2}], \\ c_r = \dfrac{\delta_r}{\delta_r + \delta_{r+1} + \delta_{r+2}} & \text{and} \quad c_{r+1} = \dfrac{\delta_r + \delta_{r+1}}{\delta_r + \delta_{r+1} + \delta_{r+2}}, \qquad \text{if } \operatorname{supp}(N_{j+1,m-1}) = [x_r, x_{r+3}], \end{cases}$$

and $\delta_r := (x_{r+1} - x_r)/d_r$. Substituting the explicit expressions at the right-hand side of (3), one can directly verify that, at x_{r+1}, the left and right first derivatives of $N_{j,m}$ are identical. In addition, the recurrence scheme (3)–(4) guarantees that $N_{j,m}$ has a zero of multiplicity $d_r - 2$ at x_r and hence satisfies the endpoint property in Proposition 2-*iii)*.

Finally, we observe that all the pieces of $N_{j,m}$ on $[s_i, t_i] \setminus [x_r, x_{r+1}]$ turn out to be correct, either by construction or by the aforementioned partition of unity argument. In fact, if this was not true, then there would be another interval $[x_p, x_{p+1}] \subset [s_i, t_i] \setminus [x_r, x_{r+1}]$ on which there are four nonconventional functions such that $N_{j,m}$ is either the second or third of them. It is easy to see that this circumstance can never occur.

We can thus conclude that $N_{j,m}$ is the unique function in the spline space satisfying the properties in Proposition 2 (see also Remark 1) and thus is a C^1 MDB-spline basis function. This concludes the proof that $N_{j,m}$ is correct on $[x_r, x_{r+1}]$ and the proof of the proposition. □

Remark 6. *We observe that, when $n < m$, each function $\phi_i^n(x)$ in (4) would have the same values if defined on* $\mathbf{s}$ *and* $\mathbf{t}$ *(see Lemmas 5 and 2). This allowed us to formulate the recurrence scheme (3)–(4) in terms of the scaled and translated partitions both at each step $n < m$ and at the last step $n = m$, and in the latter case both for conventional B-splines and for MDB-splines having pieces of different degrees, obtaining a simplified version of the Cox-de Boor-type evaluation scheme in [1].*

Remark 7. *It should be recalled that a similar scaling and translation strategy was firstly proposed in [8], though within a different setting. In that paper, however, the scaled and translated knot intervals must necessarily be used*

[1] Note that ϕ_{j+1}^{m-1} has been constructed only on the two intervals where it needs to be evaluated and recall that $d_{r+1} = 1$ in case $\operatorname{supp}(N_{j+1,m-1}) = [x_r, x_{r+3}]$, as observed in Remark 3.

at each recurrence step to guarantee that the generated curves will be C^1 continuous between segments of different degrees.

Example 1. This example illustrates the use of recurrence scheme (3)–(4) for the evaluation of all MDB-splines nonvanishing at a given point. To this end, consider the C^1 MD-spline space $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ defined on $[0, 9]$, with $\mathcal{X} = \{3, 6, 7\}$, $\mathbf{d} = (3, 3, 1, 2)$ and $\mathcal{K} = (2, 1, 1)$. The sequence of breakpoint intervals and the associated extended partitions $\mathbf{s}$ and $\mathbf{t}$ are illustrated in Figure 2, together with the degree of the polynomial segments and the continuity at breakpoints.

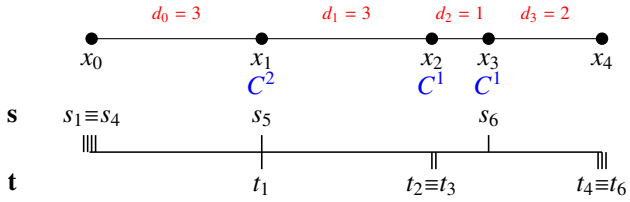


Figure 2: Setting for the spline space in Example 1.

Prior to starting the evaluation procedure, the scaled and translated partition $\tilde{\mathcal{X}} = \{1, 2, 3\}$, $[\tilde{x}_0, \tilde{x}_{q+1}] = [0, 4]$ and the extended partitions $\tilde{\mathbf{s}} = \{0, 0, 0, 0, 1, 3\}$ and $\tilde{\mathbf{t}} = \{1, 2, 2, 4, 4\}$ are computed.

We now aim to evaluate a C^1 MD-spline f expanded in the MDB-spline basis with coefficients $\{c_i\}_{i=1}^{K}$. Taking the evaluation point to be $\bar{x} = 4.5$, the steps of the scheme are:

1. determine j such that $\bar{x} \in [x_j, x_{j+1})$; in the considered case, this yields $j = 1$;
2. determine ℓ such that $s_\ell \leqslant x_j < \min(s_{\ell+1}, b)$; in the considered case, this yields $\ell = 5$;
3. determine $\tilde{\bar{x}}$ from equation (5), obtaining $\tilde{\bar{x}} = 1.5$.

For $m = 3$, $j = 1$, $d_j = 3$, $\ell = 5$, the function $N_{\ell,m-d_j}$ in equation (3) corresponds to $N_{5,0}$, which is identically equal to one in $[x_j, x_{j+1}) = [x_1, x_2)$. Next, for $n = m - d_j + 1, \ldots, m = 1, 2, 3$, the scheme (3)–(4) applied at $\bar{x}$ yields

$$\begin{array}{cccccc} 0 & 0 & 0 & 0 & N_{5,0}(\bar{x}) & 0 \\ 0 & 0 & 0 & N_{4,1}(\bar{x}) & N_{5,1}(\bar{x}) & 0 \\ 0 & 0 & N_{3,2}(\bar{x}) & N_{4,2}(\bar{x}) & N_{5,2}(\bar{x}) & 0 \\ 0 & N_{2,3}(\bar{x}) & N_{3,3}(\bar{x}) & N_{4,3}(\bar{x}) & N_{5,3}(\bar{x}) & 0 \end{array}$$

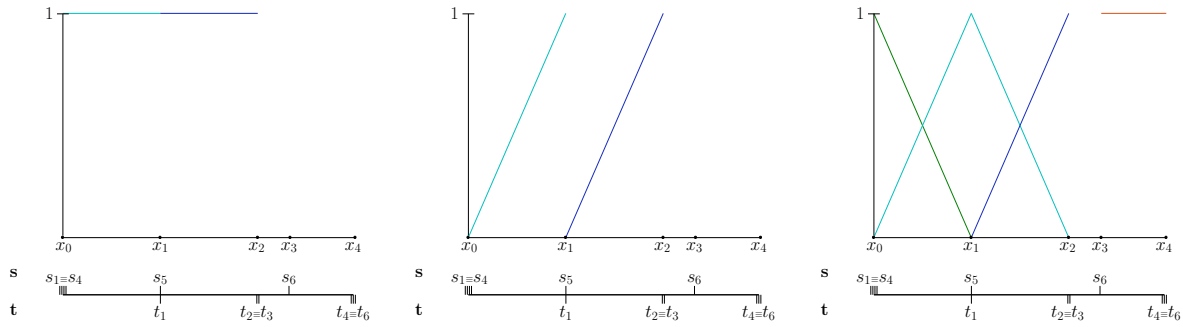
and hence

$$f(\bar{x}) = \sum_{i=2}^{5} c_i \, N_{i,3}(\bar{x}).$$

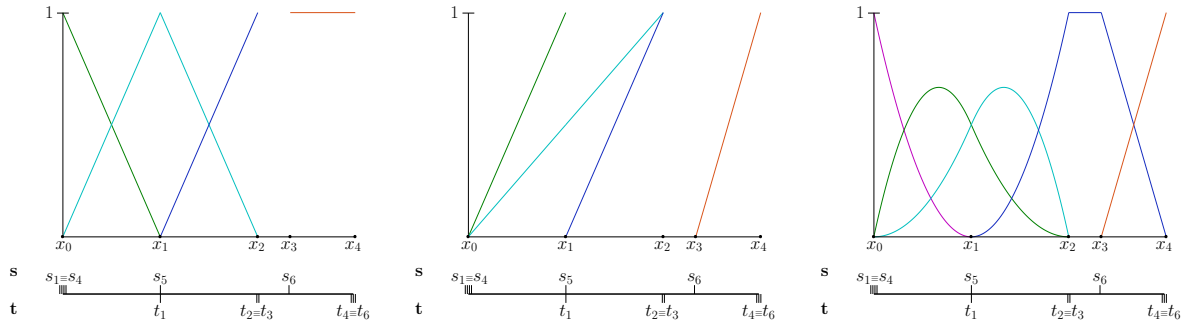
Moving the evaluation point to $\bar{x} = 6.5 \in [x_2, x_3)$ (corresponding to $\tilde{\bar{x}} = 2.5$), we get $m = 3$, $j = 2$, $d_j = 1$ and $\ell = 5$ and thus $N_{\ell,m-d_j}$ corresponds to $N_{5,2}$, which is identically one in $[x_j, x_{j+1}) = [x_2, x_3)$. For $n = m - d_j + 1, \ldots, m = 3$, the scheme (3)–(4) applied at $\bar{x} = 6.5$ yields

$$\begin{array}{cccc} 0 & 0 & N_{5,2}(\bar{x}) & 0 \\ 0 & N_{4,3}(\bar{x}) & N_{5,3}(\bar{x}) & 0. \end{array}$$

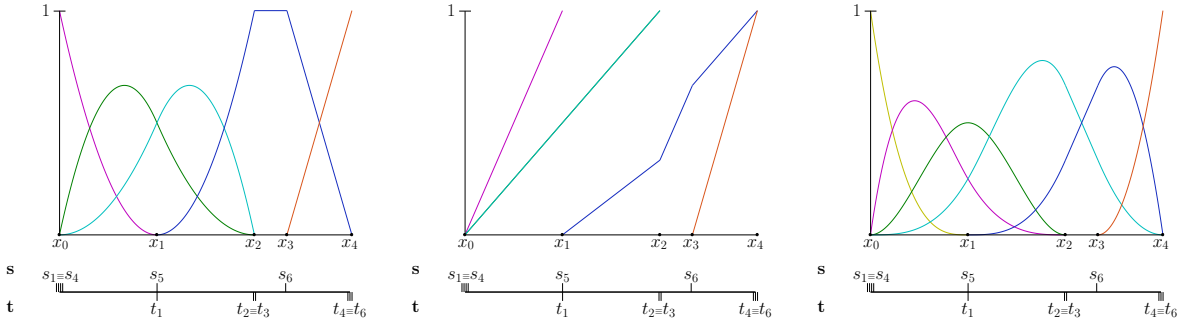
The C^1 MDB-spline basis $\{N_{i,3}\}_{i=1}^{6}$ is shown in Figure 3(i). From the illustration, it can be noted that there are four nonvanishing basis functions in the first two breakpoint intervals, whereas there are two and three such functions in the third and fourth intervals, respectively. This is consistent with the assigned degree vector $\mathbf{d} = (3, 3, 1, 2)$. The function sequences $\{N_{i,n}\}$, built by the recurrence process (3), are illustrated in Figure 3 for increasing steps $n = 0, \ldots, 3$. Note that the support of a function $N_{i,n}$ is $[s_i, t_{i-3+n}]$.



(a) $N_{i,0}$, $i = 4, \ldots, 6$. (b) ϕ_i^0, $i = 4, \ldots, 6$. (c) $N_{i,1}$, $i = 3, \ldots, 6$.



(d) $N_{i,1}$, $i = 3, \ldots, 6$. (e) ϕ_i^1, $i = 3, \ldots, 6$. (f) $N_{i,2}$, $i = 2, \ldots, 6$.



(g) $N_{i,2}$, $i = 2, \ldots, 6$. (h) ϕ_i^2, $i = 2, \ldots, 6$. (i) $N_{i,3}$, $i = 1, \ldots, 6$.

Figure 3: The function sequences $\{N_{i,n}\}$, $n = 0, \ldots, 3$, built by the recurrence relation (3)–(4) for the MD-spline space in Example 1.

Remark 8. *With reference to the extended partitions in Example 1, we observe that some breakpoints do not appear in either the left partition* **s** *or in the right partition* **t**. *This happens when one asks for* C^1 *continuity at a breakpoint separating two pieces of degree one and greater than one, respectively. The situation can thus be interpreted as having a knot of* zero multiplicity *in either one of the extended partitions.*

The standard Cox de-Boor relation naturally yields a recurrence scheme for the coefficients of conventional splines. Similarly, formulas (3)–(4) can be inserted in equation (2) to obtain the following relation, for $x \in [x_j, x_{j+1})$ and $s_\ell \leqslant x_j < \min(s_{\ell+1}, b)$:

$$c_i^k(x) = \phi_i^{m-k}(x) c_i^{k-1}(x) + \left(1 - \phi_i^{m-k}(x)\right) c_{i-1}^{k-1}(x), \quad \text{for} \quad k = 1, \dots, d_j, \quad i = \ell - d_j + k, \dots, \ell, \tag{16}$$

with $c_i^0 := c_i, i = \ell - d_j, \dots, \ell$, and where functions ϕ_i^{m-k} are defined as in (4). In particular, for a given $\bar{x} \in [x_j, x_{j+1})$, $f(\bar{x})$ will be equal to $c_\ell^{d_j}(\bar{x})$. In addition, we observe that formula (16) allows for subdividing an MD-spline at $\bar{x} \in [x_j, x_{j+1}]$ in two pieces with coefficients

$$c_1, \dots, c_{\ell-d_j}, c_{\ell-d_j+1}^1, \dots, c_\ell^{d_j}, \qquad \text{and} \qquad c_\ell^{d_j}, c_\ell^{d_j-1}, \dots, c_\ell^1, c_\ell, \dots, c_K.$$

In the particular case where $\bar{x}$ corresponds to a breakpoint, some of the above coefficients will be the same and the recurrence scheme (16) will require less than d_j steps.

We remark that applying the coefficients recurrence scheme at breakpoints yields the Bézier representation of each piece.

3.2. Recurrence scheme for derivatives computation

The following proposition provides a recurrence relation for the evaluation of derivatives of C^1 MDB-splines.

Proposition 5. *In the same setting of Proposition 4, the derivatives of the MDB-spline functions are given by*

$$D_+ N_{i,n}(x) = \begin{cases} 0, & n = m - d_j, \\ \dfrac{d_j - m + n}{d_j} \left(\dfrac{N_{i,n-1}(x)}{\tilde{t}_{i-m+n-1} - \tilde{s}_i} - \dfrac{N_{i+1,n-1}(x)}{\tilde{t}_{i-m+n} - \tilde{s}_{i+1}} \right), & n = m - d_j + 1, \dots, m, \quad i = \ell - n + m - d_j, \dots, \ell, \end{cases} \tag{17}$$

for $x \in [x_j, x_{j+1})$ *and where* ℓ *is such that* $s_\ell \leqslant x_j < \min(s_{\ell+1}, b)$.

Proof. Up to step $m - 1$ the scheme involves C^0 MDB-splines. Hence, proceeding by analogy with the proof of Lemma 2, we can conclude that, for any $n \leqslant m - 1$, equation (17) coincides with the standard recurrence relation for the derivatives of B-splines. There remains to prove that the last step, corresponding to $n = m$, generates the derivatives of the MDB-splines $\{N_{i,m}\}_{i=1}^K$. To this aim, for any $N_{i,m}$, we differentiate (3) and use formula (17), with $n = m - 1$, to express the derivatives of functions $N_{i,m-1}$ in terms of functions $N_{i,m-2}$. After suitable algebraic manipulations, we again use (3) to express functions $N_{i,m-2}$ in terms of $N_{i,m-1}$. The passages are developed in detail in Appendix A. □

For $x \in [x_j, x_{j+1})$ and $s_\ell \leqslant x_j < \min(s_{\ell+1}, b)$, from equation (17) the following scheme for the coefficients of the derivatives of an MD-spline is obtained:

$$c_i^{(k)} = \begin{cases} c_i, & k = 0, \\ \dfrac{d_j - k + 1}{d_j} \dfrac{c_i^{(k-1)} - c_{i-1}^{(k-1)}}{\tilde{t}_{i-k} - \tilde{s}_i}, & k \geqslant 1 \quad \text{and} \quad \tilde{t}_{i-k} > \tilde{s}_i, \\ 0, & k > 0 \quad \text{and} \quad \tilde{t}_{i-k} = \tilde{s}_i, \end{cases} \tag{18}$$

where $c_i^{(k)}$, $i = \ell - d_j + k, \dots, \ell$, are the coefficients of the derivative of order $k \leqslant d_j$, namely:

$$D_+^{(k)} f(x) = \sum_{i=\ell-d_j+k}^{\ell} c_i^{(k)} N_{i,m-k}(x), \quad x \in [x_j, x_{j+1}), \quad s_\ell \leqslant x_j < \min(s_{\ell+1}, b).$$

Alternatively, for a given order of differentiation k, it is possible to determine all the coefficients $c_i^{(k)}$ from equation (18). In this case, for each coefficient $c_i^{(k)}$ it is necessary to take the corresponding degree d_j, where the correspondence between $c_i^{(k)}$ and d_j is given by the relation $i = k+1, \ldots, \sum_{j=0}^{q}(d_j - k_j)$. Hence the derivative is expressed as

$$D_+^{(k)} f(x) = \sum_{i=k+1}^{K} c_i^{(k)} N_{i,m-k}(x), \quad x \in [a,b].$$

4. Modeling tools

In this section we consider C^1 MD-splines generated by the proposed Cox-de Boor-type recurrence relation and derive efficient formulas for common design tools, including knot-insertion and removal and local degree elevation and reduction. These generalize the classical analogous tools for conventional splines.

4.1. Knot insertion

Knot insertion corresponds to either decreasing continuity at a breakpoint or to inserting a new breakpoint in the middle of one interval. When inserting a breakpoint $\hat{x}$ such that $x_j < \hat{x} < x_{j+1}$ then the interval $[x_j, x_{j+1}]$ is divided in two subintervals with the same associated degree d_j and thus, in the new space, a spline will be continuous of order $d_j - 1$ at $\hat{x}$. Instead, if $\hat{x} \equiv x_j$, a spline in the new space will be continuous of order $k_j - 1$. In both cases, the extended partitions $\mathbf{s}$ and $\mathbf{t}$ should suitably be updated according to the new configuration of breakpoints and associated continuities. In particular knot insertion requires inserting one knot in both $\mathbf{s}$ and $\mathbf{t}$. We will denote the updated breakpoint sequence by $\widehat{\mathcal{X}}$, the $(K+1)$-dimensional space obtained after knot-insertion by $S(\mathcal{P}_{\widehat{\mathbf{d}}}, \widehat{\mathcal{X}}, \widehat{\mathcal{K}})$, the associated extended partitions by $\hat{\mathbf{s}} = \{\hat{s}_i\}_{i=1}^{K+1}$ and $\hat{\mathbf{t}} = \{\hat{t}_i\}_{i=1}^{K+1}$ and the MDB-spline basis by $\{\hat{N}_{i,m}\}_{i=1}^{K+1}$.

The MDB-spline bases $\{N_{i,m}\}_{i=1}^{K}$ and $\{\hat{N}_{i,m}\}_{i=1}^{K+1}$ are related through

$$N_{i,m} = \alpha_i \hat{N}_{i,m} + (1-\alpha_{i+1})\hat{N}_{i+1,m}, \quad i = 1, \ldots, K, \tag{19}$$

and in the case of C^1 MDB-splines the coefficients α_i are given by:

$$\alpha_i = \begin{cases} 1, & i \leqslant \ell - d_j, \\ \dfrac{\tilde{\hat{x}} - \tilde{\hat{s}}_i}{\tilde{\hat{t}}_i - \tilde{\hat{s}}_i}, & \ell - d_j + 1 \leqslant i \leqslant \ell - r + 1, \\ 0, & i \geqslant \ell - r + 2, \end{cases} \tag{20}$$

where $\tilde{\hat{\mathbf{s}}} = \{\tilde{\hat{s}}_j\}$ and $\tilde{\hat{\mathbf{t}}} = \{\tilde{\hat{t}}_j\}$ are associated with $\widetilde{\widehat{\mathcal{X}}}$, $\tilde{\hat{x}}$ is the scaled and translated version of $\hat{x} \in [x_j, x_{j+1})$, r is the multiplicity of $\tilde{\hat{x}}$ in $\tilde{\hat{\mathbf{s}}}$ (or equivalenlty of $\hat{x}$ in $\hat{\mathbf{s}}$) and ℓ is such that $s_\ell \leqslant x_j < \min(s_{\ell+1}, b)$.

Equation (20) is obtained from [1, formula (13)], assuming that the functions f_i in the referenced formula are C^1 MD-splines, recalling that $f_i = \sum_{j=i}^{K} N_{j,m}$ and computing derivatives through (17).

As a consequence of relation (19), a C^1 MD-spline can be expressed as

$$f(x) = \sum_{i=1}^{K} c_i N_{i,m}(x) = \sum_{i=1}^{K+1} \hat{c}_i \hat{N}_{i,m}(x), \quad x \in [a,b],$$

where

$$\hat{c}_i = \begin{cases} c_i, & i \leqslant \ell - d_j, \\ \alpha_i c_i + (1-\alpha_i) c_{i-1}, & \ell - d_j + 1 \leqslant i \leqslant \ell - r + 1, \\ c_{i-1}, & i \geqslant \ell - r + 2, \end{cases}$$

with coefficients α_i, $i = \ell - d_j + 1, \ldots, \ell - r + 1$, given by (20), $\hat{x} \in [x_j, x_{j+1})$ and where ℓ is such that $s_\ell \leqslant x_j < \min(s_{\ell+1}, b)$.

4.2. Degree elevation

As a peculiarity of MD-splines, degree elevation can be performed locally, namely one can raise the degree of one (or more) section space(s) only, maintaining the other degrees unchanged. Apparently, this feature is a major and important difference with respect to conventional splines, where elevating the degree is a global operation.

Let $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ be an MD-spline space having degree sequence $\mathbf{d} = (d_0, \dots, d_q)$ and dimension K. It suffices to discuss the case in which one wants to elevate the degree on a single interval, say $[x_j, x_{j+1}]$, from d_j to $d_j + 1$, generating a $(K+1)$-dimensional space $S(\mathcal{P}_{\widehat{\mathbf{d}}}, \mathcal{X}, \mathcal{K})$. In this setting, each MDB-spline basis function of $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ nonvanishing in $[x_j, x_{j+1}]$ will be expressed as a combination of two "new" ones. At the same time, all the MDB-spline basis functions in $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$ which are zero on $[x_j, x_{j+1}]$ remain unchanged in the degree elevated space. Thus degree elevation only requires to recompute a limited number of basis functions, as already observed in [1].

A C^1 MD-spline can be expressed as

$$f(x) = \sum_{i=1}^{K} c_i \, N_{i,m}(x) = \sum_{i=1}^{K+1} \hat{c}_i \, \hat{N}_{i,\hat{m}}(x), \quad x \in [a, b],$$

where

$$\hat{c}_i = \begin{cases} c_i, & i \leqslant \ell - d_j, \\ \alpha_i \, c_i + (1 - \alpha_i) \, c_{i-1}, & \ell - d_j + 1 \leqslant i \leqslant \ell, \\ c_{i-1}, & i \geqslant \ell + 1, \end{cases}$$

with α_i given by:

$$\alpha_i = \frac{\ell + 1 - i}{d_j} \, \frac{\tilde{\hat{t}}_{i-1} - \tilde{\hat{s}}_i}{\tilde{t}_{i-1} - \tilde{s}_i}, \quad i = \ell - d_j + 1, \dots, \ell, \tag{21}$$

where $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{t}}$ are the scaled and translated extended partitions relative to $S(\mathcal{P}_{\mathbf{d}}, \mathcal{X}, \mathcal{K})$, $\tilde{\hat{\mathbf{s}}}$ and $\tilde{\hat{\mathbf{t}}}$ are the scaled and translated extended partitions relative to $S(\mathcal{P}_{\widehat{\mathbf{d}}}, \mathcal{X}, \mathcal{K})$, and ℓ is such that $s_\ell \leqslant x_j < \min(s_{\ell+1}, b)$. A proof of (21) is given in Appendix B.

Remark 9. *We shall observe that, when elevating the degree of one piece, one may generate adjacent breakpoint intervals with associated different degrees. Therefore, prior to performing degree elevation, it will be necessary to apply multiple knot insertions to reduce the continuity to be at most C^1 where necessary.*

Remark 10. *One can also use the proposed local degree elevation method to perform global degree elevation in conventional spline spaces of any continuity. In particular, to do so, first knot insertion needs to be performed at each breakpoint, so as to decrease the continuity to be at most C^1. Successively local degree elevation shall be applied on each breakpoint interval. Finally knot removal shall be performed to restore the original continuity.*

4.3. Knot removal and degree reduction by dual basis

Knot removal and degree reduction are the inverse operations of knot insertion and degree elevation and they consist in representing an MD-spline function in a space with lower dimension. Such representation can be exact or approximated, depending on whether the lower dimensional space contains the initial function or not. As a peculiarity of MD-splines, degree reduction can be performed locally. One advantage of having the MDB-spline basis at our disposal is that we can use its dual basis in order to approximate the initial function, when necessary. The dual basis can efficiently be computed using the algorithm in [15], which naturally lends itself to be generalized to MD-splines and can be applied to perform a constrained or unconstrained least squares approximation of the initial function. We remark that, despite least squares approximation can either be local or global, according to our numerical experiments a global method results in significantly higher quality.

5. Periodic MD-splines

For ease of presentation, so far we have dealt with spline spaces on clamped partitions. Nevertheless the results in this paper are readily generalizable to the case of periodic partitions.

To generate a periodic MD-spline function we will additionally need to specify a continuity $k_0 = k_{q+1}$ at the endpoints $a \equiv x_0$ and $b \equiv x_{q+1}$, in such a way that $-1 \leqslant k_0 \leqslant \min(d_0, d_q)$. Note that $k_0 = k_{q+1} = -1$ corresponds to a non periodic setting. Generalizing Proposition 1, we can see that the spline space has dimension:

$$K = \sum_{i=0}^{q} d_i - k_i = \sum_{i=0}^{q} d_i - k_{i+1}.$$

To generate a periodic spline, we will replicate a suitable number of breakpoints and associated continuities outside the interval $[a, b]$. More precisely, the amount of information to be replicated is given by as many breakpoint intervals and continuities as necessary in order to generate a left extended partition with $d_0 + 1$ knots smaller than or equal to a; Similarly we will need to generate an extended right partition having $d_q + 1$ knots t_i greater than or equal to b. A periodic MD-spline is defined by K coefficients and according to the discussed wrapping strategy the first $k_0 + 1$ coefficients must be replicated at the end of the sequence so as to generate $K + k_0 + 1$ coefficients overall.

The following two examples illustrate the procedure for obtaining a periodic C^1 MD-spline. We remark, however, that the periodic construction applies more generally to the wider class of MD-splines in [1].

Example 2. Let us consider an interval $[0, 4]$, a partition $\mathcal{X} = \{2\}$, a degree vector $\mathbf{d} = (3, 3)$ and a continuity vector $\mathcal{K} = (2)$. We additionally require C^1 continuity at the closure, namely $k_0 = k_2 = 1$. The resulting periodic MD-spline space has dimension $K = 3$ and in order to construct the corresponding MDB-splines, we will need to augment the

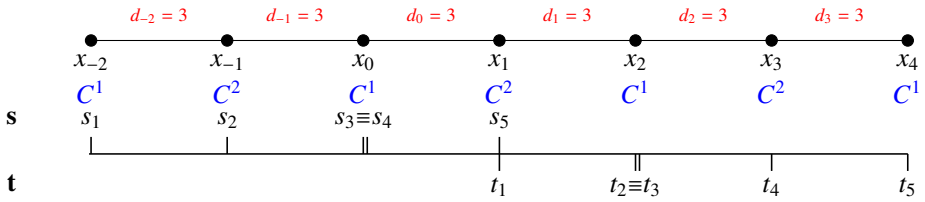


Figure 4: Setting for the spline space in Example 2.

sequence of breakpoints and the degree and continuity vectors as shown in Figure 4. In this way we obtain

$$\mathcal{X} = \{-4, -2, 0, 2, 4, 6, 8\}, \qquad \mathbf{d} = (3, 3, 3, 3, 3, 3), \qquad \mathcal{K} = (2, 2, 1, 2, 1, 2, 2),$$

and the corresponding left and right extended partitions become

$$\mathbf{s} = \{-4, -2, 0, 0, 2\} \quad \text{and} \quad \mathbf{t} = \{2, 4, 4, 6, 8\}.$$

The peculiarity of this example is that the first entry in the augmented continuity vector $\mathcal{K}$ is 2, despite replicating periodically the spline space configuration would give C^1 continuity at x_{-2} (see Figure 4). This is due to the fact that all MDB-spline basis functions nonvanishing in $[0, 2]$ can be determined from a left extended partition $\mathbf{s}$ having just one knot at x_{-2}. The same reasoning explains the continuity at the last entry of $\mathcal{K}$ and the multiplicity of the last knot in $\mathbf{t}$.

The C^1 MDB-spline basis and a curve from the considered space are illustrated in Figure 6(a)–(b). Note that, since we require C^1 continuity at the closure, the first two control points have been replicated at the end.

Example 3. Let us consider an interval $[0, 9]$, a partition $\mathcal{X} = \{3, 6, 7\}$, a degree vector $\mathbf{d} = (3, 3, 1, 2)$ and a continuity vector $\mathcal{K} = (2, 1, 1)$. Furthermore, let us require C^1 continuity at the closure, namely $k_0 = k_4 = 1$. The corresponding spline space has dimension $K = 4$ and we will need to augment the sequence of breakpoints and the vectors of degrees and continuities as shown in Figure 5. This yields

$$\mathcal{X} = \{-6, -3, -2, 0, 3, 6, 7, 9, 12, 15\}, \qquad \mathbf{d} = (3, 1, 2, 3, 3, 1, 2, 3, 3), \qquad \mathcal{K} = (2, 1, 1, 1, 2, 1, 1, 1, 2, 2),$$

and, accordingly, the left and right extended partitions become

$$\mathbf{s} = \{-6, -2, 0, 0, 3, 7\} \quad \text{and} \quad \mathbf{t} = \{3, 6, 6, 9, 12, 15\}.$$

In particular, note that the last entry in $\mathcal{K}$ is 2 (instead of 1) and thus there is only the one knot t_6 in correspondence to the breakpoint x_6. This is explained by the fact that only three knots greater than or equal to $x_4 \equiv b$ are required in order to generate a periodic closure.

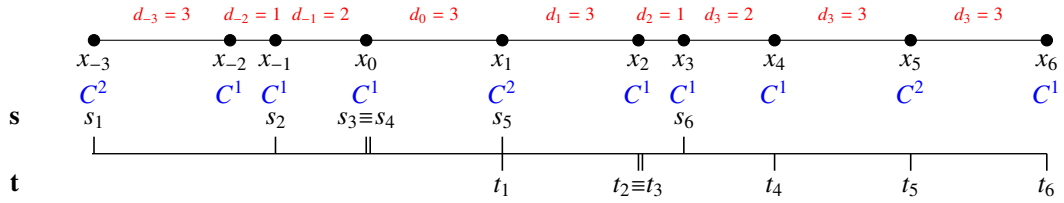


Figure 5: Setting for the spline space in Example 3.

The C^1 MDB-spline basis and a curve from this space are illustrated in Figure 6(c)–(d). Note that, since we require C^1 continuity at the closure, the first two coefficients have been replicated at the end.

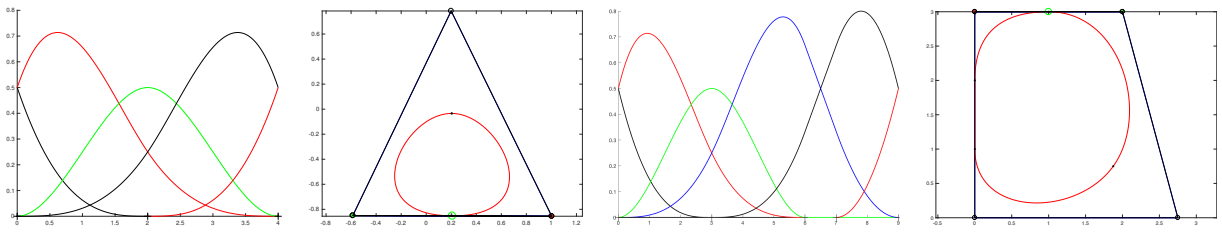
Remark 11. *Note that the discussed Cox de-Boor-type recurrence relation does not depend on the location of the first knot in* **s***, nor of the last knot in* **t***. As a consequence one could set these knots to be coincident with the next neighboring ones. For instance, in Example 2, setting $s_1 \equiv s_2$ and $t_5 \equiv t_4$ the result would be unchanged, but we would avoid replicating two breakpoint intervals. The same holds for Example 3.*

6. Examples and applications

In this section we address two specific applications, aimed at demonstrating the potential of C^1 MD-splines. One is concerned with modeling and storing images in SVG format, whereas the other with designing non-uniform rational C^1 MD-splines reproducing circles and quadrics.

6.1. C^1 MD-spline curves and SVG image format

As part of this research we designed a simple yet powerful Web-App to experience and appreciate the benefits of modeling with C^1 MD-splines. The Web-App, available at http://www.dm.unibo.it/~casciola/MD_spline, allows for generating a single 2D C^1 MD-spline curve and for editing it by exploiting the modeling tools proposed in this paper. In particular, the user starts by choosing a degree and an order of continuity and interactively places the control points. Hence a conventional spline curve is generated and can then be edited by all the tools described in the previous sections, namely local degree elevation and reduction, decrease/increase of continuity at a breakpoint (i.e. knot insertion/removal), subdivision of one interval by insertion of a new breakpoint and by simply moving the control points.



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

Figure 6: Periodic curves obtained from the C^1 MD-spline spaces described in Examples 2 and 3 and related periodic MDB-spline bases. The endpoints of the different curve pieces are marked by black dots and the green circle indicates the first endpoint of the curve, that is oriented clockwise; the first and last control points are marked by a red and green asterisk, respectively.

The potential of MD-splines goes far beyond the well-known savings in terms of more compact storage. To make this apparent it is necessary to say something more about existing technologies for handling vector graphics. Indeed, nowadays there exist several Web-Apps for online drawing of vector images and nearly all of them allow to load and save vector graphics in SVG format.

From Wikipedia: *Scalable Vector Graphics (SVG) is an XML-based vector image format for two-dimensional graphics with support for interactivity and animation. The SVG specification is an open standard developed by the World Wide Web Consortium (W3C) since 1999. SVG images and their behaviors are defined in XML text files. This means that they can be searched, indexed, scripted, and compressed. As XML files, SVG images can be created and edited with any text editor, as well as with drawing software. All major modern web browsers including Mozilla Firefox, Internet Explorer, Google Chrome, Opera, Safari and Microsoft Edge provide SVG rendering support* .

One of the most powerful tags in SVG is the `<path>`. A path is typically composed of a number of distinct curves, each of which can be formed by several polynomial pieces of degree 1, 2 or 3; the continuity between pieces of the same degree can be C^0 or C^1. In fact, the SVG format provides the commands `L, Q, C` (Linear, Quadratic, Cubic) that allow for defining curve pieces joined with C^0 continuity, but also the `S` and `T` commands that define respectively a cubic or quadratic piece connected with C^1 continuity with the previous one, if both have the same degree. Table 1 presents some examples of SVG syntax to define piecewise curves with different continuities at the joins.

`M x1 y1 L x2 y2 L x3 y3`	two line pieces with a C^0 join
`M x1 y1 Q x2 y2 x3 y3 Q x4 y4 x5 y5`	two quadratic pieces with a C^0 join
`M x1 y1 Q x2 y2 x3 y3 T x5 y5`	two quadratic pieces with a C^1 join
`M x1 y1 C x2 y2 x3 y3 x4 y4 C x5 y5 x6 y6 x7 y7`	two cubic pieces with a C^0 join
`M x1 y1 C x2 y2 x3 y3 x4 y4 S x6 y6 x7 y7`	two cubic pieces with a C^1 join

Table 1: Continuity between polynomial pieces stored in SVG format.

Of course, one can modify a vector image in SVG format by editing its source file. This is possible for small files and simple drawings, but it turns out impractical in all other cases, where it is by far preferable to use a drawing software that allows for loading and saving in SVG format, such as for example the well known and professional Inkscape.

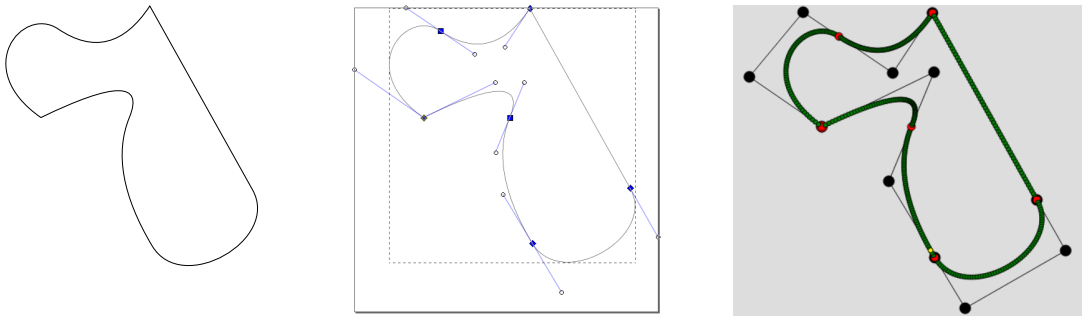
Any software that allows for editing an SVG file represents the curves of a path in its own internal format. For example, Inkscape represents a linear piece (command `L`) with a line segment and a quadratic or cubic piece (`Q` or `C` commands) with a cubic polynomial. When two consecutive quadratic or cubic pieces join with C^1 continuity (`T` or `S` commands), then Inkscape stores them as two cubic pieces with a C^0 join and identifies with a blue square the breakpoint in between, allowing the user to modify the control points while keeping the C^1 constraint. However, as soon as one piece of a curve path is modified, Inkscape saves the path in its internal format, namely as one sequence of linear and cubic pieces joined with C^0 continuity.

On the contrary, a system based on C^1 MD-splines has the potential of maintaining the degrees and continuities of the different pieces during the editing phase and to save the path information in the most compact possible way. The following examples aim at illustrating these advantages. Table 2 shows the syntax of a simple SVG file containing a path with only one closed curve, formed by a sequence of pieces of degrees 3,2,1,3,2,2 (`CQLCQT`), with orders of continuity $0, 0, 0, 0, 1$ and 0 at the closure; the corresponding curve is shown in Figure 7(a).

Figure 7(b) shows the same curve in Inkscape, along with the control points of the internal representation (circles, squares and diamonds, the latter two are also the images of the breakpoints). Note that blue squares correspond to breakpoints where C^1 continuity is required, whereas blue diamonds identify breakpoints where only C^0 continuity is required. Figure 7(c) shows the same curve represented as a C^1 MD-spline, along with its control points (black dots) and breakpoint images (red dots). It can be noted that Inkscape converts the two quadratic pieces with a C^1 join (`QT`) into two cubic pieces with a C^0 join, thus using 7 control points. Oppositely the C^1 MD-spline representation enables us to maintain unaltered the different degrees of the pieces and continuities, in such a way that the same two quadratic pieces joined C^1 can be represented with 4 control points only.

```
<?xml version="1.0" encoding="UTF-8"?>
<svg version="1.1" xmlns="http://www.w3.org/2000/svg" viewBox="0,0,1000,1000">
<g transform="translate(98.198506,-0.000000)">
<path d="M 129.797768 361.737805 C -98.198506 203.887510 71.210508 0.000000
184.877504 76.664516 Q 355.377997 191.661290 479.965266 2.894955 L 809.990080
593.465674 C 901.801494 754.755997 584.183628 935.897436 487.905712 775.103391
Q 343.482830 533.912329 414.208442 361.455749 T 129.797768 361.737805 "
stroke="black" fill="none"/>
</g>
</svg>
```

Table 2: A simple SVG file defining one path.



(a) (b) (c)

Figure 7: (a) Curve in SVG format; (b) Inkscape screenshot; (c) Web-App screenshot.

(a) (b) (c)

Figure 8: (a) Glyph ca; (b) Facebook logo; (c) Apple logo.

```
M 129.797768 361.737805 C -98.198506 203.887510 71.210508 0.000000
184.877504 76.664516 Q 355.377997 191.661290 479.965266 2.894955 L 809.990080
593.465674 C 901.801494 754.755997 584.183628 935.897436 487.905712 775.103391
Q 343.482830 533.912329 414.208442 361.455749 T 129.797768 361.737805
```

Table 3: Example of a file containing a path that can be uploaded through the Web-App.

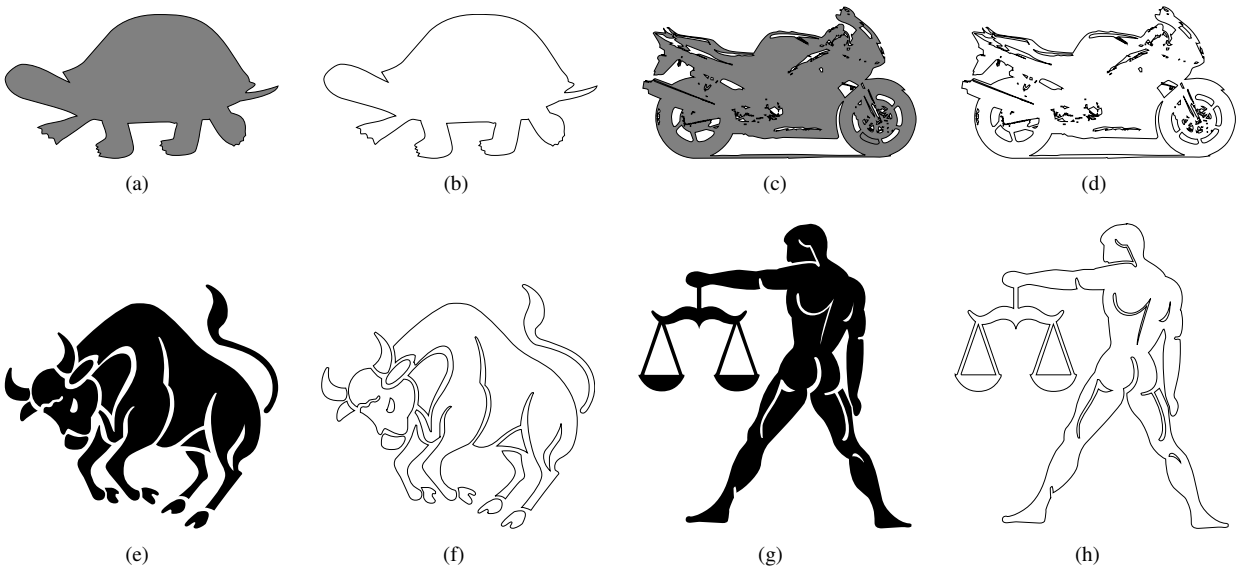


Figure 9: Some TrueType characters and their outlines.

Note that although the SVG path defines only one C^1 join (`QT`), our SVG parser detects that there is another C^1 join between the first cubic and quadratic pieces (`CQ`) and maintains that continuity.

The Web-App made as a demo for 2D C^1 MD-spline curves allows for uploading files containing a single curve path in SVG format. Figure 7(c) is a screenshot of the Web-App canvas and the corresponding uploaded file is provided in Table 3.

Name	Curve	Segments	LQC (Linear/Quadratic/Cubic)	CPs	MD CPs	Inkscape CPs
ca	c	11	LQQQQLQQQQQ	21	17	30
ca	a out	14	LQQQQQQLCCQQLQ	28	25	37
ca	a into	5	QQQLQ	10	10	14
facebook	f	17	LLLLLLCLLLCLLLLLL	22	22	22
facebook	□	8	QLQLQLQL	13	13	17
facebook	⬭	8	QLQLQLQL	13	13	17
apple	apple	10	CCCCCCCCCC	31	25	31
apple	leaf	2	CC	7	7	7

Table 4: Defining parameters for the SVG images in Figure 8.

Figure 8 presents examples of simple SVG images, each of which is formed by various piecewise curves having different degrees. Table 4 lists the related information: in particular columns CPs, MD CPs and Inkscape CPs report respectively the number of control points stored in the initial SVG file, used for the internal representation as C^1 MD-splines (note that the implemented algorithm determines the minimum configuration) and for the internal representation in Inkscape. Although only examples of small size have been considered, the savings in terms of control points is noticeable.

As mentioned, as soon as just one piece is edited, Inkscape requires to store a file with as many control points as those of its internal representation, while a software based on C^1 MD-splines will save it with the control points indicated in the CPs column, to comply with the SVG format.

Figure 9 shows characters (or glyphs) extracted from TrueType files and exported in SVG format; the outlines in the True-Type files are made of straight lines and quadratic Bézier curves. This means that once imported into Inkscape they will be converted to linear and cubic pieces, and, after editing, their saving in SVG format will reflect the internal representation in Inkscape with a significant increase in file size; this increase is not only due to the use

of cubic sections to represent quadratic pieces, but also to the floating point format used by Inkscape for saving (fixed point representation with 5 decimal places). In Table 5 some useful information for analysis and comparison are reported.

Name	Curves	Segments	CPs	Bytes	Inkscape Bytes
Turtle	1	116	185	2411	4586
Libra	7	259	458	4112	8510
Taurus	18	343	624	5294	11855
Motorbike	162	1179	1951	14045	28183

Table 5: Defining parameters for the SVG images in Figure 9.

In all these examples, the files saved by Inkscape require more than double storage occupancy in bytes with respect to the original SVG files.

To conclude this section, we observe that the SVG standard handles only C^0 joins between curve pieces of different degrees, whereas a C^1 join is only possible when the two pieces have the same degree. The discussed benefits of C^1 MD-splines suggest that the SVG standard could conveniently be extended to handle C^1 continuity between pieces of different degrees and C^2 continuity between cubic segments.

6.2. Multi degree NURBS and reproduction of circles and quadrics

Non-uniform rational multi-degree curves and surfaces can be constructed in the same way as NURBS from conventional splines. The rational formulation enables us to exactly (or approximately) represent conic sections or quadric surface patches with the additional benefits of the multi-degree framework, namely using less control points compared to the conventional representation. As a demonstration, we will discuss how to construct surfaces including exact spherical patches by means of rational MD-spline segments joined with C^0 continuity. Moreover we will discuss how to approximate the same surfaces using less control points by means of rational MD-splines joined with C^1 continuity.

Let us consider the problem of designing a square frame with spherical corners. We will see how this can be done using a trajectory curve and three different profile curves modeled by means of MD-NURBS. The trajectory curve represents a closed frame (Figure 10(a)), composed of four rational pieces of degree 2/2, each of which is a circular arc of 90^o joined to two linear segments with C^0 continuity. The corresponding MD-NURBS space $S_{1,u}$ is determined by the following parameters:

$$S_{1,u}\colon\ [a,b]=[0,8],\ \mathcal{X}=\{1,2,3,4,5,6,7\},\ \mathbf{d}=(2,1,2,1,2,1,2,1),\ \mathcal{K}=(0,0,0,0,0,0,0,0),\ k_0=k_8=0,$$
$$\mathbf{w}=(1,\ \sqrt{2}/2,1,1,\ \sqrt{2}/2,1,1,\ \sqrt{2}/2,1,1,\ \sqrt{2}/2,1),$$

where $\mathbf{w}$ are the weights of the rational representation.

The three profile curves (Figure 10(b)) are each composed of 3 sections and defined by the same 5 control points (indicated by small circles and blue control polygon). However, they are different in shape since they have different associated breakpoint sequences, degree vectors and continuity vectors. In particular, the corresponding MD-NURBS spaces $S_{1,v}$, $S_{2,v}$, $S_{3,v}$ are described by parameters $[a,b]=[0,3]$, $k_0=k_3=-1$, $\mathbf{w}=(1,\ \sqrt{2}/2,1,1,1)$ and

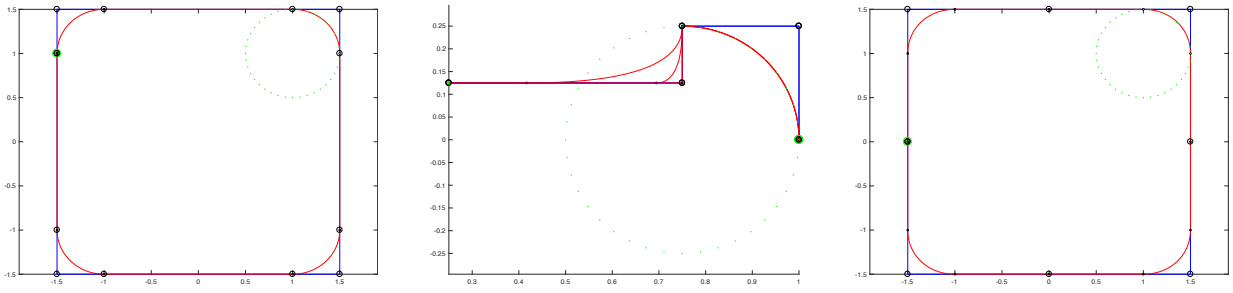
$$S_{1,v}\colon\ \mathcal{X}=\{1,2\},\ \mathbf{d}=(2,1,1);$$

$$S_{2,v}\colon\ \mathcal{X}=\{1,1+2h\},\ \text{with } h=0.2,\ \mathbf{d}=(2,2,1);$$

$$S_{3,v}\colon\ \mathcal{X}=\{1,1+2h\},\ \text{with } h=0.8,\ \mathbf{d}=(2,2,1).$$

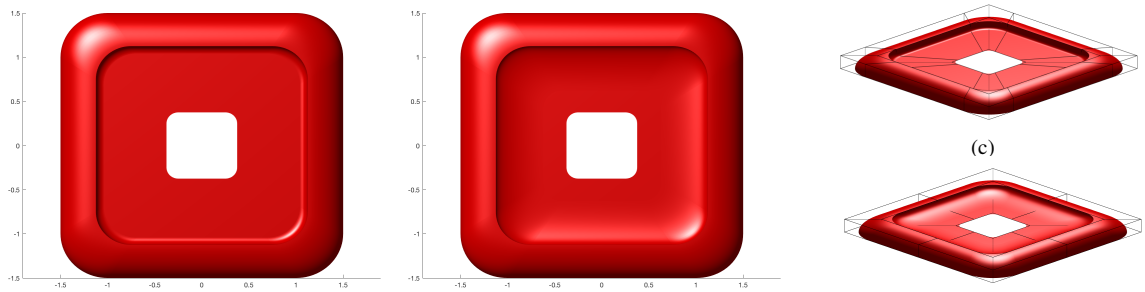
Note that the first piece of all three profile curves is a circular segment of 90^o.

The trajectory and profile curves have been composed to generate swinging surfaces of MD-NURBS type, representing three 3D frames with different profile sections. Figure 12 shows a section of each surface, highlighting the different profiles. Figures 11(a) and 11(c) show the entire surface obtained from the profile curve in $S_{2,v}$, respectively from the top and from a generic point of view, with the related grid of 12×5 control points. The smooth corner



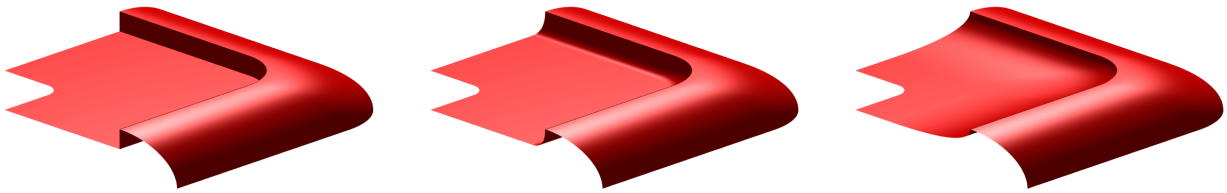
(a) (b) (c)

Figure 10: Construction of the MD-NURBS surfaces discussed in Section 6.2. (a) Trajectory curve in space $S_{1,u}$; (b) Profile curves in spaces $S_{1,v}$, $S_{2,v}$, $S_{3,v}$; (c) Trajectory curve in space $S_{2,u}$.



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

Figure 11: Construction of the MD-NURBS surfaces discussed in Section 6.2. (a) Swinging surface obtained from a trajectory curve in space $S_{1,u}$ and a profile curve in space $S_{2,v}$; (b) Swinging surface obtained from a trajectory curve in space $S_{2,u}$ and a profile curve in space $S_{3,v}$; (c)–(d) Perspective views of the surfaces in (a) and (b) respectively.



(a) (b) (c)

Figure 12: Parts of the MD-NURBS swinging surfaces obtained from a trajectory curve in space $S_{1,u}$ and profile curves (a) in $S_{1,v}$, (b) in $S_{2,v}$ and (c) in $S_{3,v}$.

patches are exact ellipsoidal patches. Note that a conventional NURBS representation (i.e. where all pieces have same degree) of the same surface would result in a 17×5 grid of control points and weights.

To conclude, we illustrate how to model a similar surface, where the circular sections are just approximated. This will result in higher smoothness and less control points. To obtain a close approximation of the trajectory curve (see Figure 10(c)), we rely on the MD-NURBS space

$$S_{2,u}\colon\ [a,b] = [0,8],\ \mathcal{X} = \{1, 1.25, 2.25, 3.25, 3.5, 4.5, 5.5, 5.75, 6.75, 7.75\},\ \mathbf{d} = (1,2,1,1,2,1,1,2,1,1,2,1),$$
$$\mathcal{K} = (1,1,0,1,1,0,1,1,0,1,1),\ k_0 = k_{12} = 0,\ \mathbf{w} = (1, 0.25, 1, 0.25, 1, 0.25, 1, 0.25).$$

The resulting curve has more pieces than the previous one, and, more precisely, is composed of 4 rational pieces of degree 2/2 and 8 pieces of degree 1/1, joined with C^1 continuity. The corresponding control polygon has 8 control points only, so that the swinging surface obtained by composing the approximated trajectory curve with one of the profile curves from $S_{1,v}$, $S_{2,v}$ or $S_{3,v}$ will be defined by a grid of 8×5 control points and weights.

The rational quadratic approximation of the circular arcs can be modified by altering the weights associated with the control points of the quadratic rational segments. In the example, the relative error between the rational quadratic and the circle of radius 0.5 (in green in the figure) is 1/100 (that is the distance between the curve points and the circle is at most 1/100 of the radius). Figures 11(b) and 11(d) show the swinging surface generated using profile curve $S_{3,v}$, respectively from the top and from an arbitrary point of view, together with the related grid of 8×5 control points.

7. Conclusion

We have proposed a Cox-de Boor-type recurrence relation for evaluating multi-degree splines that are at most C^1 continuous between pieces of different degrees, along with a similar formula for the computation of derivatives and algorithms for knot-insertion/removal and local degree elevation/reduction. The impact of efficient algorithms for MD-splines, even though with limited continuity, has been illustrated by several numerical examples concerning applications in geometric modeling as well as vector graphics. These examples emphasize that one can efficiently evaluate C^1 MD-splines, perform modeling operations maintaining the continuity, save the necessary data and load them again to continue editing. This is not possible within any available modeling software nor within the current SVG standard.

Cox-de Boor-type recurrence relations are probably the simplest and most efficient technique for spline evaluation. However, as proved in [1], these relations are available for C^1 MD-splines only, whereas no similar method may exist if one allows more general continuities between pieces of different degrees. Our future research extends in the direction of studying alternative algorithms for more general multi-degree spline spaces and applications of the multi-degree framework in surface modeling.

Acknowledgements

The authors gratefully acknowledge support from the Italian GNCS-INdAM.

Appendix A. Proof of the recurrence scheme for derivatives computation

Without loss of generality, we prove relation (17) for a C^1 MDB-spline $N_{i,m}(x)$, with $x \in [x_j, x_{j+1})$, $j = ps_i, \ldots, pt_i - 1$. By differentiating (3) and substituting $D_+N_{i,m-1}$ and $D_+N_{i+1,m-1}$ with (17), which is known to be correct up to step $m-1$, we get

$$\begin{aligned} D_+N_{i,m}(x) = {} & \frac{1/d_j}{\tilde{t}_{i-1} - \tilde{s}_i} N_{i,m-1}(x) + \frac{\tilde{x} - \tilde{s}_i}{\tilde{t}_{i-1} - \tilde{s}_i} D_+N_{i,m-1}(x) - \frac{1/d_j}{\tilde{t}_i - \tilde{s}_{i+1}} N_{i+1,m-1}(x) + \frac{\tilde{t}_i - \tilde{x}}{\tilde{t}_i - \tilde{s}_{i+1}} D_+N_{i+1,m-1}(x) \\ = {} & \frac{1/d_j}{\tilde{t}_{i-1} - \tilde{s}_i} N_{i,m-1}(x) + \frac{\tilde{x} - \tilde{s}_i}{\tilde{t}_{i-1} - \tilde{s}_i} \frac{d_j - 1}{d_j} \left(\frac{N_{i,m-2}(x)}{\tilde{t}_{i-2} - \tilde{s}_i} - \frac{N_{i+1,m-2}(x)}{\tilde{t}_{i-1} - \tilde{s}_{i+1}} \right) \\ & - \frac{1/d_j}{\tilde{t}_i - \tilde{s}_{i+1}} N_{i+1,m-1}(x) + \frac{\tilde{t}_i - \tilde{x}}{\tilde{t}_i - \tilde{s}_{i+1}} \frac{d_j - 1}{d_j} \left(\frac{N_{i+1,m-2}(x)}{\tilde{t}_{i-1} - \tilde{s}_{i+1}} - \frac{N_{i+2,m-2}(x)}{\tilde{t}_i - \tilde{s}_{i+2}} \right). \end{aligned}$$

By adding and subtracting to the right-hand side of the last equality the quantities

$$\frac{\tilde{t}_{i-1}-\tilde{x}}{\tilde{t}_{i-1}-\tilde{s}_{i+1}}N_{i+1,m-2}(x) \qquad \text{and} \qquad \frac{\tilde{x}-\tilde{s}_{i+1}}{\tilde{t}_{i-1}-\tilde{s}_{i+1}}N_{i+1,m-2}(x)$$

we obtain

$$\begin{aligned} D_+N_{i,m}(x) = \quad & \frac{1/d_j}{\tilde{t}_{i-1}-\tilde{s}_i}N_{i,m-1}(x) + \frac{(d_j-1)/d_j}{\tilde{t}_{i-1}-\tilde{s}_i}\left(\frac{\tilde{x}-\tilde{s}_i}{\tilde{t}_{i-2}-\tilde{s}_i}N_{i,m-2}(x) + \frac{\tilde{t}_{i-1}-\tilde{x}}{\tilde{t}_{i-1}-\tilde{s}_{i+1}}N_{i+1,m-2}(x) - \frac{\tilde{t}_{i-1}-\tilde{x}}{\tilde{t}_{i-1}-\tilde{s}_{i+1}}N_{i+1,m-2}(x) - \frac{\tilde{x}-\tilde{s}_i}{\tilde{t}_{i-1}-\tilde{s}_{i+1}}N_{i+1,m-2}(x)\right) \\ & - \frac{1/d_j}{\tilde{t}_i-\tilde{s}_{i+1}}N_{i+1,m-1}(x) + \frac{(d_j-1)/d_j}{\tilde{t}_i-\tilde{s}_{i+1}}\left(\frac{\tilde{t}_i-\tilde{x}}{\tilde{t}_{i-1}-\tilde{s}_{i+1}}N_{i+1,m-2}(x) + \frac{\tilde{x}-\tilde{s}_{i+1}}{\tilde{t}_{i-1}-\tilde{s}_{i+1}}N_{i+1,m-2}(x) - \frac{\tilde{x}-\tilde{s}_{i+1}}{\tilde{t}_{i-1}-\tilde{s}_{i+1}}N_{i+1,m-2}(x) - \frac{\tilde{t}_i-\tilde{x}}{\tilde{t}_i-\tilde{s}_{i+2}}N_{i+2,m-2}(x)\right). \end{aligned}$$

Using (3) and adding up we obtain

$$\begin{aligned} D_+N_{i,m}(x) = \quad & \frac{1/d_j}{\tilde{t}_{i-1}-\tilde{s}_i}N_{i,m-1}(x) + \frac{(d_j-1)/d_j}{\tilde{t}_{i-1}-\tilde{s}_i}N_{i,m-1}(x) - \frac{(d_j-1)/d_j}{\tilde{t}_{i-1}-\tilde{s}_i}\frac{\tilde{t}_{i-1}-\tilde{s}_i}{\tilde{t}_{i-1}-\tilde{s}_{i+1}}N_{i+1,m-2}(x) \\ & - \frac{1/d_j}{\tilde{t}_i-\tilde{s}_{i+1}}N_{i+1,m-1}(x) - \frac{(d_j-1)/d_j}{\tilde{t}_i-\tilde{s}_{i+1}}N_{i+1,m-1}(x) + \frac{(d_j-1)/d_j}{\tilde{t}_i-\tilde{s}_{i+1}}\frac{\tilde{t}_i-\tilde{s}_{i+1}}{\tilde{t}_{i-1}-\tilde{s}_{i+1}}N_{i+1,m-2}(x). \end{aligned}$$

Simplifying the above expression and using again (3) we finally get

$$D_+N_{i,m}(x) = \quad \frac{N_{i,m-1}(x)}{\tilde{t}_{i-1}-\tilde{s}_i} - \frac{N_{i+1,m-1}(x)}{\tilde{t}_i-\tilde{s}_{i+1}}.$$

This completes the proof.

Appendix B. Proof of the degree-elevation formula for C^1 MD-splines

In the following we will show the correctness of equation (21). To perform degree-elevation, we consider the derivative of a C^1 MD-spline, which is a polynomial of degree $d_j - 1$ on $[x_j, x_{j+1}]$. By taking the expansion in the Bernstein basis of degree d_{j-1} relative to $[x_j, x_{j+1}]$, degree-elevating and computing the primitive we get the expression of the degree-elevated MD-spline in the considered interval.

For completeness, we recall the well-known relation between the coefficients of a polynomial f and its degree-elevated version in the Bernstein basis, namely

$$f(x) = \sum_{i=0}^{n} b_i B_{i,n}(x) = \sum_{i=0}^{n+1} \hat{b}_i B_{i,n+1}(x),$$

with

$$\hat{b}_i = \alpha_i b_{i-1} + (1-\alpha_i)b_i \quad i = 0, \dots, n+1 \quad \text{and} \quad \alpha_i = \frac{i}{n+1}. \tag{B.1}$$

To perform degree elevation in $[x_j, x_{j+1}]$, with $s_\ell \leqslant x_j < \min(s_{\ell+1}, b)$, we shall first reduce to C^1 the continuity at x_j and x_{j+1} and then consider the coefficients c_i, for $i = \ell - d_j, \dots, \ell$, which define the MD-spline function in the considered interval. Hence we proceed as follows:

1. using (18) determine $c_i^{(1)}$, for $i = \ell - d_j + 1, \dots, \ell$, that are the coefficients of a polynomial of degree $d_j - 1$ in the Bernstein basis;
2. using (B.1) elevate the degree from $d_j - 1$ to d_j, obtaining coefficients $\hat{c}_i^{(1)}$, for $i = \ell - d_j + 1, \dots, \ell + 1$;
3. using the inverse of (18) and coefficients $\hat{c}_i^{(1)}$, compute the coefficients $\hat{c}_i$, for $i = \ell - d_j + 1, \dots, \ell$.

Equation (21) is the result of the above three steps.

References

[1] Beccari C, Casciola G, Morigi S. On multi-degree splines. Computer Aided Geometric Design 2017;58:8–23. doi:10.1016/j.cagd.2017.10.003.

[2] Sederberg TW, Zheng J, Song X. Knot intervals and multi-degree splines. Comput Aided Geom Design 2003;20(7):455–68. doi:10.1016/S0167-8396(03)00096-7.

[3] Buchwald B, Mühlbach G. Construction of B-splines for generalized spline spaces generated from local ECT-systems. J Comput Appl Math 2003;159(2):249–67. doi:10.1016/S0377-0427(03)00533-8.

[4] Nürnberger G, Schumaker LL, Sommer M, Strauss H. Generalized chebyshevian splines. SIAM J Math Anal 1984;15(4):790–804. doi:10.1137/0515061.

[5] Wang G, Deng C. On the degree elevation of B-spline curves and corner cutting. Comput Aided Geom Design 2007;24(2):90–8. doi:10.1016/j.cagd.2006.10.004.

[6] Shen W, Wang G. Changeable degree spline basis functions. J Comput Appl Math 2010;234(8):2516–29. doi:10.1016/j.cam.2010.03.015.

[7] Shen W, Wang G. A basis of multi-degree splines. Comput Aided Geom Design 2010;27(1):23–35. doi:10.1016/j.cagd.2009.08.005.

[8] Li X, Huang ZJ, Liu Z. A geometric approach for multi-degree spline. Journal of Computer Science and Technology 2012;27(4):841–50. doi:10.1007/s11390-012-1268-2.

[9] Shen W, Wang G, Yin P. Explicit representations of changeable degree spline basis functions. J Comput Appl Math 2013;238(1):39–50. doi:10.1016/j.cam.2012.08.017.

[10] Shen W, Yin P, Tan C. Degree elevation of changeable degree spline. Journal of Computational and Applied Mathematics 2016;300:56 – 67. doi:http://dx.doi.org/10.1016/j.cam.2015.11.030.

[11] Toshniwal D, Speleers H, Hiemstra RR, Hughes TJ. Multi-degree smooth polar splines: A framework for geometric modeling and isogeometric analysis. Comput Methods Appl Mech Engrg 2017;316:1005–61. doi:10.1016/j.cma.2016.11.009.

[12] Speleers H. Algorithm xxx: Computation of multi-degree B-splines; ???? ACM Trans. Math. Software, to appear (arXiv:1809.01598).

[13] Cox M. The numerical evaluation of B-splines. J Inst Maths Applics 1972;10:134–49.

[14] de Boor C. On calculating with B-splines. J Approx Theory 1972;6(1):50–62. doi:10.1016/0021-9045(72)90080-9.

[15] Wozny P. Construction of dual B-spline functions. J Comput Appl Math 2014;260:301–11. doi:10.1016/j.cam.2013.10.003.