Ph.D. Thesis in the Doctoral Program
Computer Science and Computational Mathematics

Tesi di Dottorato di Ricerca in
Informatica e Matematica del Calcolo

Ciclo XXXIII

Università degli Studi dell'Insubria
Dipartimento di Scienza e Alta Tecnologia

Solving the Hermite interpolation problem with scalar subdivision schemes

Author:

Rafael Díaz Fuentes

Advisors:

Dr. Mariarosa Mazza
Prof. Stefano Serra-Capizzano
Università degli Studi dell'Insubria

Academic year 2019/2020

A mis padres,
por regalarme un origen
y facilitarme el camino.

A mi hermano,
porque $1 + 1 = 1$.

"My work always tried to unite the truth with the beautiful
but when I had to choose one or the other,
I usually chose the beautiful."
Hermann Weyl

"Every solution lies within a problem,
if you have a problem, you are already halfway there."
Anonymous

"It is better to solve one problem five different ways,
than to solve five problems one way."
George Polya

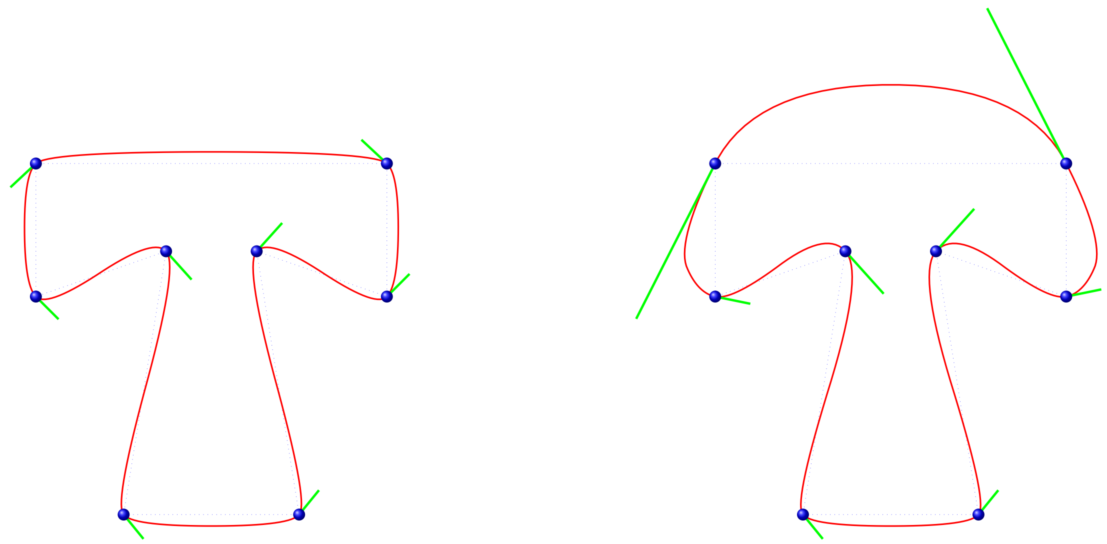
Abstract

Subdivision schemes represent an efficient and simple class of methods to generate curves and surfaces by successive refinements of a set of points with an associated connectivity defining polygons or meshes in the respective cases. Many applications require the possibility of interpolating points and associated derivatives. The interpolation is guaranteed with the use of interpolatory scalar and interpolatory Hermite subdivision schemes in each respective case. Moving beyond those schemes, in this thesis we study the point interpolation and the Hermite interpolation problems with curves by using the class of scalar linear uniform subdivision schemes.

The motivation for this research is the gaps evident in the literature regarding interpolation with certain approximating schemes. The gaps include the interpolation with dual subdivision schemes and the derivatives interpolation for any scalar scheme. We analyze both primal and dual cases taking into consideration odd and even symmetry of their masks. That analysis provides a characterization of the singularity for the interpolation operator represented with a block-circulant matrix. Our choice of interpolation parameters differs from the usual chosen ones at integer parameters, adds a degree of freedom, and offers the possibility of constructing a family of interpolating curves. In addition, when in the presence of a singular interpolation operator, we propose a filter for the least square solution based on the kernel of that operator. This strategy provides a solution which optimizes a given fairness functional. Under some considerations that choice is found with a quadratic optimization problem, avoiding the need of facing the optimization as other fitting solutions in the literature. With the strategies proposed our research resolves the outstanding problems. The results are used for the free-form design of curves, the exact offset computation, and an image segmentation algorithm based on subdivision curves.

Keywords: Scalar subdivision schemes, point interpolation, Hermite interpolation, free-form curve design, block-circulant matrices, image segmentation.

Mathematics Subject Classification (2020): 15A09, 15B05, 41A05, 65D05, 65D17, 65F22, 68U05, 68U10, 90C53.



Interpolation of points and associated tangent vectors.

Acknowledgments

A research project of this magnitude is never an independent endeavor, and it is not only office hours that influence the outcome. The work presented here is the result of the scientific collaboration and the company that I have had every day during this period.

First of all, I thank my supervisors Mariarosa Mazza and Stefano Serra-Capizzano, who guided this investigation to the end. Their patience and support made a significant part of this result comes true. Their orientation was not only dedicated to technical details but to the professional training this degree aims to achieve.

I would like not to overlook the merit of Lucia Romani and Alberto Viscardi for having contributed to the ideas that started this research. The discussions with Marco Donatelli along with his work as coordinator encouraged me with the research every moment of the journey, supporting ups and downs. Always present regardless of the distance and personal situation, I appreciate the company, support, and exchange of ideas with my advisor from previous degrees, Jorge Estrada Sarlabous.

The last stage ran alongside the stay in the CaNA Group under the direction of Giuseppe Rodriguez. I am deeply grateful for the hospitality and the conditions provided that favored the completion of this work.

Thank God I am never alone, and I received support from many friends, a very long list. In particular, I want to acknowledge Francesca, Michelle, Andrea, Isabella, Brenda, and Jorge, who contributed directly to the development of this work. Thanks to the Aquila, Alberti, Micheloni, and Poetti families, living in a foreign country has been like living at home. They have been a beautiful echo of my family's support with an ocean in between.

Last but not least, thanks to my parents Jorge Antonio and Neyda, my brother Jorge, my extended family, and everyone who trusted this result for their constant motivation.

Contents

List of Figures

Notations

- $\ell(\mathbb{Z})$: the space of all real-valued bi-infinite sequences defined on the set $\mathbb{Z}$ of all the integers.
- $\ell_0(\mathbb{Z})$: the subspace of $\ell(\mathbb{Z})$ with only finitely many non-zero elements.
- $\operatorname{supp}\mathbf{a} = [N, M]$: for a sequence $\mathbf{a} = \{a_j, j \in \mathbb{Z}\} \in \ell_0(\mathbb{Z})$, the *support* is the set of indexes corresponding to non-zero elements, i.e., $a_j = 0$ for all $j \notin [N, M]$.
- $\mathbf{A} + \mathbf{B} = \{a + b \,|\, a \in \mathbf{A} \subset \mathbb{R}, b \in \mathbf{B} \subset \mathbb{R}\}$: the Minkowski sum of two sets.
- $C_0 = C_0(\mathbb{R})$: the subspace of continuous functions in $C(\mathbb{R})$ with compact support, i.e., that vanishes outside some bounded interval.
- $\operatorname{supp}\varphi = [\mu, \nu]$: for a function $\varphi \in C_0$, $\varphi(t) = 0$ for all $t \leq \mu$ and $t \geq \nu$, with:

$$\mu :\, = \inf_t \{\varphi(t) \neq 0\}, \qquad \nu :\, = \sup_t \{\varphi(t) \neq 0\}.$$

- δ_j^0: Kronecker delta defined by:

$$\delta_j^0 = \begin{cases} 1, & j = 0 \\ 0, & j \neq 0. \end{cases}$$

- $A_{s,t}$: element in s-row and t-column of the matrix A. (All matrices and vectors are indexed starting from 1.)
- $A^\top$: transpose of matrix A (and analogous for vectors).
- A^*: conjugate transpose of A.
- $A^\dagger$: pseudo-inverse of matrix A.

- diag (v) = $\text{diag}_{s=1,\ldots,n}(v_s)$: diagonal matrix with the entries of $v \in \mathbb{C}^n$ in the diagonal. If $v \in \mathbb{C}^{d\times nd}$, then we refer to a block diagonal matrix with $d \times d$ blocks.

- $P_j(\ell)$: denotes the ℓ-th coordinate of the point $P_j \in \mathbb{R}^m$, which belongs to the ordered set of points $\mathbf{P} = \{P_j,\ j = 0,\ldots,n-1\}$.

  $\mathbf{P}(:,\ell)$: denotes the vector:

$$\mathbf{P}(:,\ell) = [P_0(\ell)\ldots P_{n-1}(\ell)]^\top.$$

- $\mathbf{0}_{p\times q}$: the null matrix of dimension $p \times q$.

- $\mathbf{1}_{p\times q}$: the matrix with all entries equal to 1 of dimension $p \times q$.

- $\mathbf{A} \otimes \mathbf{B}$: the *Kronecker matrix product* (or tensor product) of matrices $\mathbf{A} \in \mathbb{R}^{p\times q}$ and $\mathbf{B} \in \mathbb{R}^{r\times s}$ defined as:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \ldots & a_{1q}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \ldots & a_{2q}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1}\mathbf{B} & a_{p2}\mathbf{B} & \ldots & a_{pq}\mathbf{B} \end{bmatrix}.$$

- $Ker(A)$: the nullspace of the matrix A, i.e., $Ker(A) = \{w \mid Aw = 0\}$.

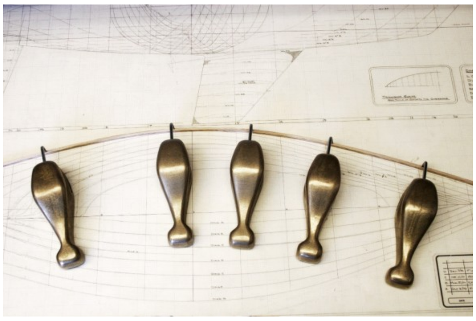
- $\Lambda(M)$: the spectrum of the matrix M, i.e., the set of eigenvalues of the matrix M.

- $\|x\|$: when non-specified, the norm considered is Euclidean norm $\|x\|_2$.

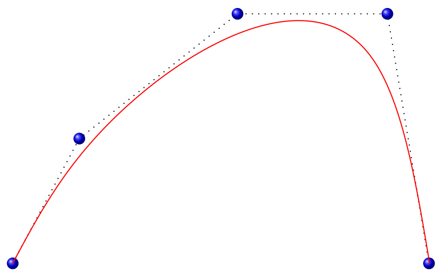
Chapter 1

Introduction

Curves and surfaces representation and generation is an old problem that has followed different approaches through the times, from physical meanings and mechanical constraints to aesthetic purposes. In particular, the design of ships provides the use of geometrical handles to achieve a desired shape. Those handles were metal weights known as *ducks* to shape the *splines*[1] and they were translated into curves and surfaces piecewise designed by using control points (see Fig. 1.1) [53].



(a) Mechanical splines drived by "ducks"



(b) B-spline curve designed by "control points"

Figure 1.1: Spline curve design

The splines are piecewise curves defined by a set of basis functions of certain space, in the particular case of piecewise polynomial space we can consider as an example of the well known *B-splines* basis [26]. In these settings a piecewise polynomial curve $c(t)$

[1] A flexible strip of wood.

can be represented as:

$$c(t) = \sum_{j=0}^{n} P_j B_j^{g,\tau}(t), \tag{1.1}$$

where the basis depends on a *knot partition* $\tau = \{\tau_j = j, j \in \mathbb{Z}\}$, so that

$$B_j^{g,\tau}(t) = \frac{t - \tau_j}{\tau_{j+g-1} - \tau_j} B_j^{g-1,\tau}(t) + \frac{\tau_{j+g} - t}{\tau_{j+g} - \tau_{j+1}} B_{j+1}^{g-1,\tau}(t) \tag{1.2}$$

and

$$B_j^{0,\tau}(t) = \begin{cases} 1, & t \in [\tau_j, \tau_{j+1}), \\ 0, & \text{otherwise.} \end{cases} \tag{1.3}$$

The coefficients (or coordinates with respect to that basis) are the so-called control points. As the name suggests, they allow to control the geometry of the curve with an intuitive control, based on their geometrical position on the space. Given that the basis is a partition of the unity, each point on the curve is a convex combination of those control points.

When the knots partition is uniform, i.e., $\tau = \{\tau_j = j, j \in \mathbb{Z}\}$, the basis functions are shifts of a single one $B_j^g(t) = B_0^g(t-j) = B^g(t-j)$. Additionally, they verify the *refinability property* [14]

$$B^g(t) = \frac{1}{2^g} \sum_{j=0}^{g+1} \binom{g+1}{j} B^g(2t - j). \tag{1.4}$$

The latter allows representation of the same curve by basis functions with a smaller support [24]:

$$c(t) = \sum_{j=0}^{n} P_j B^g(t-j) = \sum_{j=0}^{2n} \widehat{P}_j B^g(2t - j) \tag{1.5}$$

and

$$\widehat{P}_j = \sum_{s \in \mathbb{Z}} a_{j-2s} P_s. \tag{1.6}$$

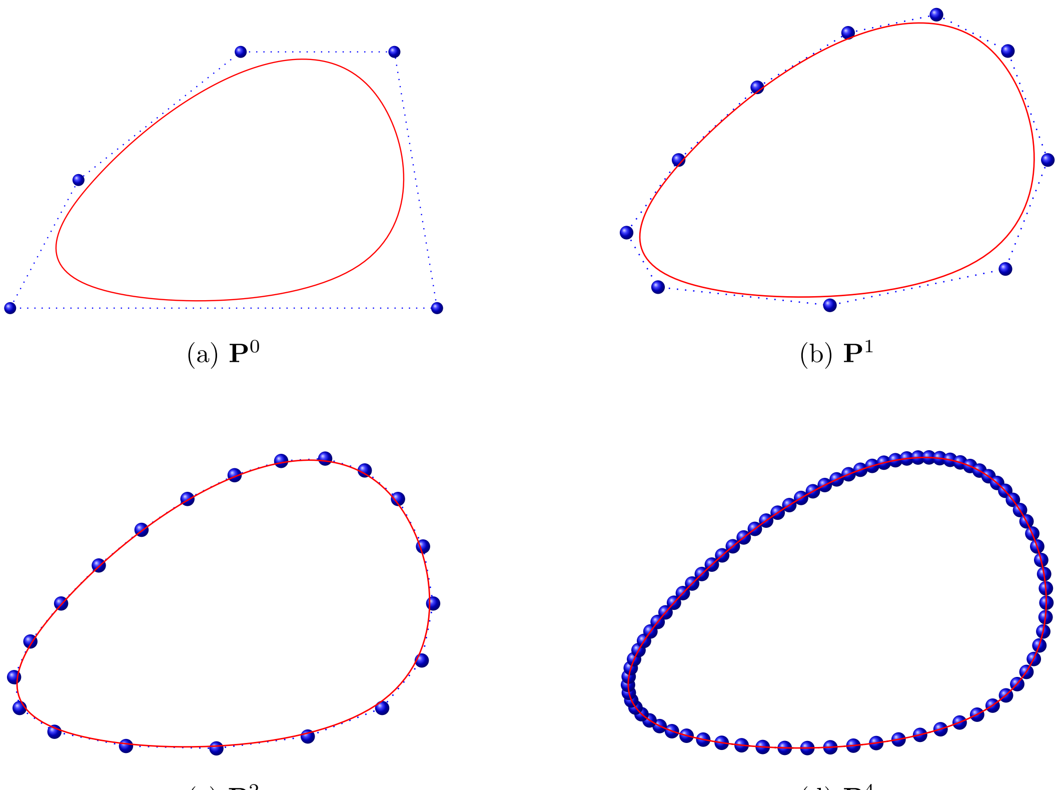
The new set of control points $\{\widehat{P}_j, j \in \mathbb{Z}\}$ are called a *refinement* of the previous set $\{P_j, j \in \mathbb{Z}\}$. From this, a sequence of control points $\mathbf{P}^k = \{P_j^k \in \mathbb{R}^d, j \in \mathbb{Z}\}$ can be produced, satisfying

$$P_j^{k+1} = \sum_{s \in \mathbb{Z}} a_{j-2s} P_s^k, \tag{1.7}$$

with $\mathbf{P}^0 = \{P_j^0 \in \mathbb{R}^d, j \in \mathbb{Z}\}$ the initial set of control points and $\mathbf{a} = \{a_j \in \mathbb{R}, j \in \mathbb{Z}\}$ the so-called *subdivision mask*. The previous transformation can be defined by a *refinement operator* S such that

$$S\mathbf{P}^k = \mathbf{P}^{k+1}. \tag{1.8}$$

Consequently, we can create a sequence of piecewise linear functions, interpolating each set $\mathbf{P}^k$, which converges uniformly to the curve $c(t)$ [77] (see Fig. 1.2).



(a) $\mathbf{P}^0$ (b) $\mathbf{P}^1$ (c) $\mathbf{P}^2$ (d) $\mathbf{P}^4$

Figure 1.2: Subdivision of the control points

This idea is related with the *subdivision schemes*, which are algorithms that generate curves and surfaces by repeated refinements of an initial polygonal or mesh [14, 45, 49, 103]. For example, the curve in Fig. 1.2 can be obtained by the rules:

$$\begin{cases} P_{2j}^{k+1} &= \frac{1}{8}P_{j-1}^k + \frac{3}{4}P_j^k + \frac{1}{8}P_{j+1}^k \\ P_{2j+1}^{k+1} &= \frac{1}{2}P_j^k + \frac{1}{2}P_{j+1}^k \end{cases}, \qquad j \in \mathbb{Z}, k \in \mathbb{N} \tag{1.9}$$

considering the periodization $P_{j+2^k n}^k = P_j^k$ and the initial control points $\mathbf{P}^0 = \{P_j^0 \in$

$\mathbb{R}^d, j = 0, \ldots, n-1\}$, which generates a cubic uniform B-spline curve[2].

Before entering in details regarding this topic, let us make some observations on the interpolation problem. The control points may not lie on the curve, as it can be seen in Fig. 1.1b and Fig. 1.2, for the respective cases of open and closed curves. Nevertheless, for some basis functions that could be possible.

A particular relevance of the interpolation requirement can be observed in Fig. 1.3. For a designer it is hard to model the shape with the control points (in blue), because there is no visual correlation the between points positions and the final shape. Instead, with the interpolated points (in red) the design is more intuitive as it works directly on the shape. In fact, Fig. 1.1a exhibits a tool which makes use of the interpolation property at chosen points.

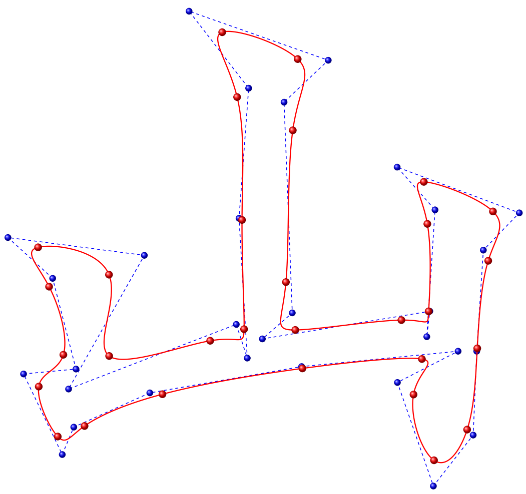


Figure 1.3: Ideogram design (meaning “mountain” in Chinese and Japanese)

Therefore, for many applications, a natural question arises on the possibility of interpolating data points and some derivatives. In other words, we could consider it as the *point interpolation problem*, the *Hermite interpolation problem*, or the *Birkhoff interpolation problem* with certain classes of curves generated by subdivision schemes.

In the first case we look for a curve $c(t)$ such that $c(t_j) = U_j^0$, $j = 0, \ldots, m-1$ for a given set of points $\mathbf{U}^0 = \{U_j^0,\ j = 0, \ldots, m-1\}$. For the Hermite case the curve is required to satisfy $c^{(k)}(t_j) = U_j^k$, $j = 0, \ldots, m-1$ with the sets $\mathbf{U}^k = \{U_j^k,\ j = 0, \ldots, m-1\}$, and $k = 0, \ldots, d$. The Birkhoff interpolation demands that certain derivatives have specified values at specified points. The latter does not enter into our

[2]In case of equidistant knots the term *cardinal B-splines* is also used [26]

discussion, while we address the first two.

Let us consider the set of control points $\mathbf{P}^0 = \{P_j^0, j = 0, \ldots, n-1\}$. Given a set of parameters $\mathbf{t} = \{t_j, j = 0, \ldots, n-1\}$, with the linear relation in (1.5) we define as a *direct problem* finding the values $c(t_j) = U_j^0$. As the values $c(t_j)$ depend linearly on the control points, it is possible to define an interpolating operator M, dependent on the parameters and the subdivision rule, such that

$$M\mathbf{P}^0 = \mathbf{U}^0. \tag{1.10}$$

On the contrary, given the set $\mathbf{U}^0$, if we fix the set of parameters $\mathbf{t}$, then we define as a *inverse problem* finding the control points $\mathbf{P}^0$ such that the curve $c(t)$ interpolates the points in $\mathbf{U}^0$ at the parameters in $\mathbf{t}$. That inverse problem defines what we call the interpolation problem and involves the same operator M defined for the direct case.

The generalization of the previous inverse problem to considering sets $\mathbf{U}^k$ with $k = 0, \ldots, d$ defines our Hermite interpolation problem.

The next section shows a brief survey on the interpolation with subdivision schemes.

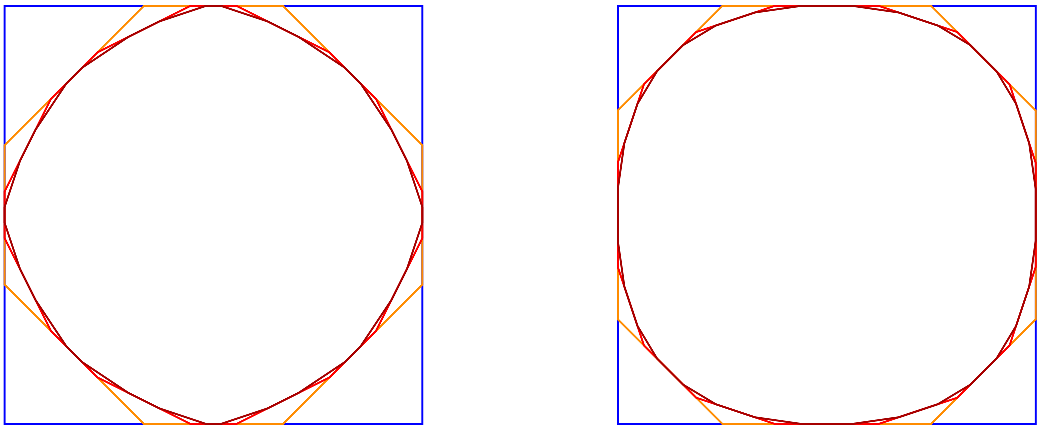
1.1 State of the art

The literature about subdivision schemes has grown since its origin in 1974. The present section covers a few selected works which we consider important to understanding the scope and objectives of this research. Since it is impossible to provide a complete list of references, a broad survey about the state of the art is provided in [102, 18].

1.1.1 Subdivision schemes

The subdivision schemes origins date back to de Rham's *corner cutting* iterations with trisection of a polygon edges [27] (see Fig. 1.4a). That was a method to manufacture hammer handles and provide them with a rounded profile from an initial rectangular shape. De Rahm proved that this method leads to a G^1 continuous curve[3] but not analytic. In 1974, Chaikin presented a similar method but splitting each edge by 1:2:1 that was proven later to generate *quadratic uniform B-spline* curves [11, 102] (see Fig. 1.4b).

[3] A curve is said G^1 continuous if it is tangent continuous.



(a) De Rahm (1947) (b) Chaikin (1974)

Figure 1.4: Corner cutting algorithms.

The next steps regarding subdivision schemes were done on uniform B-spline curves and then extended to other curves, some of them without a known closed form. The Bézier curves, which are a particular case of B-spline curves, have as well a subdivision rule of their control polygon to generate a sequence of polygons converging to the curve [54]. However, in those cases the subdivision is a local concept for each curve segment in the Bézier spline curve.

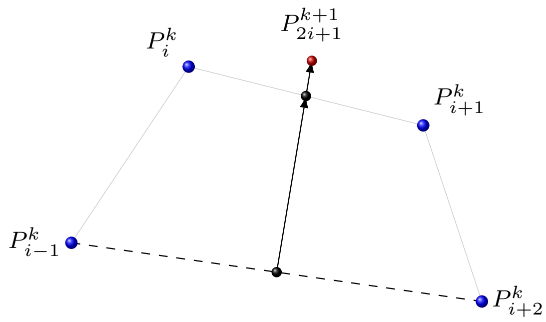
Subdivision schemes are classified in *interpolatory* and non-interpolatory, also called *approximating*. Other classifications include *scalar* and *vectorial*, *linear* and *non-linear*, *stationary* and *non-stationary*, among others [49, 102, 103].

1.1.2 Interpolatory subdivision schemes

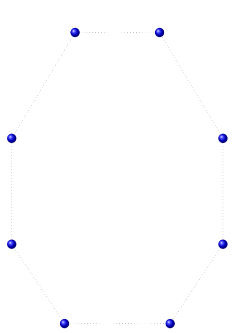
In 1986-1989 the first interpolatory schemes were proposed by Dubuc and Deslauriers by local interpolation of consecutive vertices with polynomials of a fixed degree [33]. Simultaneously, in 1987 Nira Dyn [50] et al. proposed a family of interpolatory schemes following a geometrical idea (see Fig. 1.5) but still with linear rule

$$\begin{cases} P_{2j}^{k+1} &= P_j^k, \\ P_{2j+1}^{k+1} &= -\omega P_{j-1}^k + \left(\omega + \frac{1}{2}\right) P_j^k + \left(\omega + \frac{1}{2}\right) P_{j+1}^k - \omega P_{j+2}^k, \end{cases} \qquad k \in \mathbb{N},\ j \in \mathbb{Z}. \quad (1.11)$$

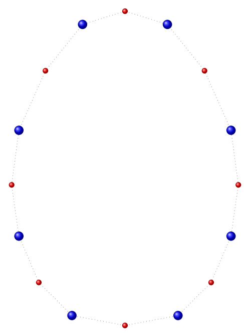
That family was based on a shape parameter ω and as a particular case $\omega = \frac{1}{16}$ it was possible to get the 4-point scheme proposed by Dubuc and Deslauriers. The subdivision curves generated with this scheme are C^1 for $\omega \in \left(0, \frac{\sqrt{5}-1}{8}\right)$ [42].



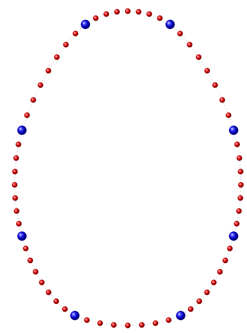
(a) Geometric interpretation of (1.11)



(b) Initial set of points



(c) First iteration



(d) Third iteration

Figure 1.5: The 4-point interpolatory subdivision scheme geometric rule and a few iterations of the scheme.

However, those schemes had lower continuity degree than the approximating scheme already proposed based on B-splines. In general, it is known so far that, for the same *support* of the *subdivision mask*[4], the approximating schemes has greater continuity of the derivatives than the interpolatory schemes.

Before [96] only primal interpolatory schemes were proposed, as all the dual[5] schemes we find in the literature are non interpolatory [23], except for those proposed in [96, 32]. The latter provides the interpolation property when the subdivision mask has infinite length, which lacks of local control of the geometry. In [98] the authors

[4]Those concepts are introduced later.

[5]The concept of primal and dual scheme is introduced in the next chapter.

investigate the dual interpolatory schemes with finite mask and propose a constructive method to produce new interpolatory schemes.

There have been many contributions to the literature on constructing interpolatory subdivision schemes from approximating schemes for curves and surfaces [71, 20, 21, 78]. Some of them construct the interpolating rules from the cardinal B-splines masks, while others use the limit positions of refinements in approximating schemes to propose an interpolating rule.

It follows then the natural question about how to interpolate control points directly with approximating schemes or how to produces subdivision schemes of higher smoothness that are able to interpolate the data. There are two possible ways, the first could be by solving the equations that allow to guess the control points to interpolate the provided data, the second is to define *Hermite subdivision schemes* that are able to interpolate points with associated derivatives, ensuring in this way the smoothness of the subdivision curve. Let us review first the second option.

1.1.3 Hermite subdivision schemes

The Hermite problem consists in interpolating points and their associated derivatives up to order m, which we refer to as the order of the *Hermite interpolation problem*. The classical problem has been extensively studied in the literature with polynomial solutions, where the Hermite problem of order m is solved with polynomial curves of degree $2m+1$ [118, 67]. The family of polynomial curves more utilized in design softwares have been the cubic Bézier curves (e.g., in Adobe, CorelDraw, OCAD). Likewise for every cubic curve two endpoints and associated tangent vectors are interpolated with a unique solution described by four control points. Those control points are the coefficients of the cubic Bernstein basis and provide an intuitive control of the geometry for the designer.

Besides polynomials curves, there are proposed solution with rational polynomials in Bernstein basis (rational Bézier curves), B-splines, and non-uniform rational B-splines [9, 55, 70, 80]. As an example of B-spline Hermite interpolation, we can consider the works of Plonka [94, 93], who studied the interpolation using B-splines with knots of multiplicity two. Even though many of those contributions have continuity degree C^{2m} as polynomial curves of degree $2m+1$, the interpolation of derivatives up to order m assures only a C^m continuity at the interpolated points. The contribution in [79] shows a solution with free parameters interpolating only first derivatives with a C^2

polynomial curve in quintic Bernstein basis.

In subdivision, the story started by Merrien work [84] following the local approach in [50]. The Hermite subdivision idea is to produce in each iteration, like (1.9), not only new points, but also associated functional data, such as derivatives or normals. His construction uses only two points on a function $f(a)$ and $f(b)$ with their derivatives $f'(a)$ and $f'(b)$ to sample that function at dyadic parameter points $f(x_k^n)$ as well as its derivatives, where:

$$x_k^n = a + k\frac{b-a}{2^n}, \quad k = 0, \dots, n. \tag{1.12}$$

In a first example (see Fig. 1.6), the subdivision rule is based on the evaluation of the unique Hermite cubic interpolant at the middle point of each interval

$$f(\tfrac{a+b}{2}) = \tfrac{1}{2}(f(a) + f(b)) - \tfrac{b-a}{8}(f'(b) - f'(a)), \tag{1.13}$$

$$f'(\tfrac{a+b}{2}) = \tfrac{3}{2(b-a)}(f(a) + f(b)) - \tfrac{1}{4}(f'(a) + f'(b)). \tag{1.14}$$

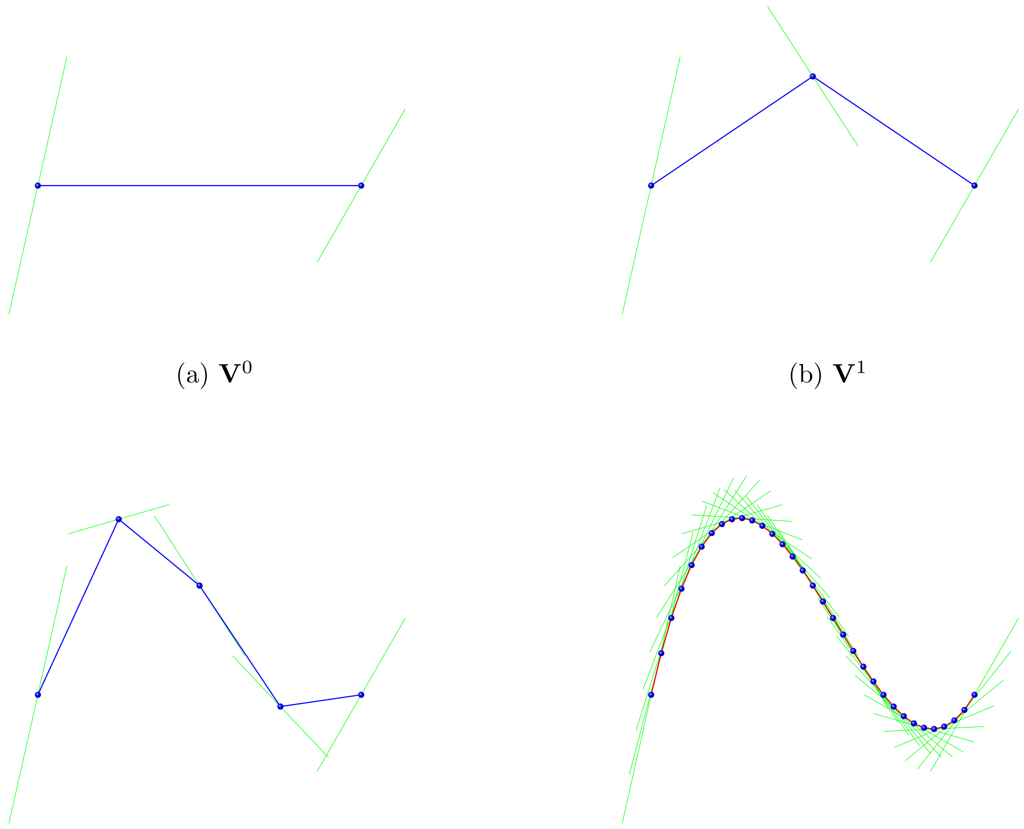
In that way, given the initial set $\mathbf{V}^0 = \{V_j^0 \in \mathbb{R}^{2\times d},\ j \in \mathbb{Z}\}$, the proposed family of Hermite subdivision schemes

$$\begin{cases} V_{2j}^{k+1} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} V_j^k, \\ V_{2j+1}^{k+1} &= \begin{bmatrix} \frac{1}{2} & \frac{\lambda}{2^k} \\ -\mu\ 2^k & \frac{1-\mu}{2} \end{bmatrix} V_j^k + \begin{bmatrix} \frac{1}{2} & -\frac{\lambda}{2^k} \\ \mu\ 2^k & \frac{1-\mu}{2} \end{bmatrix} V_{j+1}^k, \end{cases} \quad k \in \mathbb{N}, j \in \mathbb{Z}, \tag{1.15}$$

dependent on two parameters (λ, μ) is defined [84, 73]. For $(\lambda, \mu) = \left(\frac{1}{8}, \frac{3}{2}\right)$ the limit function is C^1 and the subdivision scheme (in (1.13) and (1.14)) converges to the unique Hermite cubic spline interpolating the initial data at integer parameters $[c(j), c'(j)]^\top = V_j^0 \in \mathbb{R}^{2,d}$. Meanwhile, for $(\lambda, \mu) = \left(\frac{1}{8}, 2\right)$ the limit function is a quadratic spline. It is known that Hermite schemes can be represented as spline curves [73].

This approach was extended later to Hermite interpolation on triangulations for surface generation [85]. Many schemes based on Merrien's construction appeared in the succesive years as extensions or generalization, both interpolatory and non interpolatory [73, 104, 87].

In 1999 Dyn and Levin provided some analysis tools for Hermite interpolatory schemes while considering any number of derivatives associated to the data points



(a) $\mathbf{V}^0$ (b) $\mathbf{V}^1$

(c) $\mathbf{V}^2$ (d) $\mathbf{V}^4$

Figure 1.6: A few iterations of Merrien's cubic Hermite subdivision scheme

[48]. Those tools were based on *divided-difference* operators applied on the succesive refinements $\mathbf{V}^k$ and the consideration of derived *stationary Hermite schemes*.

We can note that the Hermite subdivision rule (1.15) is non-stationary, in the sense that it depends on the iteration level k. We can rewrite them as

$$\begin{cases} V_{2j}^{k+1} &= \mathbf{D}^{-k-1} \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \mathbf{D}^k \; V_j^k, \\ \\ V_{2j+1}^{k+1} &= \mathbf{D}^{-k-1} \begin{bmatrix} \frac{1}{2} & \lambda \\ -2\mu & \frac{1-\mu}{4} \end{bmatrix} \mathbf{D}^k \; V_j^k + \mathbf{D}^{-k-1} \begin{bmatrix} \frac{1}{2} & -\lambda \\ 2\mu & \frac{1-\mu}{4} \end{bmatrix} \mathbf{D}^k \; V_{j+1}^k, \end{cases} \tag{1.16}$$

with $k \in \mathbb{N}, j \in \mathbb{Z}$, and $\mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$.

In this way we get a subdivision rule that can be written as in (1.7), this time with matrix coefficients

$$V_j^{k+1} = \sum_{s \in \mathbb{Z}} \mathbf{D}^{-k-1} \; A_{j-2s} \; \mathbf{D}^k \; V_s^k, \qquad j \in \mathbb{Z}, k \in \mathbb{N}. \tag{1.17}$$

The so-called stationary Hermite scheme in [48] considers the mask $\mathcal{A} = \{A_j, \; j \in \mathbb{Z}\}$. Further details on the analysis of convergence and smoothness of the Hermite schemes can be found in [40, 39, 41, 86, 17]. They introduce the *spectral condition* and the *Taylor operator* which determines the *Taylor scheme* associated with $\mathcal{A}$. Then, the study of convergence and smoothness is based on the properties of the derived Taylor scheme.

The contributions in [99, 95] construct Hermite subdivision rules based on cardinal B-splines of odd degree $2n+1$, $n \in \mathbb{N}$ with simple knots. The order of interpolated derivatives results n and the cubic case correspond to Merrien's scheme (1.15). In [99] the subdivision rule is extended for any *arity* (or *dilation factor*) of the refinement equation[6].

Another link between scalar and Hermite subdivision scheme is studied [81]. This research proposed a way to derive Hermite subdivision rules from scalar subdivision schemes. Let us consider a scalar stationary subdivision scheme with S as its refinement operator (1.8) and M^k the interpolating operator providing points on the subdivision curve and their associated derivatives from the control polygon $\mathbf{P}^k$. It is possible to define an Hermite subdivision scheme following the commutative diagram

$$\begin{tikzcd} \mathbf{P}^k \arrow[r, "M^k"] \arrow[d, "S"'] & \mathbf{V}^k \arrow[d, "\exists"] \\ \mathbf{P}^{k+1} \arrow[r, "M^{k+1}"'] & \mathbf{V}^{k+1} \end{tikzcd} . \tag{1.18}$$

However, there were some problems such as the ill-conditioning of the Hermite subdivision matrix constructed and the fact that for some scalar schemes, their convergence does not imply the derived vectorial scheme convergence.

[6]Equation (1.4) corresponds to a dilation factor 2 or, in other words, a binary refinement.

Although the idea behind (1.18) shows an interesting link between scalar and Hermite schemes, it is not practical in any case. Therefore, rather than explore that strategy, we study the Hermite problem with scalar schemes without trying such an approach.

This thesis does not provide a thorough analysis on Hermite subdivision schemes and any references made to distinguish them from the classic Hermite interpolation problem solved with scalar subdivision scheme, which is the goal of this document.

1.1.4 Interpolation with scalar subdivision schemes

What has been discussed so far provides the following options for interpolating data with subdivision schemes. In case of only points, interpolation can use an interpolatory scheme which has that data as initial control polygon or it can be used an approximating scheme and then used to compute an initial polygon so that the limit curve interpolates those points. In case of Hermite interpolation, we can use a Hermite subdivision scheme or a scalar subdivision scheme, with the same situation as before in case the Hermite scheme is non-interpolatory. In case of using a scalar subdivision scheme, a suitable set of initial control points has to be computed as a preprocessing of the data.

In [101], the authors proposed a local rule, named *retrofitting strategy*, to compute the initial control points such that a family of schemes called *J-splines* interpolates points. That rule can be translated into an iterative method that solve the interpolation problem (1.10). This method happens to fail when the iteration matrix has a spectral radius greater or close to 1, which is possible for certain values of the parameter that define the J-spline family.

Another iterative method was proposed in [118] for point interpolation with uniform B-spline curves, later generalized in [89] to interpolate tangent vectors and curvature with cubic B-spline curves. In this case, for tangent and curvature interpolation, the proposed iteration rules were non-linear and the authors did not provide a formal proof of convergence for that iterative method. As a restriction the authors only considered only interpolation of unitary tangent vectors which results in missing some curve features (see Fig. 1.7).

In relation to the solution of the linear system of equations in (1.10), another approach has been followed by other authors by considering the computation of the least square solution of those equations. In [62] a least square solution is proposed while introducing additional degrees of freedom to interpolate a mesh with the *Catmull-Clark*

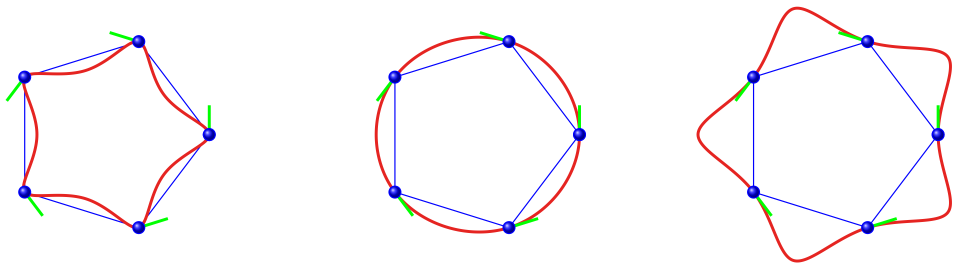


Figure 1.7: Different curves interpolating the same control points and prescribed tangents directions, but changing the length of those tangents.

surface subdivision scheme. The degrees of freedom are set by optimizing a *fairness functional* on the surface subject to a set of linear constraints given by the interpolation conditions. Meanwhile, in [65] a least square fitting approach is proposed with a surface subdivision scheme that generalizes the Loop surface subdivision scheme for triangular meshes. Althought those methods are devoted to subdivision sufaces, the nature as subdivision scheme and least square method applies naturally for both curves and surfaces.

1.2 Outline of the thesis

The problem we discuss in this thesis is the data interpolation with scalar subdivision schemes, mainly for the free-form design of curves where the referred data is scattered on the Euclidean space. Besides the use of interpolatory scalar schemes and interpolatory Hermite schemes, the approaches to interpolate points and associated derivatives described before [101, 89, 62, 65] do not cover all the cases present in subdivision. In particular, all of them are based on primal schemes and the extension to dual schemes is not considered. To fill this theoretical gap in the literature, we study the interpolation of points with scalar approximating schemes and the Hermite interpolation with scalar schemes rather than Hermite schemes or vectorial schemes. It is worth noting that scalar interpolatory schemes, (e.g., the four-point scheme [50] in (1.11)) do not interpolate associated tangent vectors.

The *second chapter* is dedicated to provide the background notions related to subdivision schemes and the tools used to solve the interpolation problem. Among those tools, a few energy functionals for describing the *fairness* of the curves are shown. The translation of the problem into a Linear Algebra model brings out the use of certain structured matrices. Therefore, the latter are introduced with some related concepts and results.

The *third chapter* shows the proposals to solve the point interpolation problem and the Hermite interpolation problem. First, a characterization of the singularity of the interpolation opertor depending on the subdivision mask is provided. When this operator is singular, we propose some alternatives for facing that situation. A comparison with iterative methods proposed in the literature is done. In addition, we compare the performance of the scalar schemes with respect to a few selected Hermite subdivision scheme as solutions of the Hermite interpolation problem.

The *fourth chapter* provides applications of the algorithms proposed to solve the interpolation problem. The first one is the free-form design of curves. The second is the generation of offset curves, considering both constant and variable distance from the subdivision curve. The last application studied is an image segmentation algorithm based on snake curves, which is a sequence of curves converging to the boundary of an object inside an image. The curves used are subdivision curves corresponding to any chosen scalar subdivision scheme.

In the *conclusion* we discuss the results of the thesis and we present a few selected open problems as well as possible future works.

Chapter 2

Background concepts and foundations

The problem we are solving is based on the use of scalar subdivision schemes. There are among others different classifications for them, including approximating or interpolatory, linear or non linear, geometrical, scalar or vectorial, stationary or non-stationary, uniform or non uniform. A broad survey can be found in [42, 102, 103, 49, 48, 14].

Some types are not independent, since geometrical subdivision schemes can be described as linear subdivision schemes. For instance, the well-known *4-point subdivision scheme* [50] is one such example.

In what follows we present the foundation and notations suitable for understanding our problem as well as the proposed solution strategies in *Chapter 3*. Therefore, we present basic definitions, properties, and results for uniform scalar subdivision schemes. Further details regarding other kinds of schemes like the Hermite ones, or the concept of using schemes of higher arities, are presented only for illustrating comparisons with our proposals [84, 73, 87, 99].

In order to deal with our problem, we use *Toeplitz* and *block-Toeplitz* matrices [107, 60] and specific functionals to measure geometrical features [116, 1] . Those are presented also here to make the presentation of the following chapters more synthetic.

2.1 Scalar subdivision schemes

This section is mainly devoted to providing the basis and tools regarding subdivision schemes that we use to model and solve the problem of interest. In this thesis we are not dealing with general subdivision schemes. Rather, we are examining the linear subdivision schemes. Inside this class we dedicate special attention to stationary uniform subdivision schemes. For more details concerning subdivision schemes, see [49, 42].

2.1.1 Basis function and refinability

Let $\varphi(t)\colon \mathbb{R} \to \mathbb{R}$ be a real continuous function with compact support in $[-N, M]$, with $N, M \in \mathbb{R}^+$ and $N > 1$, $M > 1$. Let us consider the vector space generated by the integer shifts of that *basis function* $\varphi(t)$:

$$\operatorname{span}\{\varphi(t-j),\ j \in \mathbb{Z}, t \in \mathbb{R}\}. \tag{2.1}$$

If we consider n points $\mathbf{P} = \{P_j \in \mathbb{R}^m, j = 0, \ldots, n-1\}$, then we can generate the continuous parametric curve $c(t)\colon \mathbb{R} \to \mathbb{R}^m$:

$$c(t) = \sum_{j=0}^{n-1} P_j \varphi(t-j) \in \operatorname{span}\{\varphi(t-j),\ j \in \mathbb{Z}\}, \qquad \text{for } t \in [-N, M+n-1].$$

It is easy to check that $c(t)$ has compact support in $[-N, M+n-1]$.

Provided some properties of the basis function, such as the partition of unity presented later, the coefficients P_j have an intuitive relation with the geometry of the curve $c(t)$. Therefore, more than merely coordinates in the basis (2.1), they are called *control points* of the curve, as their representation as points in $\mathbb{R}^m$ serves to control the shape of the curve.

Lemma 1. *Given n points $\{P_j \in \mathbb{R}^m, j = 0, \ldots, n-1\}$ and the basis function $\varphi(t)$ as described before, the curve*

$$c(t) = \sum_{j \in \mathbb{Z}} P_j\ \varphi(t-j) = \sum_{j=-M}^{N+n-1} P_j\ \varphi(t-j), \qquad t \in [0, n], \tag{2.2}$$

considering the periodization $P_j = P_{j+n}, j \in \mathbb{Z}$, is closed.

Proof. Let us prove that $c(t) = c(t+n)$, as:

$$\begin{aligned} c(t+n) &= \sum_{j \in \mathbb{Z}} P_j\ \varphi(t+n-j) \\ &= \sum_{j \in \mathbb{Z}} P_j\ \varphi(t-(j-n)) = \sum_{s \in \mathbb{Z}} P_{s+n}\ \varphi(t-s) \end{aligned}$$

with the change of variables $s = j - n$

$$= \sum_{s\in\mathbb{Z}} P_s \; \varphi(t-s) = c(t).$$

Therefore, the curve $c(t)$ is n-periodic in $\mathbb{R}$. As $c(0) = c(n)$, the restriction to the interval $[0, n]$ is a closed curve.

For the second equality, it is enough to check the evaluation in the edge cases $t = 0$ and $t = n - \varepsilon$ with $\varepsilon \in (0, 1)$. In those cases, provided that supp $\varphi = [-N, M]$, we get

$$c(0) = \sum_{j=-M}^{N} P_j \; \varphi(-j) = \sum_{j=-M}^{N+n-1} P_j \; \varphi(-j),$$

and

$$c(n-\varepsilon) = \sum_{j=-M+n}^{N+n-1} P_j \; \varphi(n-\varepsilon-j) = \sum_{j=-M}^{N+n-1} P_j \; \varphi(n-\varepsilon-j).$$

□

Remark 1. *Notice that the summation in* (2.2) *is not performed all over the indices, considering the compact support of the basis function. Actually, the summation is done in a subset of the indices depending on the support of* φ*.*

From now on, we use the infinite summation for simplicity, although it is a finite one as the previous Lemma states.

In this setting, each component is independent from the others, so without loss of generality, for some analysis we can consider the function $f\colon \mathbb{R} \to \mathbb{R}$ defined as

$$f(t) = \sum_{j\in\mathbb{Z}} p_j \varphi(t-j), \qquad t \in \mathbb{R}, p_j \in \mathbb{R}. \tag{2.3}$$

Indeed, the curve in (2.2) is defined component-wise in that way.

Remark 2. *If we consider the* cardinal data $\boldsymbol{\delta} = \{p_j = \delta_j^0,\ j = 0, \ldots, n-1\}$*, with* δ_j^0 *being the Kronecker delta sequence, then we find the basis function in* (2.3)*. It is a trivial observation, but it is useful for what follows.*

Linear uniform subdivision schemes are related to *refinable basis functions* $\varphi(t)$ that satisfies the relation

$$\varphi(t) = \sum_{j \in I \subset \mathbb{Z}} a_j \varphi(2t - j), \quad t \in \mathbb{R}, a_j \in \mathbb{R}. \tag{2.4}$$

The previous property shows a self-similarity of the basis function with respect to integer shifts and dilation by 2.

The *refinement sequence* $\mathbf{a} = \{a_j, j \in \mathbb{Z}\} \in \ell_0(\mathbb{Z})$ associated with the refinable function introduces us to the idea of subdivision. But before describing this concept, let us illustrate some well-known examples.

Examples:

The quadratic uniform B-splines $B^2(t)$ defined for the knots $\{-1, 0, 1, 2\}$ (see (1.2)) in (1.4) satisfies the refinement relation

$$B^2(t) = \frac{1}{4}B^2(2t+1) + \frac{3}{4}B^2(2t) + \frac{3}{4}B^2(2t-1) + \frac{1}{4}B^2(2t-2), \tag{2.5}$$

for $t \in [-1, 2]$.

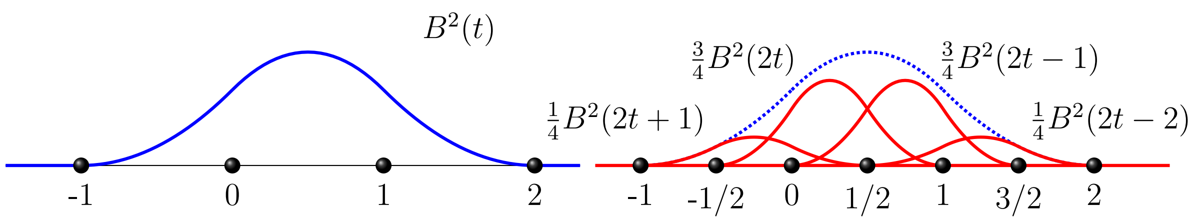


Figure 2.1: Refinability of quadratic uniform B-splines

The cubic uniform B-splines $B^3(t)$ defined for the knots $\{-2, -1, 0, 1, 2\}$ satisfies the refinement relation

$$B^3(t) = \frac{1}{8}B^3(2t+2) + \frac{1}{2}B^3(2t+1) + \frac{3}{4}B^3(2t) + \frac{1}{2}B^3(2t+1) + \frac{1}{8}B^3(2t+2), \tag{2.6}$$

for $t \in [-2, 2]$.

□

The length of $\mathbf{a}$ is related with the support of $\varphi(t)$ as described in the following result.

Theorem 1 ([14]). *Let $\varphi(t)$ be a refinable function with refinement sequence $\mathbf{a}$ satisfying supp $\mathbf{a} = [-N, M] \cap \mathbb{Z}$, then supp $\varphi = [-N, M]$.*

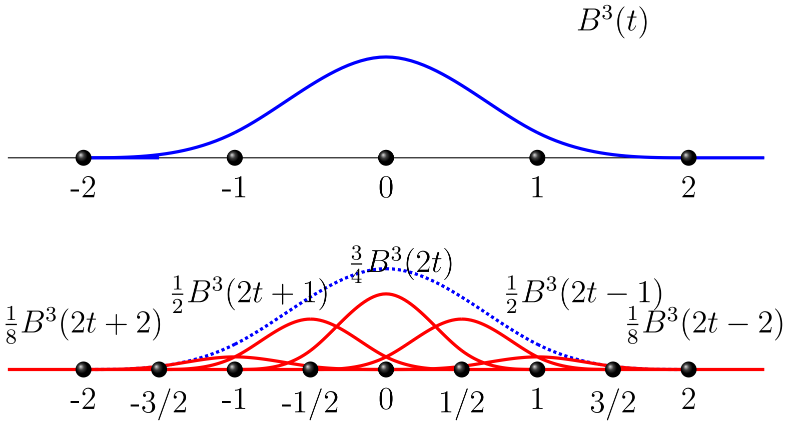


Figure 2.2: Refinability of cubic uniform B-splines

Notice that not every refinable function can be used for representing subdivision curves, except the *scaling function* [14].

Definition 1 (Scaling function)**.** *Let* $\varphi(t) \in C_0$ *a refinable function and* $\mathbf{a} \in \ell_0(\mathbb{Z})$ *its refinement sequence:*

$$\varphi(t) = \sum_{j\in\mathbb{Z}} a_j \varphi(\rho t - j), \qquad t \in \mathbb{R}.$$

Then φ *is called a* ρ-refinable function *with refinement sequence* $\mathbf{a}$.

Furthermore, if φ *satisfies*

$$\int_{-\infty}^{\infty} \varphi(t)\mathrm{d}t = 1,$$

then the refinable function is called a scaling function.

Remark 3. *If a refinable function* φ *satisfies*

$$\int_{-\infty}^{\infty} \varphi(t)\mathrm{d}t = c \neq 0, \tag{2.7}$$

then $\frac{1}{c}\varphi$ *is a scaling function.*

From this definition we obtain an important property which is used later.

Lemma 2. *If* $\mathbf{a} = \{a_j, j \in \mathbb{Z}\} \in \ell_0(\mathbb{Z})$ *is the refinement sequence of a refinable function* φ *that satisfies* (2.7)*, then*

$$\sum_{j\in\mathbb{Z}} a_j = \rho. \tag{2.8}$$

Proof. Integrating both sides of the refinement relation we obtain

$$\begin{aligned}\int_{-\infty}^{\infty}\varphi(t)\mathrm{d}t &= \int_{-\infty}^{\infty}\sum_{j\in\mathbb{Z}}a_j\varphi(\rho t-j)\mathrm{d}t\\ &= \sum_{j\in\mathbb{Z}}a_j\int_{-\infty}^{\infty}\varphi(\rho t-j)\mathrm{d}t = \frac{1}{\rho}\sum_{j\in\mathbb{Z}}a_j\int_{-\infty}^{\infty}\varphi(\rho t-j)\mathrm{d}(\rho t-j)\\ &= \frac{1}{\rho}\int_{-\infty}^{\infty}\varphi(t)\mathrm{d}t\sum_{j\in\mathbb{Z}}a_j\end{aligned}$$

recalling that the summation is indeed finite. Provided that $\int_{-\infty}^{\infty}\varphi(t)\mathrm{d}t = c \neq 0$, we arrive at the desired result. □

Definition 2 (Partition of unity)**.** *A function $\varphi \in C_0$ is said to provide a partition of unity if*

$$\sum_{j\in\mathbb{Z}}\varphi(t-j) = 1, \quad t\in\mathbb{R}, \tag{2.9}$$

where the summation is finite since φ has compact support.

Remark 4. *In what follows we make use of the dilation factor 2, which corresponds to 2-refinable functions. The latter consideration simplifies the discussion, but all the following results can be extended to a general dilation factor.*

Let us now consider again the closed curve in (2.2) represented in the basis of integer shifts of a scaling function φ with refinement sequence $\mathbf{a} \in \ell_0(\mathbb{Z})$. If we use the refinement relation for each element of the basis, then we deduce

$$\begin{aligned}c(t) &= \sum_{j\in\mathbb{Z}}P_j\varphi(t-j) = \sum_{j\in\mathbb{Z}}P_j\sum_{s\in\mathbb{Z}}a_s\varphi(2(t-j)-s)\\ &= \sum_{j\in\mathbb{Z}}\sum_{s\in\mathbb{Z}}P_ja_s\varphi(2t-(s+2j)) = \sum_{s\in\mathbb{Z}}\sum_{j\in\mathbb{Z}}P_ja_{s-2j}\varphi(2t-s)\\ &= \sum_{s\in\mathbb{Z}}\widehat{P}_s\varphi(2t-s),\end{aligned} \tag{2.10}$$

where

$$\widehat{P}_s = \sum_{j\in\mathbb{Z}}a_{s-2j}P_j. \tag{2.11}$$

Recalling that each summation is finite, the switch of summation order is justified.

From this, we can notice that the curve $c(t)$, parametrized in $[0,n]$ in (2.2), is represented in the basis $\{\varphi(2t-j), j \in \mathbb{Z}, t \in \mathbb{R}\}$. In this new basis the amount of control points is twice the amount of control points given in the previous basis (2.1).

If we denote by $\mathbf{P}^0 = \{P_j^0 \in \mathbb{R}^m, j = 0, \ldots, n-1\}$ the initial control points for the representation of the curve $c(t)$, and by $\mathbf{P}^1 = \{P_j^1 \in \mathbb{R}^m, j = 0, \ldots, 2n-1\}$ the ones obtained in (2.11), then we can generate a sequence of such sets by the rule

$$P_s^{k+1} = \sum_{j \in \mathbb{Z}} a_{s-2j} P_j^k. \tag{2.12}$$

The closed subdivided control polygon preserves the periodicity in (2.2) [14].

Lemma 3 (Preservation of periodicity)**.** *Let* $\mathbf{P}^0 \in \ell(\mathbb{Z})$ *denote the set of* n *initial control points in* $\mathbb{R}^m$*, for* $m \geq 2$*, such that the periodicity condition*

$$P_{j+n}^0 = P_j^0, \qquad j \in \mathbb{Z},$$

is satisfied. Then, for the refinement sequence $\mathbf{a} \in \ell_0(\mathbb{Z})$*, the sequence* $\mathbf{P}^k$ *generated recursively in* (2.12)*, is also periodic, with*

$$P_{j+2^k n}^0 = P_j^0, \qquad j \in \mathbb{Z}, k \in \mathbb{N}.$$

By induction it can be proved that the closed curve $c(t)$ is represented as

$$c(t) = \sum_{j \in \mathbb{Z}} P_j^k \ \varphi\left(2^k t - j\right), \qquad t \in \mathbb{R}, k \in \mathbb{N}. \tag{2.13}$$

The sequence obtained with the sets $\mathbf{P}^k = \{P_j^k \in \mathbb{R}^m, j = 0, \ldots, n-1\}$, $k \in \mathbb{N}$, defines the *subdivision scheme* (see Fig. 1.2). The relation in (2.12) is known as *subdivision rule* and the refinement sequence forms the so-called *subdivision mask* $\mathbf{a} = \{a_j, j \in \mathbb{Z}\}$.

In addition, from (2.2) follows that it is possible to compute as well the derivatives of the subdivision curves as

$$\frac{\mathrm{d}}{\mathrm{d}t} c(t) = 2^k \sum_{j \in \mathbb{Z}} P_j^k 2^k \frac{\mathrm{d}}{\mathrm{d}t} \ \varphi\left(2^k t - j\right), \qquad t \in \mathbb{R}, k \in \mathbb{N}. \tag{2.14}$$

2.1.2 Linear subdivision schemes

A subdivision scheme is an iterative method where a curve is generated by repeated refinements of an initial polygon with the set of control points as vertices. The case we are dealing with in this thesis is the linear case, as the rule (2.12) to generate each refinement is linear.

As stated previously, in this thesis we use scaling functions with dilation factor 2, consequently the theory in connection with subdivision schemes is the one concerning *binary schemes*. In case of considering higher dilation factors, the schemes under consideration are of higher *arities* [34, 115, 64, 103, 98].

Definition 3. *For a given mask* $\mathbf{a} \in \ell_0(\mathbb{Z})$*, the* subdivision operator $\mathcal{S}_{\mathbf{a}} \colon \ell(\mathbb{Z}) \to \ell(\mathbb{Z})$ *associated to* $\mathbf{a}$ *is defined by:*

$$(\mathcal{S}_{\mathbf{a}}\mathbf{P})_j := \sum_{s\in\mathbb{Z}} a_{j-2s} P_s, \qquad j \in \mathbb{Z}, \tag{2.15}$$

where $\mathbf{P} = \{P_j, j \in \mathbb{Z}\} \in \ell(\mathbb{Z})$*.*

By iterating the subdivision operator repeatedly to an initial set of control points $\mathbf{P}^0 = \left\{P^0_j, j \in \mathbb{Z}\right\} \in \ell(\mathbb{Z})$ we obtain the sequence of refinements given below:

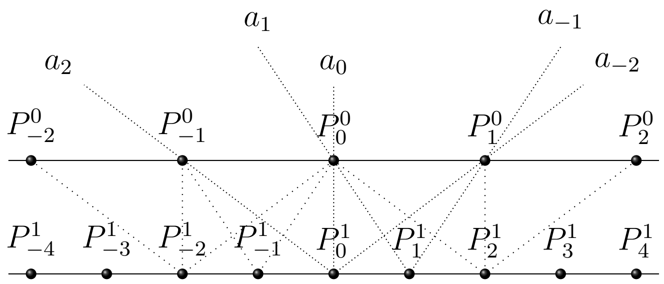
$$\mathbf{P}^0 \longmapsto \mathbf{P}^1 = \mathcal{S}_{\mathbf{a}}\mathbf{P}^0 \longmapsto \mathbf{P}^2 = \mathcal{S}_{\mathbf{a}}\mathbf{P}^1 \longmapsto \dots . \tag{2.16}$$

Returning to *Remark 2*, the basic function φ can be sampled with a parameterization at dyadic parameters by the sequence of refinements of the cardinal data

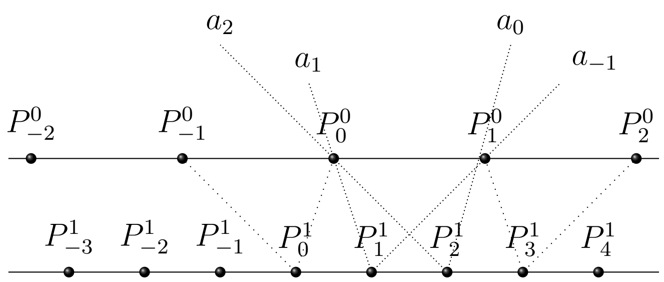
$$\varphi = \lim_{k\to\infty} \mathcal{S}^k_{\mathbf{a}}\, \boldsymbol{\delta}. \tag{2.17}$$

The subdivision operator applied in each step maps different dimensions with respect to other steps. Taking into account that the number of points in $\mathbf{P}^{k+1}$ is twice the number of points in $\mathbf{P}^k$, the last statement is easier to check. Therefore, the matrix representation of the subdivision operator in (2.15) depends on the amount of initial points and the level of refinement applied. To provide an eigenanalysis of that operator, we introduce the concept of *local subdivision operator* that maps a neighborhood of each point at the k-refinement onto a neighborhood with the same "structure" at the $(k+1)$-refinement (see Fig. 2.3) [103].

The parity of the number of elements in the support of $\mathbf{a}$ defines what is termed as



(a) Primal scheme with mask $\{a_{-2}, a_{-1}, a_0, a_1, a_2\}$



(b) Dual scheme with mask $\{a_{-1}, a_0, a_1, a_2\}$

Figure 2.3: Refinement structure for primal and dual subdivision schemes in a neighborhood of P_0^0.

the *primality* or *duality* of a subdivision scheme [103]. An odd number of elements in the support of **a** provides a *primal scheme* (see Fig. 2.3a), where old vertices map into new vertices and another new vertices are inserted. On the other hand, if there is an even amount of elements, it is defined a *dual scheme*. In this case, old edges map into new edges and another edges are inserted.

Examples:

Let us provide two examples for the local subdivision matrix, one for each type related with the previous example of quadratic and cubic uniform B-splines. In order to make more clear the matrix structure, we consider the general case, and more precisely, the masks $\{a_{-2}, a_{-1}, a_0, a_1, a_2\}$ for a primal scheme and $\{a_{-1}, a_0, a_1, a_2\}$ for a dual scheme.

In the primal case, we consider as local subdivision matrix:

$$S_{\mathbf{a}} = \begin{bmatrix} a_2 & a_0 & a_{-2} & 0 & 0 \\ 0 & a_1 & a_{-1} & 0 & 0 \\ 0 & a_2 & a_0 & a_{-2} & 0 \\ 0 & 0 & a_1 & a_{-1} & 0 \\ 0 & 0 & a_2 & a_0 & a_{-2} \end{bmatrix} \quad \text{such that} \quad S_{\mathbf{a}} \begin{bmatrix} P^0_{j-2} \\ P^0_{j-1} \\ P^0_j \\ P^0_{j+1} \\ P^0_{j+2} \end{bmatrix} = \begin{bmatrix} P^1_{2j-2} \\ P^1_{2j-1} \\ P^1_{2j} \\ P^1_{2j+1} \\ P^1_{2j+2} \end{bmatrix}. \tag{2.18}$$

On the other hand, for a dual scheme we consider the matrix:

$$S_{\mathbf{a}} = \begin{bmatrix} a_2 & a_0 & 0 & 0 \\ 0 & a_1 & a_{-1} & 0 \\ 0 & a_2 & a_0 & 0 \\ 0 & 0 & a_1 & a_{-1} \end{bmatrix} \quad \text{such that} \quad S_{\mathbf{a}} \begin{bmatrix} P^0_{j-1} \\ P^0_j \\ P^0_{j+1} \\ P^0_{j+2} \end{bmatrix} = \begin{bmatrix} P^1_{2j-1} \\ P^1_{2j} \\ P^1_{2j+1} \\ P^1_{2j+2} \end{bmatrix}. \tag{2.19}$$

□

Definition 4. *For a given mask* $\mathbf{a} \in \ell_0(\mathbb{Z})$ *with support in* $[\mu, \nu]$*, associated to a subdivision scheme, the* local subdivision matrix $S_{\mathbf{a}}$ *is defined as:*

$$S_{\mathbf{a}} = (a_{s-2j-\mu+1})_{s,j=1}^{\nu-\mu+1} \tag{2.20}$$

such that

$$S_{\mathbf{a}} \begin{bmatrix} P^0_{j+\mu} \\ P^0_{j+\mu+1} \\ \vdots \\ P^0_j \\ \vdots \\ P^0_{j+\nu-1} \\ P^0_{j+\nu} \end{bmatrix} = \begin{bmatrix} P^1_{2j+\mu} \\ P^1_{2j+\mu+1} \\ \vdots \\ P^1_{2j} \\ \vdots \\ P^1_{2j+\nu-1} \\ P^1_{2j+\nu} \end{bmatrix}. \tag{2.21}$$

The subdivision scheme converges if the sequence of piecewise linear functions $\mathbf{f}^k(t)$ which satisfies the interpolation conditions

$$\mathbf{f}^k\left(\frac{j}{2^k}\right) = P^k_j, \quad j \in \mathbb{Z} \tag{2.22}$$

converges uniformly to a limit *subdivision curve.* The curve $c(t)$ in (2.2) is the continuous limit function

$$c(t) = \lim_{k\to\infty} \mathbf{f}^k(t). \tag{2.23}$$

In practice, just a few iterations are enough to provide a polygon that looks smooth for the human eye.

In the subdivision rule (2.12), for each point every component is independent from the others, so the analysis of convergence can be stated for functional data as pointed before in (2.3). Taking that into account, we can formulate the following definition [45].

Definition 5. *A binary subdivision scheme is said to be C^d if for every initial data $\mathbf{P}^0 \in \ell(\mathbb{Z})$, there exist a limit function $f \in C^d(\mathbb{R})$ such that:*

$$\lim_{k\to\infty} \sup_{j\in\mathbb{Z}} \left| P_j^k - f\left(\frac{j}{2^k}\right) \right| = 0, \tag{2.24}$$

and $f \neq 0$ for some initial data $\mathbf{p}^0$.

This definition can be extended straightforward to higher arities.

A necessary condition for the uniform convergence is that [45]:

$$\sum_{j\in\mathbb{Z}} a_{2j} = \sum_{j\in\mathbb{Z}} a_{2j+1} = 1. \tag{2.25}$$

Remark 5. *We do not cover further details concerning convergence and smoothness analysis of subdivision schemes, such as sufficient conditions and tools to check when they are satisfied. That topic is not within the scope of our interest, as this research considers known convergent subdivision schemes, instead of deriving new subdivision rules. Further details can be found in [103, 45, 49, 14].*

From (2.2) it follows that the *basic function* $\varphi(t)$ can be obtained by iterating the initial data $\mathbf{P}^0 = \{\delta_j^0, j = 0, \dots, n-1\}$.

When the mask is such that $a_{2j} = \delta_j^0$, for $i \in \mathbb{Z}$, then $\varphi(0) = 1$ and the subdivision scheme is said to be interpolatory (see (1.11)), as $c\left(\frac{j}{2^k}\right) = P_j^k$ for all $i, k \in \mathbb{Z}$. Thus, the points in $\mathbf{P}^k$ are a sampling of the curve $c(t)$ for every $k \in \mathbb{N}$. Instead, in the case of approximating subdivision schemes, to know the values $c(\frac{j}{2^k})$ it is necessary to compute the values of $\varphi\left(\frac{1}{2^k}\mathbb{Z}\right)$ and evaluate in (2.13). In particular, the values $\beta_j = \varphi(j)$ for $j \in \mathbb{Z}$, provide the so-called *first limit stencil* and

$$c(i) = V_i = \sum_{j\in\mathbb{Z}} \beta_{i-j} P_j^0. \tag{2.26}$$

The equation (2.26) can be represented as

$$M_n\mathbf{P}^0 = \begin{bmatrix} \beta_0 & \beta_{-1} & \beta_{-2} & \dots & \beta_2 & \beta_1 \\ \beta_1 & \beta_0 & \beta_{-1} & \dots & \beta_3 & \beta_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \beta_{-1} & \beta_{-2} & \beta_{-3} & \dots & \beta_1 & \beta_0 \end{bmatrix} \begin{bmatrix} P_0^0 \\ P_1^0 \\ \vdots \\ P_{n-1}^0 \end{bmatrix} = \begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_{n-1} \end{bmatrix} = \begin{bmatrix} c(0) \\ c(1) \\ \vdots \\ c(n-1) \end{bmatrix}. \tag{2.27}$$

We refer to M_n as the matrix that represents the *point interpolation operator* for linear subdivision schemes.

In general we could speak about d-th limit stencil. Let $V_j^k = c^{(k)}(j)$ be the k-th derivative of the curve $c(t)$ at $t = j$, $j \in \mathbb{N}$. This case can also be represented by a matrix. In this case, better described as a block-matrix which is shown and discussed later.

Definition 6. *The ordered set* $\boldsymbol{\beta}^{k-1} = \left\{\beta_j^{k-1} = \varphi^{(k-1)}(j),\ j \in \mathbb{Z}\right\},\ k = 1, 2, \ldots,$ *is called* k-th limit stencil *(or limit stencil of order k) and by* (2.2) *it holds that the k-th derivative of the curve at integer parameters can be computed as*

$$c^{(k)}(j) = V_j^k = \sum_{s\in\mathbb{Z}} \beta_{j-s}^k P_s^0. \tag{2.28}$$

The condition in (2.25) ensures the existence of $\varphi^{[1]}(t)$ from $\varphi(t)$ such that [45]

$$\frac{\mathrm{d}\varphi(t)}{\mathrm{d}t} = \varphi^{[1]}(t+1) - \varphi^{[1]}(t) \tag{2.29}$$

and therefore from (2.2)

$$c'(t) = \sum_{j\in\mathbb{Z}} (P_j^0 - P_{j-1}^0)\varphi^{[1]}(t-j). \tag{2.30}$$

It turns that $\varphi^{[1]}(t)$ is a refinable function with the mask given below

$$a_{2j}^{[1]} = 2\sum_{s=0}^{j} (a_{2s} - a_{2s-1}) \tag{2.31}$$

$$a_{2j+1}^{[1]} = 2\left[\sum_{s=0}^{j-1} (a_{2s-1} - a_{2s}) + a_{2j-1}\right]. \tag{2.32}$$

In analogous way, by recursion, $\varphi^{[k]}(t)$ can be defined if there exists a C^k limit curve for the subdivision scheme. In such a way the k-th limit stencil can be computed from the evaluation of $\varphi^{[k]}(t)$ in $\mathbb{Z}$.

One of the usual requirements is the symmetry[1] of the mask.

Definition 7. *A subdivision scheme is said to be* odd-symmetric *if* $a_{-j} = a_j$ *and*

[1] Here symmetry makes reference to the index.

even-symmetric *if* $a_{1-j} = a_j$ *for* $j \in \mathbb{N}$.

These symmetries are particular cases of the primal and dual form of subdivision schemes respectively, and the limit stencils have the same kind of symmetries [103].

As a consequence the odd-order limit stencil (recall *Definition 6*) inherits the odd or even symmetry

$$\beta_{-j}^{2d} = \begin{cases} \beta_j^{2d}, & \text{for odd-symmetric schemes,} \\ \beta_{j+1}^{2d}, & \text{for even-symmetric schemes,} \end{cases} \qquad d \in \mathbb{N}. \tag{2.33}$$

For the even-order limit stencil then we obtain

$$\beta_{-j}^{2d+1} = \begin{cases} -\beta_j^{2d+1}, & \text{for odd-symmetric schemes,} \\ -\beta_{j+1}^{2d+1}, & \text{for even-symmetric schemes,} \end{cases} \qquad d \in \mathbb{N}. \tag{2.34}$$

For $d \geq 1$ we have $\sum\limits_{j \in \mathbb{Z}} \beta_j^d = 0$ and $\sum\limits_{j \in \mathbb{Z}} \beta_j^0 = 1$.

As can be seen from (2.2), changing the position of the initial control points changes the geometry of the curve. Therefore, for free-form design of curves, among other applications, the relation between the position of the points in $\mathbf{P}^0$ and the geometry of the subdivision curve $c(t)$ should be intuitive. In particular, it is relevant to solve either the interpolation problem of finding the control points $\mathbf{P}^0$ given the points $\{V_j,\ j = 0, \ldots, n-1\}$ in (2.28) or the Hermite interpolation problem of finding $\mathbf{P}^0$ given $\{V_j,\ j = 0, \ldots, n-1\}$ and certain derivatives on those points.

2.1.3 Exact evaluation of linear uniform stationary subdivision schemes

In case of the interpolatory subdivision schemes, as $\varphi(j-s) = \delta_j^s$, we have

$$c\left(\frac{j}{2^k}\right) = \sum_{s \in \mathbb{Z}} P_s^k \delta_j^s = P_j^k. \tag{2.35}$$

Moreover, because of (2.22) and (2.23) it holds

$$c\left(\frac{j}{2^k}\right) = \lim_{q \to \infty} \mathbf{f}^{k+q}\left(\frac{2^q j}{2^{k+q}}\right) = \lim_{q \to \infty} P_{2^q j}^{k+q}, \qquad q \in \mathbb{N}. \tag{2.36}$$

In case of interpolatory schemes, the center and right hand expressions are limits of constant sequences that converge to the point P_j^k as stated in (2.35). Otherwise, the right hand sequence allows to analyze the exact value of $c\left(j/2^k\right)$.

Remark 6. *A subdivision curve $c(t)$ is usually represented by a polygon (that we defined as $\mathbf{f}^k$ in (2.22)) that interpolates $\mathbf{P}^k$ whose vertices are obtained after some refinements of an initial polygon $\mathbf{P}^0$. For the sake of simplicity, we shall refer to $\mathbf{P}^k$ as the set of refined control points in the k iteration, as well as the related polygon, called the* control polygon*, whose vertices are given by the previous ordered set of points. For computations we write the involved control polygon as vector like in (2.27).*

When k increases the control polygon $\mathbf{P}^k$ provides a better approximation of $c(t)$. In this research we approximate $c(t)$ by the polygon $\{c(j/2^k), j \in \mathbb{Z}, k \in \mathbb{N}\}$ whose vertices are on the curve. For interpolatory subdivision schemes both $\mathbf{P}^k$ and $\{c(j/2^k)\}$ are the same, but they are different in the case of non-interpolatory schemes. In this way we use a unified framework to represent the subdivision curve, always by points lying on the curve.

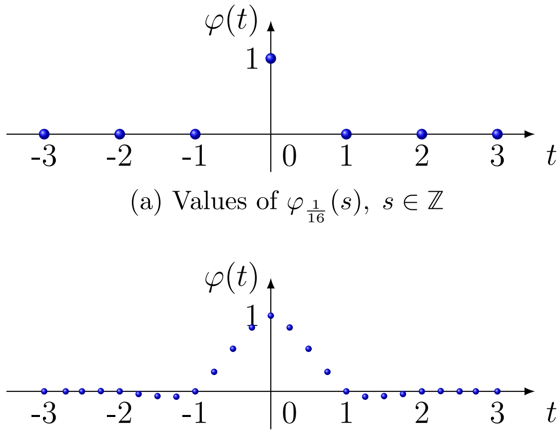
In many applications, no matter if the scheme is interpolatory or not, it is enough to work with $\mathbf{P}^k$ instead of $\{c(j/2^k), j \in \mathbb{Z}, k \in \mathbb{N}\}$ and in fact this is one of the advantages of applying subdivision schemes. The computational cost of computing the latter adds an $O(n)$ operations (with n the amount of control points) to the one of generating $\mathbf{P}^k$. However we take profit of that sampling for energies quadrature in applications presented more ahead.

The evaluation of the basic function at dyadic parametric values $\varphi\left(\frac{j}{2^k} - s\right)$ for any $s, j \in \mathbb{Z}$ and $k \in \mathbb{N}$ can be computed as the subdivision of the polygon with vertices $\mathbf{P}^0 = \left\{(j, \delta_j^0),\ j \in \mathbb{Z}\right\}$ by k times (see Fig. 2.4 and 2.5).

The procedure used to compute the sampling $\{c(j/2^k), j \in \mathbb{Z}, k \in \mathbb{N}\}$ is not by subdividing the $\mathbf{P}^k$ towards the limit. Rather we employ (2.2) with a pre-computed sampling $\varphi\left(\frac{j}{2^k} - s\right)$ of the basis function, storing the results in a table. In this way, it is possible not only to compute that particular sampling in dyadic parameters, but also n-adic parameters [106].

Moreover, our approach has the advantage of working with different samplings of the subdivision curve, while the other allows only to work with dyadic parameters.

The values of the basic function can be found from the eigenanalysis of the local subdivision matrix (2.20) [103, 49, 111]. Likewise, they can be computed from the



(a) Values of $\varphi_{\frac{1}{16}}(s),\ s \in \mathbb{Z}$

(b) Values of $\varphi_{\frac{1}{16}}(s),\ s \in \frac{1}{2}\mathbb{Z}$

(c) Values of $\varphi_{\frac{1}{16}}(s),\ s \in \frac{1}{4}\mathbb{Z}$

(d) Values of $\varphi_{\frac{1}{16}}(s),\ s \in \frac{1}{8}\mathbb{Z}$

Figure 2.4: Generating the values of the basic function for the 4-point subdivision scheme.

refinement equation (2.4). Let us illustrate it with two examples, with the quadratic and cubic uniform B-splines as before.

Examples:

For the quadratic uniform B-spline with refinement relation in (2.5) we have

$$\begin{bmatrix} \varphi(-1) \\ \varphi(0) \\ \varphi(1) \\ \varphi(2) \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ \frac{3}{4} & \frac{3}{4} & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & \frac{3}{4} & \frac{3}{4} \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} \varphi(-1) \\ \varphi(0) \\ \varphi(1) \\ \varphi(2) \end{bmatrix}. \tag{2.37}$$

Thus, the values $\{\varphi(-1), \varphi(0), \varphi(1), \varphi(2)\}$ are obtained as solution of an homogeneous equation or has an eigenvector corresponding to the eigenvalue 1 for that matrix, which is, in fact, the transpose of the local subdivision matrix (2.19). The property of partition of unity (2.9) provides the particular solution of interest in the eigenspace.

In this way, we obtain the solution, which is also the first limit stencil

$$\begin{bmatrix} \varphi(-1) \\ \varphi(0) \\ \varphi(1) \\ \varphi(2) \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{bmatrix}. \tag{2.38}$$

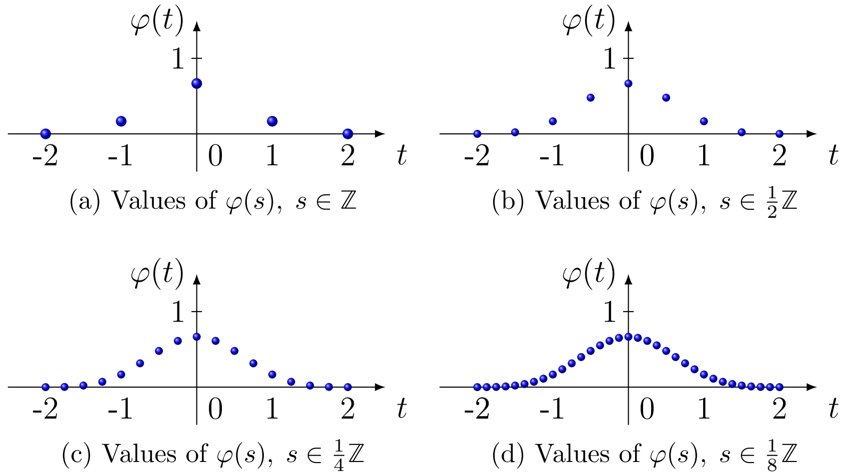


Figure 2.5: Generates the values of the basic function for the cubic B-spline subdivision scheme

On the other hand, for the cubic uniform B-spline from (2.6) we obtain that

$$\begin{bmatrix} \varphi(-2) \\ \varphi(-1) \\ \varphi(0) \\ \varphi(1) \\ \varphi(2) \end{bmatrix} = \begin{bmatrix} \frac{1}{8} & 0 & 0 & 0 & 0 \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{8} & 0 & 0 \\ \frac{1}{8} & \frac{1}{2} & \frac{3}{4} & \frac{1}{2} & \frac{1}{8} \\ 0 & 0 & \frac{1}{8} & \frac{1}{2} & \frac{3}{4} \\ 0 & 0 & 0 & 0 & \frac{1}{8} \end{bmatrix} \begin{bmatrix} \varphi(-2) \\ \varphi(-1) \\ \varphi(0) \\ \varphi(1) \\ \varphi(2) \end{bmatrix}. \tag{2.39}$$

Analogous to the previous case, with the restriction imposed by the partition of unity property, we obtain the solution to the eigenvector problem with respect to the eigenvalue 1, corresponding to the first limit stencil

$$\begin{bmatrix} \varphi(-2) \\ \varphi(-1) \\ \varphi(0) \\ \varphi(1) \\ \varphi(2) \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{6} \\ \frac{2}{3} \\ \frac{1}{6} \\ 0 \end{bmatrix}. \tag{2.40}$$

It is worth noting again that the analyzed matrix is the transpose of the local subdivision matrix (2.18).

$\square$

In general, if the basis function $\varphi(t)$ has support in $[\mu, \nu]$, then we find

$$\begin{bmatrix} \varphi(\mu) \\ \varphi(\mu+1) \\ \vdots \\ \varphi(0) \\ \vdots \\ \varphi(\nu-1) \\ \varphi(\nu) \end{bmatrix} = G_{\mathbf{a}} \begin{bmatrix} \varphi(\mu) \\ \varphi(\mu+1) \\ \vdots \\ \varphi(0) \\ \vdots \\ \varphi(\nu-1) \\ \varphi(\nu) \end{bmatrix} \tag{2.41}$$

with

$$G_{\mathbf{a}} = (a_{2s-j+\mu-1})_{s,j=1}^{\nu-\mu+1} . \tag{2.42}$$

For symmetric masks we have that $G_{\mathbf{a}} = S_{\mathbf{a}}^{\mathrm{T}}$, that is the transpose of the local subdivision matrix in (2.20).

Provided that for a convergent subdivision scheme the local subdivision matrix has dominant eigenvalue 1 [49], we can claim the following.

Lemma 4. *The first limit stencil of a subdivision scheme with symmetric mask* $\mathbf{a} = \{a_\mu, a_{\mu+1}, \ldots, a_\nu\}$ *is the dominant left eigenvector of the local subdivision matrix with sum 1.*

Although derivatives can be numerically approximated with divided differences from the function sampling, it is possible to extend the previous result to these cases. For the derivatives we can obtain an analogous result, as the refinement relation becomes

$$\varphi^{(k)}(t) = \sum_{j\in\mathbb{Z}} 2^k a_j \varphi^{(k)}(2t-j). \tag{2.43}$$

Therefore, in this case the k-th limit stencil is a left eigenvector associated with the eigenvalue $\frac{1}{2^k}$.

The partition of unity property cannot be used in those cases, as it leads to homogeneous system of equations with not unique solutions. Thus, in order to choose a proper vector in the eigenspace associated to the eigenvalue $\frac{1}{2}$, in [106] the following auxiliary constraint is proposed

$$\sum_{j\in\mathbb{Z}} (j-t)\varphi'(t-j) = 1, \tag{2.44}$$

requiring that the first derivative of the linear function t must yield the value 1.

The preceding restriction can be extended for higher order derivatives.

In case of computing a sampling of the subdivision curve in the grid $\frac{1}{n}\mathbb{Z}$, we can also use the refinement equation [106]. For a basic function with $\operatorname{supp}\varphi = [\mu, \nu]$ we have

$$\varphi\left(\frac{r}{n}\right) = \sum_{j\in\mathbb{Z}} a_j \varphi\left(\frac{2r - nj}{n}\right), \qquad r \in \mathbb{Z}, n \in \mathbb{N}\backslash\{0\}. \tag{2.45}$$

Therefore, substituting all the values of $\frac{1}{n}\mathbb{Z} \cap [\mu, \nu]$ in the refinement relation (2.4), we get $(\nu-\mu)n-1$ homogeneous equations or an eigenvalue problem, because the vector of ordered evaluations $\varphi\left(\frac{1}{n}\mathbb{Z} \cap [\mu, \nu]\right)$ is an eigenvector associated to the eigenvalue 1 for the matrix denoted as $G_{\mathbf{a},n}$. More precisely we have

$$G_{\mathbf{a},n} = \left(b_{2s-j+\mu-1}\right)_{s,j=1}^{n(\nu-\mu)+1}, \text{ with } b_j = \begin{cases} a_{\frac{j-\mu}{n}+\mu}, & \text{if } (j-\mu) \mod n = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2.46}$$

It can be verified that $G_{\mathbf{a},1} = G_{\mathbf{a}}$ in (2.42) and each row in (2.46) is obtained from an up-sampling of the mask $\mathbf{a}$ with factor n. Analogous to (2.41), with (2.46) we get the linear relation

$$G_{\mathbf{a},n}\Phi_{\mathbf{a},n} = \Phi_{\mathbf{a},n}, \quad \text{with } \Phi_{\mathbf{a},n} = \left[\varphi(\mu), \varphi\left(\mu + \tfrac{1}{n}\right), \ldots, \varphi(\nu)\right]^{\mathrm{T}}. \tag{2.47}$$

The partition of unity property, as in the previous examples for integer sampling, allows us to choose the proper vector in the eigenspace. In this case we can write it as

$$\left[\begin{array}{c|c} \mathbb{1}_{\nu-\mu} \otimes I_n & \begin{matrix} 1 \\ 0 \\ \vdots \\ 0 \end{matrix} \end{array}\right] \Phi_{\mathbf{a},n} = \mathbb{1}_n^{\mathrm{T}}, \qquad \text{with } \mathbb{1}_n = [11\ldots1] \in \mathbb{R}^{1\times n}. \tag{2.48}$$

Once more, this restriction can be used to find a feasible solution for the eigenvector problem in (2.47) or to define the linear system of equations

$$\begin{bmatrix} G_{\mathbf{a},n} - I_{n(\nu-\mu)+1} & \\ & \begin{array}{c|c} \mathbb{1}_{\nu-\mu} \otimes I_n & \begin{matrix} 1 \\ 0 \\ \vdots \\ 0 \end{matrix} \end{array} \end{bmatrix} \Phi_{\mathbf{a},n} = \begin{bmatrix} \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} n(\nu-\mu)+1 \\ \left.\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}\right\} n \end{bmatrix}. \tag{2.49}$$

Both approaches should lead to the solution, and the choice for one or the other depends on the involved solver. The dimension of both problems depends on n and $\nu - \mu$, which for practical applications are at most 10^3 and 10, respectively.

For the derivatives an analogous procedure is applied.

2.1.4 Linear non-stationary subdivision schemes

A more general form of linear subdivision schemes are the non-stationary ones [49], where the refinement rule (2.12) becomes

$$P_s^{k+1} = \sum_{j \in \mathbb{Z}} a_{s-2j}^k P_j^k, \tag{2.50}$$

with masks $\mathbf{a}_k$ depending on each level of iteration. As in the previous presentation, it can be generalized to higher dilation factor, but we still restrict our discussion to the binary case for the sake of simplicity in the considered examples.

These schemes extend the class of limit curves generated by the stationary ones. Those classes include hyperbolic, trigonometric (and as a consequence conics), and exponential-polynomials curves [4, 22, 97].

When the mask $\mathbf{a}^k = \left\{a_j^k, j \in \mathbb{Z}\right\}$ is independent of the refinement level, i.e., $\mathbf{a}^k = \mathbf{a}$ for all $k \in \mathbb{N}$, we find the stationary case already described. In the non-stationary case, each refinement uses a different subdivision operator $\mathcal{S}_{\mathbf{a}^k}$.

Instead of the procedure in (2.17) for stationary scheme, in this case we have a sequence of basic functions $\{\varphi_k(t)\}$ defined as

$$\varphi_{\mathbf{a}^k} = \lim_{s \to \infty} \left(\prod_{j=0}^{s} \mathcal{S}_{\mathbf{a}^{k+j}} \right) \boldsymbol{\delta}. \tag{2.51}$$

In other words, each basic function $\varphi_{\mathbf{a}^k}$ is obtained starting with the refinement iterations from the mask of k-th level.

The functions in the sequence $\{\varphi_k(t)\}$ in (2.51) are related by a system of *refinement equations* [49]

$$\varphi_{\mathbf{a}^k}(t) = \sum_{j\in\mathbb{Z}} a_j^k \varphi_{\mathbf{a}^{k+1}}(2t-j), \qquad k \in \mathbb{N}. \tag{2.52}$$

In the stationary case this system reduces to (2.4).

If the non-stationary scheme is convergent starting from the 0-the level (i.e., with $\mathbf{a}^0$), then it is convergent starting from any k-th level [46]. Thus, if we consider a fixed starting level k, then the limit of the refinements starting from $\mathbf{P}^0$ is the curve represented in the basis of integer shifts of the basic function $\varphi_{\mathbf{a}^k}(t)$ as

$$c(t) = \sum_{j\in\mathbb{Z}} P_j^0 \varphi_{\mathbf{a}^k}(t-j). \tag{2.53}$$

The support of the latter basic function can be determined by the Minkowski sum of the support of the masks $\mathbf{a}^l$ for $l \geq k$ [49]

$$\operatorname{supp}\varphi_{\mathbf{a}^k} = \sum_{l=k}^{\infty} \frac{2^k}{2^{l+1}} \operatorname{supp}\mathbf{a}_l, \tag{2.54}$$

as an extension of the result for stationary schemes (see *Theorem* 1).

The difference for the analysis of convergence and smoothness with the stationary schemes is subtle. A special technique to analyze the convergence of a non-stationary schemes relies on the identification of a convergent stationary subdivision scheme asymptotically equivalent to it [49, 21, 47, 19, 18]. However, it is not our purpose to study the convergence of subdivision schemes here, rather to use known schemes for solving the interpolation problem.

Finally, we want to highlight the possibility of evaluating the basic function at some parameter, like dyadic for example, for such schemes.

Example:

As an example we can consider the family of non-stationary schemes generalizing the cubic B-spline scheme proposed in [97] with masks

$$\mathbf{a}^k = \left\{\frac{\alpha_k}{8}, \frac{1}{2}, \frac{4-\alpha_k}{4}, \frac{1}{2}, \frac{\alpha_k}{8}\right\}, \qquad \text{with} \quad \alpha_k \in \begin{cases} [0,2), & k=0, \\ (0,2), & k \in \mathbb{N}\backslash\{0\}. \end{cases} \tag{2.55}$$

This scheme generates C^1 subdivision curves if $\lim_{k\to\infty} \alpha_k = \alpha \in (0,2)$, and C^2 subdivision curves if $\sum_{k=0}^{\infty}|\alpha_k - 1| < \infty$. For $\alpha_k = 1$ the cubic uniform B-spline is obtained.

□

For the scheme

$$\begin{cases} P_{2j}^{k+1} &= \frac{\alpha_k}{8} P_{j-1}^k + \frac{4-\alpha_k}{4} P_j^k + \frac{\alpha_k}{8} P_{j+1}^k, \\ P_{2j+1}^k &= \frac{1}{2} P_j^k + \frac{1}{2} P_{j+1}^k, \end{cases} \qquad j \in \mathbb{Z}, k \in \mathbb{N}, \tag{2.56}$$

corresponding to the mask in the example already presented [97], it is possible to provide exact evaluation of the limit curve and its first derivative at integer parameter values. The given result can be extended to dyadic parameter values by considering the initial iteration from another refinement level.

2.2 Toeplitz, circulant and block-circulant matrices

Toeplitz and circulant matrices appear in many areas and applications [25, 75, 90, 88], but for the particular problem we are analyzing, we consider particular cases of them. In this section we also present a few properties regarding block-circulant matrices, as they appear in the modeling of our problem of interest.

Definition 8. *Let* $\boldsymbol{\alpha} = [\alpha_0, \alpha_{-1}, \ldots, \alpha_{-n+1}]$ *with* $\alpha_j \in \mathbb{R}$. *A* circulant matrix $\mathbf{C} = circ(\boldsymbol{\alpha})$ *is defined as satisfying* $\mathbf{C}_{s,t} = \mathbf{C}_{s+1,t+1} = \alpha_{s-t}$, *i.e.,*

$$\mathbf{C} = \begin{bmatrix} \alpha_0 & \alpha_{-1} & \alpha_{-2} & \ldots & \alpha_{-n+1} \\ \alpha_1 & \alpha_0 & \alpha_{-1} & \ddots & \alpha_{-n+2} \\ \vdots & \alpha_1 & \alpha_0 & \ddots & \vdots \\ \alpha_{n-2} & \ddots & \ddots & \ddots & \alpha_{-1} \\ \alpha_{n-1} & \alpha_{n-2} & \ldots & \alpha_1 & \alpha_0 \end{bmatrix} \tag{2.57}$$

with $\alpha_j = \alpha_{j \mod (n)}$ *for all* $j \in \mathbb{Z}$.

This matrix $\mathbf{C} = \text{circ}(\boldsymbol{\alpha})$ can be represented as

$$\mathbf{C} = \sum_{j=0}^{n-1} \Pi_n^j \alpha_j, \tag{2.58}$$

where Π_n is the permutation matrix

$$\Pi_n = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 1 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}. \tag{2.59}$$

From [25] we have the following result.

Lemma 5. *The matrix Π_n can be factorized as*

$$\Pi_n = \mathbf{F}_n \Omega_n \mathbf{F}_n^*, \tag{2.60}$$

where $\mathbf{F}_n$ is the Fourier matrix defined as

$$\mathbf{F}_n = \left[\frac{e^{-2\pi \mathrm{i} jk/n}}{\sqrt{n}} \right]_{j,k=0}^{n-1}, \qquad \mathrm{i}^2 = -1, \tag{2.61}$$

and

$$\Omega_n = diag_{s=1,\dots,n} \left(e^{2\pi \mathrm{i}(s-1)/n} \right). \tag{2.62}$$

As a consequence, $\mathbf{C}$ can be diagonalized as

$$\mathbf{C} = \mathbf{F}_n \mathbf{L}_n \mathbf{F}_n^*, \tag{2.63}$$

where

$$\mathbf{L}_n = \mathrm{diag}_{s=1,\dots,n} \left(\sum_{j=0}^{n-1} e^{2\pi \mathrm{i}(s-1)j/n} \alpha_j \right). \tag{2.64}$$

Note that the diagonal matrix $\mathbf{L}_n$ is defined by the Fourier transform of the first column of $\mathbf{C}$.

From [59, Theorem 6.4] we infer the following result.

Theorem 2. *In the case where $\mathbf{C} = \sum_{j=-\mu}^{\nu} \Pi_n^j \alpha_j$ with fixed $\mu, \nu < \lfloor n/2 \rfloor$, then*

$$\Lambda(\mathbf{C}) = \left\{ f\left(\frac{2\pi j}{n} \right), j = 0, \dots, n-1 \right\}, \tag{2.65}$$

with

$$f(\theta) := \sum_{j=-p}^{q} \alpha_j e^{\mathrm{i}j\theta} \tag{2.66}$$

also called symbol *of* $\mathbf{C}$.

Corollary 1. *The singularity of* $\mathbf{C}$ *depends on whether* $f(\theta)$ *has roots in the grid* $\frac{2\mathbb{N}\pi}{n} \cap [0, 2\pi]$.

As an extension, we can also consider the *block-circulant matrices*, where $\boldsymbol{B}_j \in \mathbb{R}^{k\times k}$ for $j = 0, \ldots, n-1$.

Definition 9. *Let* $\mathcal{A} = [\mathcal{A}_0, \mathcal{A}_{-1}, \ldots, \mathcal{A}_{-n+1}]$ *with* $\mathcal{A}_j \in \mathbb{R}^{d\times d}$, *and consider* $\mathcal{A}_j = \mathcal{A}_{j \bmod (n)}$. *An* $(d \times d)$-block-circulant matrix $\mathcal{C} = circ(\mathcal{A})$, *is defined as satisfying* $\mathcal{C}_{s,t} = \mathcal{C}_{s+1,t+1} = \mathcal{A}_{s-t}$, *i.e.,*

$$\mathcal{C} = \begin{bmatrix} \mathcal{A}_0 & \mathcal{A}_{-1} & \mathcal{A}_{-2} & \ldots & \mathcal{A}_{-n+2} & \mathcal{A}_{-n+1} \\ \mathcal{A}_1 & \mathcal{A}_0 & \mathcal{A}_{-1} & \ddots & \mathcal{A}_{-n+3} & \mathcal{A}_{-n+2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathcal{A}_{n-1} & \mathcal{A}_{n-2} & \mathcal{A}_{n-3} & \ldots & \mathcal{A}_1 & \mathcal{A}_0 \end{bmatrix} = \sum_{j=0}^{n-1} \mathbf{\Pi}_n^j \otimes \mathcal{A}_j. \tag{2.67}$$

Analogously to the previous case, we have the following result.

Lemma 6. *The block-circulant matrix* $\mathcal{C}$ *can be diagonalized by fast Fourier transforms as* $\mathcal{C} = (\mathbf{F}_n \otimes \mathbf{I}_d)\mathcal{L}_n(\mathbf{F}_n^* \otimes \mathbf{I}_d)$, *where* $\mathbf{I}_d$ *is the identity matrix of dimension* $d \in \mathbb{N}$ *and* $\mathcal{L}_n$ *a block-diagonal matrix:*

$$\mathcal{L}_n = \sum_{j=0}^{n-1} \mathbf{\Omega}_n^j \otimes \mathcal{A}_j, \tag{2.68}$$

where

$$\mathcal{L}_n = diag_{s=1,\ldots,n}\left(\sum_{j=0}^{n-1} e^{2j(s-1)\mathrm{i}\pi/n} \mathcal{A}_j\right). \tag{2.69}$$

Proof. We need the following property of the Kronecker product:

- $(\mathbf{AC}) \otimes (\mathbf{BD}) = (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D})$ whenever it makes sense the products $\mathbf{AC}$ and $\mathbf{BD}$.

From (2.67) and (2.60) we have:

$$\mathbf{C} = \sum_{j=0}^{n-1} \mathbf{\Pi}_n^j \otimes \mathcal{A}_j$$

$$
\begin{aligned}
&= \sum_{j=0}^{n-1} \left(\mathbf{F}_n \boldsymbol{\Omega}_n^j \mathbf{F}_n^*\right) \otimes (\mathcal{A}_j \mathbf{I}_d) = \sum_{j=0}^{n-1} \left(\left(\mathbf{F}_n \boldsymbol{\Omega}_n^j\right) \otimes \mathcal{A}_j\right) (\mathbf{F}_n^* \otimes \mathbf{I}_d) \\
&= \sum_{j=0}^{n-1} \left(\left(\mathbf{F}_n \boldsymbol{\Omega}_n^j\right) \otimes (\mathbf{I}_d \mathcal{A}_j)\right) (\mathbf{F}_n^* \otimes \mathbf{I}_d) = \sum_{j=0}^{n-1} (\mathbf{F}_n \otimes \mathbf{I}_d) \left(\boldsymbol{\Omega}_n^j \otimes \mathcal{A}_j\right) (\mathbf{F}_n^* \otimes \mathbf{I}_d) \\
&= (\mathbf{F}_n \otimes \mathbf{I}_d) \left(\sum_{j=0}^{n-1} \boldsymbol{\Omega}_n^j \otimes \mathcal{A}_j\right) (\mathbf{F}_n^* \otimes \mathbf{I}_d) = (\mathbf{F}_n \otimes \mathbf{I}_d) \;\; \mathcal{L}_n \;\; (\mathbf{F}_n^* \otimes \mathbf{I}_d) \,.
\end{aligned}
$$

□

Remark 7. *In the case where* $\mathcal{C} = \sum_{j=-p}^{q} \Pi_n^j \otimes \mathcal{A}_j$ *with fixed* $p, q < \lfloor n/2 \rfloor$*, then*

$$
\Lambda(\mathcal{C}) = \left\{ \lambda_k \left(f \left(\frac{2\pi j}{n} \right) \right), \;\; k = 1, \ldots, m, \;\; j = 0, \ldots, n-1 \right\},
$$

with $f(\theta) := \sum_{j=-p}^{q} \mathcal{A}_j e^{\imath j \theta}$ *an* $m \times m$*-matrix valued function and* $\lambda_k(f(\theta))$, $k = 1, \ldots, m$ *its eigenvalue functions.*

We can consider a more general kind of matrices, known as *Toeplitz matrices*, with the circulant matrices being a particular case.

Definition 10. *A Toeplitz matrix is defined as* $T_n = (\alpha_{j-k})_{j,k=1}^{n}$*, i.e.,*

$$
T_n = \begin{bmatrix}
\alpha_0 & \alpha_{-1} & \alpha_{-2} & \ldots & \alpha_{-n+1} \\
\alpha_1 & \alpha_0 & \alpha_{-1} & \ddots & \alpha_{-n+2} \\
\vdots & \alpha_1 & \alpha_0 & \ddots & \vdots \\
\alpha_{n-2} & \ddots & \ddots & \ddots & \alpha_{-1} \\
\alpha_{n-1} & \alpha_{n-2} & \ldots & \alpha_1 & \alpha_0
\end{bmatrix}. \tag{2.70}
$$

As an extension we can consider the γ-Toeplitz matrices defined as follows.

Definition 11. *A* γ*-Toeplitz matrix is defined as* $T_{n,\gamma} = (\alpha_{j-\gamma k})_{j,k=1}^{n}$*, i.e.,*

$$
T_{n,\gamma} = \begin{bmatrix}
\alpha_0 & \alpha_{-\gamma} & \alpha_{-2\gamma} & \ldots & \alpha_{-(n-1)\gamma} \\
\alpha_1 & \alpha_{1-\gamma} & \alpha_{1-2\gamma} & \ddots & \alpha_{1-(n-1)\gamma} \\
\vdots & \alpha_{2-\gamma} & \alpha_{2-2\gamma} & \ddots & \vdots \\
\alpha_{n-2} & \ddots & \ddots & \ddots & \alpha_{n-2-(n-1)\gamma} \\
\alpha_{n-1} & \alpha_{n-1-\gamma} & \ldots & \alpha_{n-1-(n-2)\gamma} & \alpha_{n-1-(n-1)\gamma}
\end{bmatrix}. \tag{2.71}
$$

From these definitions, we notice that the local subdivision operator in (2.20) as well as the transpose of the matrices defined in (2.42) and (2.46) are 2-Toeplitz matrices.

For the previous class of matrices an eigendecomposition is not possible. However many spectral properties are known [59, 88].

2.2.1 The ω-circulant and the ω-block-circulant matrices

The result about circulant and block-circulant matrices can be extended to ω-circulant matrices (see, e.g., [15, 5, 107] for more details) by considering the perturbed matrix

$$\Pi_{n,\omega} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & \omega \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \tag{2.72}$$

with the permutation matrix Π_n in (2.59) being such that $\Pi_{n,1} = \Pi_n$.

In this setting, we can provide the following definition.

Definition 12. *Let* $\boldsymbol{\alpha} = [\alpha_0, \alpha_{-1}, \dots, \alpha_{-n+1}]$ *with* $\alpha_j \in \mathbb{R}$, *and consider* $\alpha_{-j} = \alpha_{n-j}$. *An* ω-circulant matrix $\mathbf{C}_\omega = circ_\omega(\boldsymbol{\alpha})$, *is defined as satisfying*

$$(\mathbf{C}_\omega)_{s,t} = \begin{cases} \alpha_{s-t}, & \text{if } s > t, \\ \omega\alpha_{s-t}, & \text{if } s \le t. \end{cases}$$

We can notice that the case $\omega = 1$ provides the original circulant matrices.

Analogously to the original case, the matrix $\mathbf{C}_\omega$ can be represented as

$$\mathbf{C}_\omega = \sum_{j=0}^{n-1} \Pi_{n,\omega}^j \alpha_j, \tag{2.73}$$

Let us write $\omega = \rho e^{\mathrm{i}\psi}$ with $\rho > 0$ and consider $\sqrt[n]{\omega} = \sqrt[n]{\rho}\, e^{\mathrm{i}\frac{\psi}{n}}$. Then, thanks to (2.60), the matrix $\Pi_{n,\omega}$ can be factorized as

$$\Pi_{n,\omega} = \sqrt[n]{\omega} D_\omega \Pi_n D_\omega^{-1} = \sqrt[n]{\omega} D_\omega F_n \Omega_n F_n^* D_\omega^{-1} \tag{2.74}$$

with $D_\omega = \mathrm{diag}_{s=1,\ldots,n}\left(\omega^{-\frac{s-1}{n}}\right)$ and Ω_n as in (2.62). Note that this is equivalent to write $\Pi_{n,\omega} = \sqrt[n]{|\omega|} D_{n,\omega} F_n \Omega_{n,\omega} F_n^* D_\omega^{-1}$ with

$$\Omega_{n,\omega} = \mathrm{diag}_{s=1,\ldots,n}\left(e^{\mathrm{i}(2\pi(s-1)+\psi)/n}\right)$$

(compare with (2.62) for a better understanding of the role of ω).

Let us define $\mathbf{F}_{n,\omega} = D_\omega \mathbf{F}_n$. As it is expected, for $\omega = 1$ we get $\mathbf{F}_{n,1} = \mathbf{F}_n$. Then, we get the factorization $\mathbf{C}_\omega = \mathbf{F}_{n,\omega}\mathbf{L}_{n,\omega}\mathbf{F}_{n,\omega}^{-1}$ where

$$\mathbf{L}_{n,\omega} = \mathrm{diag}_{s=1,\ldots,n}\left(\sum_{j=0}^{n-1} \omega^{j/n} e^{2\pi\mathrm{i}(s-1)j/n}\alpha_j\right). \tag{2.75}$$

Remark 8. *In the case where* $\mathbf{C}_\omega = \sum_{j=-\mu}^{\nu} \Pi_{n,\omega}^j \alpha_j$ *with fixed* $\mu, \nu < \lfloor n/2 \rfloor$, *and* $\omega = e^{\mathrm{i}\psi}$, *then*

$$\Lambda(\mathbf{C}_\omega) = \left\{ f\left(\frac{2\pi j + \psi}{n}\right), j = 0, \ldots, n-1 \right\},$$

with $f(\theta) := \sum_{j=-\mu}^{\nu} \alpha_j e^{\mathrm{i}j\theta}$.

The extension to ω-block-circulant matrices is similar to the block-circulant case in (2.67), considering in this case the tensor products $\Pi_{n,\omega}^j \otimes \mathcal{A}_j$. As expected, when $\omega = 1$ the latter case reduces to $(d \times d)$-block-circulant matrices.

Also in the block case, ω-circulants can be diagonalized by fast Fourier transforms as follows

$$\begin{aligned}\mathcal{C}_\omega = \sum_{j=0}^{n-1} \Pi_{n,\omega}^j \otimes \mathcal{A}_j &= (D_\omega \otimes I_d)(F_{n,1} \otimes I_d)\mathcal{L}_{n,\omega}(F_{n,1}^* \otimes I_d)(D_\omega^{-1} \otimes I_d) \\ &= (F_{n,\omega} \otimes I_d)\mathcal{L}_{n,\omega}(F_{n,\omega}^{-1} \otimes I_d)\end{aligned}$$

where

$$\mathcal{L}_{n,\omega} = \mathrm{diag}_{s=1,\ldots,n}\left(\sum_{j=0}^{n-1} \omega^{j/n} e^{2\pi\mathrm{i}(s-1)j/n}\mathcal{A}_j\right), \tag{2.76}$$

and I_d is the identity of size d.

Remark 9. *In the case where* $\mathcal{C}_\omega = \sum_{j=-\mu}^{\nu} \Pi_{n,\omega}^j \otimes \mathcal{A}_j$ *with fixed* $\mu, \nu < \lfloor n/2 \rfloor$, *and* $\omega = e^{\mathrm{i}\psi}$ *then*

$$\Lambda(\mathcal{C}_\omega) = \left\{ \lambda_k\left(f\left(\frac{2\pi j + \psi}{n}\right)\right),\ k = 1, \ldots, d,\ j = 0, \ldots, n-1 \right\},$$

with $f(\theta) := \sum_{j=-\mu}^{\nu} \mathcal{A}_j e^{\imath j\theta}$ *an* $d \times d$*-matrix valued function and* $\lambda_k(f(\theta))$*,* $k = 1, \ldots, d$ *its eigenvalue functions.*

2.3 Energy functionals

Besides the interpolatory condition of a subdivision curve, we should check the *fairness* [56] of the curve when there are so many possible curves, if not infinite, that satisfy the considered requirement. In the last setting, the *fairest* one is chosen. The fairness may be seen as a subjective criteria, based on how pleasant the curve is for certain applications and the absence of artifacts. There are many proposals referring to how to measure it, based on physical or geometrical properties of the curve that are described by energies functionals [55].

Those functionals are described originally based on arc-length parameterization of the curve. However, in practice for geometrical algorithms only straight lines admit arc-length parameterization by rational functions [105]. Thus, to overcome the computationally expensive cost of evaluating such functionals, approximations based on other parameterizations are needed. A few contributions to the literature can be found in [72, 1, 116, 62].

Let us consider a curve $c(t)$ and the associated curvature $\kappa(t)$ and torsion $\tau(t)$ functions [38]. The following functionals (see [116]):

$$\mathcal{E}_{stretch}(c) := \int_{\mathbb{R}} \|c'(t)\| \, \mathrm{d}t,$$

$$\mathcal{E}_{bend}(c) := \int_{\mathbb{R}} \kappa(t)^2 \, \|c'(t)\| \, \mathrm{d}t,$$

$$\mathcal{E}_{twist}(c) := \int_{\mathbb{R}} \tau(t)^2 \, \|c'(t)\| \, \mathrm{d}t,$$

relate the variation of tangent, curvature, and torsion along the curve, respectively. The first energy controls curve length stretching and its minimization restricts the occurrence of loops or other undesired effects of the curve length. The second energy indicates the global curve bending. The latter is combined with the first as in some cases the minimization of bending can introduce large loops [116]. The minimization of the third energy, (i.e., the twisting energy,) prevents the curve from coming out of the osculating plane at each point. According to this criterion the curve is locally as close to a plane curve as possible. For many applications it is desired that those values are not excessively large, since too much stretch implies the presence of loops, while the bending and twisting reflects unpleasant shapes.

Remark 10. *We can notice that for plane curves the torsion $\tau(t)$ is zero at every point, which implies the omission of the twisting energy for the fairness measurement.*

As the fairness reflects the *beauty* of the curve, we require low values of those functionals or the combined expressions obtained from them.

The previous energies considered are non-linear, as a consequence their optimizations lead to a non-linear and non-convex problem. In order to solve a convex problem, these energies can be approximated [1, 116, 72] as

$$\widetilde{\mathcal{E}}_{stretch}(c) := \int_{\mathbb{R}} \|c'(t)\|^2 \,\mathrm{d}t, \tag{2.77}$$

$$\widetilde{\mathcal{E}}_{bend}(c) := \int_{\mathbb{R}} \|c''(t)\|^2 \,\mathrm{d}t, \tag{2.78}$$

$$\widetilde{\mathcal{E}}_{twist}(c) := \int_{\mathbb{R}} \|c'''(t)\|^2 \,\mathrm{d}t, \tag{2.79}$$

and for modelling purposes one can choose to minimize a linear combination of these functionals, i.e.

$$\mathcal{E}(c) = \alpha_1 \,\widetilde{\mathcal{E}}_{stretch}(c) + \alpha_2 \,\widetilde{\mathcal{E}}_{bend}(c) + \alpha_3 \,\widetilde{\mathcal{E}}_{twist}(c), \tag{2.80}$$

for some $\alpha_k \in \mathbb{R}$, $k \in \{1,2,3\}$. If we normalize these energies, we could consider a convex combination with $\alpha_k \in [0,1]$, $k \in \{1,2,3\}$, and $\alpha_1 + \alpha_2 + \alpha_3 = 1$.

The values of the functionals in (2.77), (2.78), and (2.79) are not in the same range. A normalization to make them lie in the interval $[0,1]$ is not trivial. Therefore, the choices for each α_k in (2.80) have to care of that fact.

As a result of the previous considerations, the optimization problem obtained with respect to the control points is a quadratic problem.

Let $c(t)$ be a closed curve defined as in (2.2). To minimize $\mathcal{E}(c)$, we start by considering

$$\widetilde{\mathcal{E}}_{stretch}(c) = \int_{\mathbb{R}} \left\| \sum_{j\in\mathbb{Z}} (P_j^0 - P_{j-1}^0)\varphi^{[1]}(t-j) \right\|^2 \mathrm{d}t$$

$$
\begin{aligned}
&= \int_{\mathbb{R}} \sum_{\ell=1}^{m} \left\| \sum_{j\in\mathbb{Z}} (P_j^0 - P_{j-1}^0)(\ell)\varphi^{[1]}(t-j) \right\|^2 \mathrm{d}t \\
&= \sum_{\ell=1}^{m} \mathbf{P}^0(:,\ell)^\top \mathbf{D}_1^\top \mathbf{G}_1 \mathbf{D}_1 \mathbf{P}^0(:,\ell), \qquad (2.81)
\end{aligned}
$$

where D_1, $D_1^\top$ are circulant matrices with

$$
\mathbf{D}_1 = \begin{bmatrix} 1 & 0 & \dots & 0 & -1 \\ -1 & 1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & \dots & -1 & 1 \end{bmatrix} \qquad (2.82)
$$

and

$$
\mathbf{G}_1 \; := \; \left[\int_{\mathbb{R}} \varphi^{[1]}(t-j)\; \varphi^{[1]}(t-k)\; dt \right]_{j,k=1}^{n}. \qquad (2.83)
$$

Since $\varphi^{[1]}$ is a refinable function, the elements of $\mathbf{G}_1$ can be computed as in [76]. In a similar way one can treat the other energies.

For the second energy we find

$$
\begin{aligned}
\widetilde{\mathcal{E}}_{bend}(c) &= \int_{\mathbb{R}} \|c''(t)\|^2 \mathrm{d}t \qquad (2.84) \\
&= \int_{\mathbb{R}} \left\| \sum_{j\in\mathbb{Z}} (P_{j+1}^0 - 2P_j^0 + P_{j-1}^0)\varphi^{[2]}(t-j) \right\|^2 \mathrm{d}t \\
&= \int_{\mathbb{R}} \sum_{\ell=1}^{m} \left\| \sum_{j\in\mathbb{Z}} (P_{j+1}^0 - 2P_j^0 + P_{j-1}^0)(\ell)\varphi^{[2]}(t-j) \right\|^2 \mathrm{d}t \\
&= \sum_{\ell=1}^{m} \mathbf{P}^0(:,\ell)^\top \mathbf{D}_2^\top \mathbf{G}_2 \mathbf{D}_2 \mathbf{P}^0(:,\ell). \qquad (2.85)
\end{aligned}
$$

Analogously, the twisting energies can be computed as

$$
\widetilde{\mathcal{E}}_{twist}(c) = \sum_{\ell=1}^{m} \; \mathbf{P}^0(:,\ell)^\top \; \mathbf{D}_3^\top \; \mathbf{G}_3 \; \mathbf{D}_3 \; \mathbf{P}^0(:,\ell). \qquad (2.86)
$$

The matrices D_2, D_3, and their transposes are circulant matrices defined as

$$\mathbf{D}_2 := \begin{bmatrix} -2 & 1 & 0 & \dots & 0 & 1 \\ 1 & -2 & 1 & \dots & 0 & 0 \\ 0 & 1 & -2 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & -2 & 1 \\ 1 & 0 & 0 & \dots & 1 & -2 \end{bmatrix} \tag{2.87}$$

and

$$\mathbf{D}_3 := \begin{bmatrix} -3 & 3 & -1 & 0 & \dots & 0 & 1 \\ 1 & -3 & 3 & -1 & \dots & 0 & 0 \\ 0 & 1 & -3 & 3 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ -1 & 0 & 0 & 0 & \ddots & -3 & 3 \\ 3 & -1 & 0 & 0 & \dots & 1 & -3 \end{bmatrix}. \tag{2.88}$$

Moreover

$$\mathbf{G}_r := \left[\int_{\mathbb{R}} \varphi^{[r]}(t-j)\varphi^{[r]}(t-k)\mathrm{d}t \right]_{j,k=1}^{n}, \tag{2.89}$$

for $r = 2, 3$.

Finally, we obtain the expression for the fairness energy

$$\mathcal{E}(c) = \sum_{\ell=1}^{m} \mathbf{P}^0(:,\ell)^\top \, \left(\alpha_1 \mathbf{D}_1^\top \mathbf{G}_1 \mathbf{D}_1 + \alpha_2 \mathbf{D}_2^\top \mathbf{G}_2 \mathbf{D}_2 + \alpha_3 \mathbf{D}_3^\top \, \mathbf{G}_3 \, \mathbf{D}_3\right) \, \mathbf{P}^0(:,\ell). \tag{2.90}$$

Remark 11. *The matrices $\mathbf{G}_r$ depend only on the basic function and they are independent of the control polygon. Hence, those matrices can be pre-computed in the implementation once the subdivision scheme is chosen.*

Chapter 3

Interpolating with scalar subdivision schemes

Interpolatory subdivision schemes provide a direct solution to the interpolation of data points [50, 33, 34]. Similarly, Hermite subdivision schemes generate a curve interpolating given points and associated derivatives [84, 73]. However, in the first case they do not generate curves smoother than the ones obtained with approximating schemes. On the other hand, Hermite schemes usually generate spline curves that locally have higher continuity degree than in the interpolation points.

Linear scalar subdivision schemes has been extensively studied in the literature [118, 102, 42, 103, 14, 18]. Although there are proposed solutions for the interpolation with approximating schemes, both for univariate and bivariate cases, some gaps need to be covered. The interpolation with dual approximating subdivision schemes, the treatment of special cases for family of schemes as the J-splines [101], and the interpolation of derivatives, are some examples those gaps.

This chapter examines the interpolation problem. The first section addresses only points while the second section covers associated derivatives. When the interpolation operator in (2.27) is singular, we consider three possible approaches. This aspect is covered in a follow-up section. Those alternatives to solving the problem with a singular operator are the perturbation of the spectrum of the interpolation operator as a regularization problem, the perturbation of the least square solution to maximize a fairness functional, and the shift of the parameters where the interpolation occurs. The latter provides a non-singular operator making the problem easier to solve. The final section compares the strategies proposed with existing methods [89, 101].

3.1 Point interpolation problem

Let M_n be the matrix representing the point interpolation operator for linear uniform stationary subdivision schemes in (2.27), in the sense that $M_n\mathbf{P}^0 = \mathbf{V}$, with $\mathbf{V}$ the vector with entries $\{V_i = c(i), i = 0, \ldots, n-1\}$. Our goal is to compute the n control points $\mathbf{P}^0 = \{P_i^0, i = 0, \ldots, n-1\}$ which define the subdivision curve $c(t)$. Then, the matrix M_n, that we analyze in this section, is a circulant matrix because of the imposed periodicity.

Let $\boldsymbol{\beta}^0 = \{\beta_{-p}, \ldots, \beta_{-1}, \beta_0, \beta_1, \ldots, \beta_q\}$ be the first limit stencil of a subdivision scheme, with $p = q$ for primal schemes and $q = p-1$ for the dual case. Let us also consider the symmetry conditions $\beta_{-j} = \beta_j$ and $\beta_{1-j} = \beta_j$ ($j \in \mathbb{N}$) for primal and dual schemes, respectively.

In [101] the authors proposed[1] a one-parameter family of subdivision schemes, blending two well-known subdivision schemes, the cubic B-spline and the 4-point subdivision schemes, where the first is an approximating scheme, and the second is an interpolatory one. One advantage of this family is that it can produce limit curves with higher smoothness than C^2, unlike the cubic B-spline or the 4-point scheme. These schemes are called *J-splines* and their subdivision rules are reported below

$$\begin{cases} P_{2j}^{k+1} &= \frac{\nu}{8}P_{j-1}^k + \frac{8-2\nu}{8}P_j^k + \frac{\nu}{8}P_{j+1}^k, \\ P_{2j+1}^{k+1} &= \frac{\nu-1}{16}P_{j-1}^k + \frac{9-\nu}{16}P_j^k + \frac{9-\nu}{16}P_{j+1}^k + \frac{\nu-1}{16}P_{j+2}^k, \end{cases} \quad j \in \mathbb{Z}, k \in \mathbb{N}. \tag{3.1}$$

This family includes many known schemes. For $\nu = 0$ the 4-point subdivision scheme [33] is obtained, if $\nu = \frac{1}{2}$ we find the scheme proposed in [100] and, when $\nu = 1$, (3.1) reduce to the uniform cubic B-spline [77]. In general, those rules produce curves that are at least C^1 for $\nu \in [-1.7, 5.8]$, C^2 for $\nu \in (0, 4)$, C^3 for $\nu \in (1, 2.8]$, and C^4 for $\nu = 3/2$, which corresponds to the uniform quintic B-spline subdivision scheme.

The elements in the support of the limit stencils are the following

$$\boldsymbol{\beta}^0 = \frac{1}{12(6+\nu)}\left\{(\nu–1)\nu,\ 2\nu(8–\nu),\ 72 + 2(\nu–9)\nu,\ 2\nu(8–\nu),\ (\nu–1)\nu\right\}, \tag{3.2}$$

$$\boldsymbol{\beta}^1 = \frac{1}{12}\left\{1–\nu,\ 2(\nu–4),\ 0,\ –2(\nu–4),\ –(1–\nu)\right\}, \tag{3.3}$$

$$\boldsymbol{\beta}^2 = \frac{1}{2\nu}\left\{\nu-1,\ 2(2-\nu),\ 2(\nu-3),\ 2(2-\nu),\ \nu-1\right\}, \tag{3.4}$$

[1] Originally proposed by Maillot and Stam, 2001.

$$\boldsymbol{\beta}^3 = \frac{1}{2}\left\{1,\ -2,\ 0,\ 2,\ -1\right\}. \tag{3.5}$$

Let us consider as example for a primal scheme the J-spline scheme with its first limit stencil (3.2). For the interpolation of 6 and 7 points we obtain the matrices

$$M_6 = \begin{bmatrix}
\frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} & 0 & \frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} \\
\frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} & 0 & \frac{(\nu-1)\nu}{12(6+\nu)} \\
\frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} & 0 \\
0 & \frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} \\
\frac{(\nu-1)\nu}{12(6+\nu)} & 0 & \frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} \\
\frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} & 0 & \frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)}
\end{bmatrix} \tag{3.6}$$

and

$$M_7 = \begin{bmatrix}
\frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} & 0 & 0 & \frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} \\
\frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} & 0 & 0 & \frac{(\nu-1)\nu}{12(6+\nu)} \\
\frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} & 0 & 0 \\
0 & \frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} & 0 \\
0 & 0 & \frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} \\
\frac{(\nu-1)\nu}{12(6+\nu)} & 0 & 0 & \frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} \\
\frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{(\nu-1)\nu}{12(6+\nu)} & 0 & 0 & \frac{(\nu-1)\nu}{12(6+\nu)} & \frac{2\nu(8-\nu)}{12(6+\nu)} & \frac{72+2(\nu-9)\nu}{12(6+\nu)}
\end{bmatrix}, \tag{3.7}$$

respectively. In this case the matrix M_n depends on the parameter ν.

The symbol in (2.66) can be translated into the polynomial

$$p(z) = z^p \sum_{k=-p}^{q} \beta_k z^k, \tag{3.8}$$

such that, from (2.65), we deduce that M_n is singular if and only if $p(z)$ vanishes at any root of unity of order n.

For the primal schemes $\mathbf{b}_n := \left[\beta_0, \beta_{-1}, \ldots, \beta_{-p}, \mathbf{0}_{1\times(n-2p-1)}, \beta_p, \ldots, \beta_1\right] \in \mathbb{R}^n$ and we find

$$b(\theta) = \sum_{k=-p}^{p} \beta_k e^{k\mathrm{i}\theta} = \beta_0 + 2\sum_{k=1}^{p} \beta_k \cos(k\theta). \tag{3.9}$$

On the other hand, for the dual schemes $\mathbf{b}_n := \left[\beta_0, \beta_{-1}, \ldots, \beta_{-p+1}, \mathbf{0}_{1\times(n-2p)}, \beta_p, \ldots, \beta_1\right] \in$

$\mathbb{R}^n$, we have

$$b(\theta) = \sum_{k=-p}^{q} \beta_k e^{k\mathrm{i}\theta} = \sum_{k=1}^{p} \beta_k \left(e^{(-k+1)\mathrm{i}\theta} + e^{k\mathrm{i}\theta}\right) = e^{\frac{\mathrm{i}\theta}{2}} \sum_{k=1}^{p} \beta_k \left(e^{\left(-k+\frac{1}{2}\right)\mathrm{i}\theta} + e^{\left(k-\frac{1}{2}\right)\mathrm{i}\theta}\right)$$
$$= 2e^{\frac{\mathrm{i}\theta}{2}} \sum_{k=1}^{p} \beta_k \cos\left((2k-1)\tfrac{\theta}{2}\right). \tag{3.10}$$

In both cases, the first limit stencil satisfies $b(0) = 1$.

From (3.10) we get that for even-symmetric schemes $b(\pi) = 0$ independently of the values of n and $\boldsymbol{\beta}$. As π belong to the grid $\frac{2\mathbb{N}\pi}{n} \cap [0, 2\pi]$ for even n, then by (2.65) we conclude the following.

Lemma 7. *For any even-symmetric subdivision scheme, if the amount of interpolated points n is even, then the interpolation matrix M_n is singular.*

We notice that for primal schemes:

$$b(2\pi - \theta) = \beta_0 + 2\sum_{k=1}^{p} \beta_k \cos(k(2\pi - \theta)) = \beta_0 + 2\sum_{k=1}^{p} \beta_k \cos(k\theta) = b(\theta) \tag{3.11}$$

while for the dual cases:

$$b(2\pi - \theta) = 2e^{\frac{\mathrm{i}(2\pi-\theta)}{2}} \sum_{k=0}^{p} \beta_k \cos\left((2k-1)\frac{(2\pi-\theta)}{2}\right) = 2e^{\frac{-\mathrm{i}\theta}{2}} \sum_{k=0}^{p} \beta_k \cos\left((2k-1)\tfrac{\theta}{2}\right)$$
$$= e^{-\mathrm{i}\theta} b(\theta). \tag{3.12}$$

Therefore, the study of the symbol in the interval $[0, \pi]$ give us information also in the interval $[0, 2\pi]$. In particular, for every symbol (3.9) and (3.10) we deduce that $b(\theta) = 0$ if and only if $b(2\pi - \theta) = 0$.

The symbols (3.9) and (3.10) can be rewritten by using the Chebyshev polynomials of first kind $T_n(\cos(\theta)) = \cos(n\theta)$, so that in each respective case from (3.9) we have

$$b(\theta) = \beta_0 + 2\sum_{k=1}^{p} \beta_k \, T_k(\cos(\theta)) \tag{3.13}$$

and from (3.10)

$$b(\theta) = 2e^{\frac{\mathrm{i}\theta}{2}} \sum_{k=1}^{p} \beta_k \, T_{2k-1}\left(\cos\left(\tfrac{\theta}{2}\right)\right). \tag{3.14}$$

The preceding representations provide some insight concerning the nature of the symbols namely, the amount of roots and useful bounds for the norms.

Lemma 8. *The symbol* $b(\theta)$ *in* (3.13)*, corresponding to a primal subdivision scheme, has at most* $2p$ *roots in the interval* $[0, 2\pi]$.

Proof. Let us define:

$$c(z) := \beta_0 + 2\sum_{k=1}^{p} \beta_k \, T_k(z), \tag{3.15}$$

then $b(\theta) = c(\cos(\theta))$.

If θ_0 is a root of $b(\theta)$, then $z_0 = \cos(\theta_0)$ is a root of $c(z)$. On the other hand, if there exists a root $z_0 \in [-1, 1]$ of $c(z)$, then $\theta_0 = \arccos(z_0) \in [0, \pi]$ is a root of $b(\theta)$. By (3.11), $2\pi - \theta_0 \in [\pi, 2\pi]$ is also a root of $b(\theta)$.

The polynomial $c(z)$ has degree p, because that is the higher degree of the Chebyshev polynomials in (3.15). Then $c(z)$ has at most p real roots in $[-1, 1]$. Thus, the symbols have at most $2p$ roots in the interval $[0, 2\pi]$. □

Lemma 9. *The symbol* $b(\theta)$ *in* (3.14)*, corresponding to a dual subdivision scheme, has at most* $4p + 1$ *roots in the interval* $[0, 2\pi]$*, being* π *one of them.*

Proof. As in the previous lemma, let us define:

$$c(z) := \sum_{k=1}^{p} \beta_k \, T_{2k-1}\left(z\right), \tag{3.16}$$

then $b(\theta) = 2e^{\frac{i\theta}{2}} c\left(\cos\left(\frac{\theta}{2}\right)\right)$.

Analogously to the previous Lemma, if θ_0 is a root of $b(\theta)$, then $z_0 = \cos(\frac{\theta_0}{2})$ is a root of $c(z)$. On the other hand, if there exists a root $z_0 \in [-1, 1]$ of $c(z)$, then $\theta_0 = 2\arccos(z_0) \in [0, 2\pi]$ is a root of $b(\theta)$. Because of (3.12), $2\pi - \theta_0$ is also a root of $b(\theta)$.

As $T_{2k+1}(0) = 0$ for every $k \in \mathbb{N}$, we get that $\pi = 2\arccos(0)$ is always a root of $b(\theta)$.

In this case, the polynomial $c(z)/z$ has degree $2p$ and therefore it has at most $2p$ roots in $[-1, 1]$. Thus, $b(\theta)$ has at most $4p+1$ roots in the interval $[0, 2\pi]$, by counting π at least once.

□

The analysis of the considered polynomials is done in the Chebyshev polynomial basis, instead of the standard monomial basis. Also the study of the roots can be done in that basis by using the *colleague matrix* [114]:

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \ddots & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ddots & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{2} & 0 \end{bmatrix} - \frac{1}{4\beta_p} \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_0 & 2\beta_1 & \dots & 2\beta_{p-1} \end{bmatrix} \tag{3.17}$$

of dimension $p \times p$ for the primal case (3.15), and

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \ddots & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ddots & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{2} & 0 \end{bmatrix} - \frac{1}{2\beta_p} \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \beta_1 & 0 & \beta_2 & \dots & \beta_{p-1} & 0 \end{bmatrix} \tag{3.18}$$

of dimension $(2p-1) \times (2p-1)$ for the dual case (3.16). In particular, the roots of each polynomial are equal to the eigenvalues of the respective colleague matrices. This approach is more suitable for these polynomial basis, while for the monomial basis it is used the *companion matrix* to find its eigenvalues which are also equal to the roots of the polynomial [114].

Remark 12. *The proofs of the previous lemmas provide necessary conditions for the singularity of matrix M_n by the analysis of the roots of the classes of polynomials in* (3.15) *and* (3.16)*. In this setting, if $c(t)$ has a root of type $\cos(\frac{2j\pi}{n})$ or $\cos(\frac{j\pi}{n})$, for each respective case, with $j = 0, \dots, n-1$, then M_n happens to be singular.*

Examples:

As an example of a primal scheme, for both M_6 and M_7 in (3.6) and (3.7), by using (3.9) we obtain the same symbol

$$b(\theta) = \tfrac{72+2(\nu-9)\nu}{12(6+\nu)} + \tfrac{4\nu(8-\nu)}{12(6+\nu)} \cos(\theta) + \tfrac{2(\nu-1)\nu}{12(6+\nu)} \cos(2\theta). \tag{3.19}$$

Then, for (3.6) and (3.7) we have the spectrum

$$\Lambda(M_6) = \left\{b\left(\tfrac{2j\pi}{6}\right), j=0,\dots,5\right\} \quad \text{and} \quad \Lambda(M_7) = \left\{b\left(\tfrac{2j\pi}{7}\right), j=0,\dots,6\right\}, \tag{3.20}$$

respectively.

We can notice that $b(\pi) = \frac{2\nu^2-13\nu+18}{3(6+\nu)}$ and then for $\nu = 2$ or $\nu = 4.5$ we deduce that $b(\pi) = 0$. Therefore we have a zero value in the spectrum of M_{2n} for those parameter values as $b(\pi) = b(\frac{2j\pi}{2n})$ for $j = n$, and therefore the matrix M_{2n} is singular.

We can verify[2] that $b(\theta)$ only vanishes at real solution pairs

$$\{\nu = 2, \theta = \pi\}, \{\nu = 9/2, \theta = \pi/2\}, \{\nu = 9/2, \theta = \pi\}, \text{ and } \{\nu = 9/2, \theta = 3\pi/2\}.$$

On the other hand, as an example of a dual scheme, let us consider the family of schemes proposed by Dyn, Floater, Hormann in [43] with subdivision rules

$$\begin{cases} P^{k+1}_{2j} &= -\frac{7\nu}{8}P^k_{j-1} + \frac{6+9\nu}{8}P^k_j + \frac{2+3\nu}{8}P^k_{j+1} - \frac{5\nu}{8}P^k_{j+2}, \\ P^{k+1}_{2j+1} &= -\frac{5\nu}{8}P^k_{j-1} + \frac{2+3\nu}{8}P^k_j + \frac{6+9\nu}{8}P^k_{j+1} - \frac{7\nu}{8}P^k_{j+2}, \end{cases} \quad j \in \mathbb{Z}, k \in \mathbb{N}. \tag{3.21}$$

For $\nu \in \left(0, \frac{1}{6}\right]$ the scheme generates curves with C^2 continuity and for $\nu = 0$ the generated C^1 curves are the quadratic B-splines generated by the Chaikin algorithm [11].

The first limit stencil of these schemes is

$$\boldsymbol{\beta}^0 = \left\{\tfrac{5\nu^2}{8(1-\nu)}, -\tfrac{(7\nu+8)\nu}{8(1-\nu)}, \tfrac{\nu^2+2\nu+2}{4(1-\nu)}, \tfrac{\nu^2+2\nu+2}{4(1-\nu)}, -\tfrac{(7\nu+8)\nu}{8(1-\nu)}, \tfrac{5\nu^2}{8(1-\nu)}\right\}. \tag{3.22}$$

Then, by (3.10) we obtain the symbol

$$b(\theta) = 2e^{\frac{i\theta}{2}}\left(\tfrac{\nu^2+2\nu+2}{4(1-\nu)}\cos\left(\tfrac{\theta}{2}\right) - \tfrac{(7\nu+8)\nu}{8(1-\nu)}\cos\left(\tfrac{3\theta}{2}\right) + \tfrac{5\nu^2}{8(1-\nu)}\cos\left(\tfrac{5\theta}{2}\right)\right). \tag{3.23}$$

As a consequence, it is straightforward to see that $b(\pi) = 0$ for all $\nu \in \mathbb{R}$, and so every matrix M_{2n} is singular for all $\nu \in \mathbb{R}$, as pointed out in *Lemma 7*.

$\square$

We can ask two questions:

[2] We provide the tools to check later for the general case in *Table 3.1*.

1. *Is matrix M_n singular?* This is equivalent of asking: *when the trigonometric polynomial $b(\theta)$ vanishes in the grid $\frac{2\pi}{n}\mathbb{Z} \cap [0, \pi]$?*

2. *If the matrix M_n is singular, is the matrix M_{n+1} singular too?*

The answer to the second question may require that we take into consideration a different number of control points to avoid the singular case.

To answer the first question we use the Chebyshev series to determine the roots of $b(\theta)$. For polynomials of degree up to 4, we could immediately use the well-known formulas to find their roots. That would provide sufficient conditions for the first limit stencils to check whether the matrix M_n is singular. Instead, we choose an analytic strategy (see Algorithm 1) that allows for weaker necessary conditions and might be used in the general case for a polynomial of degree larger than 4. Such a strategy consists of dividing the process of root-finding in three steps. First to check if there exists some roots (*necessary conditions*), second to locate the intervals where those roots lie, and third, to verify if the roots are in the analyzed grid (*sufficient conditions*).

Algorithm 1 Check if $b(\theta)$ vanishes in the grid $\frac{2\pi}{n}\mathbb{Z} \cap [0, \pi]$

Input: subdivision mask $\mathbf{a}$, n

Output: boolean value for singularity

```
1: Compute first limit stencils β^0 from mask a
2: if a is odd-symmetric then
3:     c(z) = β_0 + 2 Σ_{k=1}^{p} β_k T_k(z)
4: else if a is even-symmetric then
5:     c(z) = Σ_{k=1}^{p} β_k T_{2k-1}(z)
6: end if
7: Find roots z̃_j of c'(z) in [−1, 1]
8: Check c(z̃_j) c(z̃_{j+1}) ≤ 0
9: Check if the root in [z̃_j, z̃_{j+1}] is contained in {cos(2πN/n)}
10: return boolean value for singularity
```

Those roots can be found by the root finding algorithm [6] and then confirmed they match with the Chebyshev points in the grids $\frac{2\pi}{n}\mathbb{Z} \cap [0, \pi]$ for suitable n, as n

is the amount of control points provided by the user. The amount is limited by the application.

For the lower degrees of the polynomial $c(z)$ the previous Algorithm 1 can be simplified into conditions based on the first limit stencil values. What follows is an examination of odd-symmetric and even-symmetric cases.

3.1.1 Primal subdivision schemes

Let us analyze first the primal case, that is stencils of the form

$$\boldsymbol{\beta}^0 = \{\beta_p, \dots, \beta_1, \beta_0, \beta_1, \dots, \beta_p\},$$

for the particular cases of $p = 1$ and $p = 2$.

First degree trigonometric polynomials

If the first limit stencil is of the form $\boldsymbol{\beta}^0 = \{\beta_1, \beta_0, \beta_1\}$ with $\beta_1 \neq 0$, then $\mathbf{b}_n = [\beta_0, \beta_1, \mathbf{0}_{1\times(n-3)}, \beta_1] \in \mathbb{R}^n$ and

$$b(\theta) = \beta_0 + 2\beta_1 \cos(\theta), \qquad c(z) = \beta_0 + 2\beta_1 z, \tag{3.24}$$

with $b(0) = \beta_0 + 2\beta_1 = 1$ and $c(z)$ defined as in (3.13).

From *Lemma 8* the symbol $b(\theta)$ might have no zeros, one zero or two zeros in the interval $[0, 2\pi]$. In the case of only one zero the root of $c(z)$ in (3.24) is $z_0 = \arccos(\pi) = -1$ and then $\beta_0 = 2\beta_1$, n has to be even so that $\pi \in \frac{2\mathbb{N}\pi}{n} \cap [0, 2\pi]$. The case of two zeros implies that the root $|z_0| < 1$ and thus $|\beta_0| < 2|\beta_1|$. We should recall that the root has to be $z_0 = \arccos(\frac{2s\pi}{n})$ for some $s \in \{1, \dots, \lfloor \frac{n}{2} \rfloor\}$.

If the root $|z_0| > 1$ then $b(\theta)$ does not vanish and the matrix M_n is non-singular. In this case we have

$$\begin{cases} \beta_0 + 2\beta_1 = 1 \\ |\beta_0| > 2|\beta_1| \end{cases} \Leftrightarrow \begin{cases} \beta_0 + 2\beta_1 = 1 \\ \beta_0^2 - 4\beta_1^2 > 0 \end{cases} \Leftrightarrow \begin{cases} \beta_0 + 2\beta_1 = 1 \\ \beta_0 - 2\beta_1 > 0 \end{cases}$$
$$\Leftrightarrow \left\{\beta_0 > \tfrac{1}{2}, \beta_1 < \tfrac{1}{4}\right\}. \tag{3.25}$$

Therefore we obtain the following.

Lemma 10. *For a primal subdivision scheme with symmetric first limit stencil* $\boldsymbol{\beta}^0 = \{\beta_1, \beta_0, \beta_1\}$*, if* $\beta_0 > \frac{1}{2}$ *and* $\beta_1 < \frac{1}{4}$*, then* M_n *is non-singular.*

Let us suppose that $b\left(\frac{2s\pi}{n}\right) = 0$ for some $s \in \{0, \dots, n-1\}$ and $b\left(\frac{2t\pi}{n+1}\right) = 0$ for some $t \in \{0, \dots, n\}$. Then there are two possibilities as $b(\theta) = b(2\pi - \theta)$.

The first one is that

$$\frac{s\pi}{n} + \frac{t\pi}{n+1} = \pi \Leftrightarrow s(n+1) + tn = n(n+1),$$

and the latter leads to a contradiction because n and $n+1$ are coprime, and it should hold that n divides s.

On the other hand, it could happen that

$$\frac{t\pi}{n+1} - \frac{s\pi}{n} = 0 \Leftrightarrow s(n+1) = tn,$$

and this leads to the same contradiction. Thus, it does not happen that $b(\theta)$ has as roots $\frac{2s\pi}{n}$ and $\frac{2t\pi}{n+1}$ for some s and t. The latter statement addresses the second question.

In order to answer the first question, we should look at the linear system of equations

$$\begin{cases} b(0) = \beta_0 + 2\beta_1 & = 1, \\ b\left(\frac{2s\pi}{n}\right) = \beta_0 + 2\beta_1 \cos\left(\frac{2s\pi}{n}\right) & = 0. \end{cases} \tag{3.26}$$

Thus, we find

$$\beta_0 = 1 - \frac{1}{2\sin^2\left(\frac{s\pi}{n}\right)}, \quad \beta_1 = \frac{1}{4\sin^2\left(\frac{s\pi}{n}\right)} \qquad \text{for some } s \in \{1, \dots, n-1\}. \tag{3.27}$$

Then it happens that also $b\left(\frac{2(n-s)\pi}{n}\right) = 0$. If n is even and $s = n/2$, then we find the case of one root.

We can now provide the following result.

Proposition 1. *Let* $\boldsymbol{\beta}^0 = \{\beta_1, \beta_0, \beta_1\}$ *be the first limit stencil of a dual subdivision scheme. Then,* M_n *is singular if and only if* M_{n+1} *is non-singular.*

Second degree trigonometric polynomials

When the first limit stencil is $\boldsymbol{\beta}^0 = \{\beta_2, \beta_1, \beta_0, \beta_1, \beta_2\}$ with $\beta_2 \neq 0$, we have

$$\mathbf{b}_n = [\beta_0, \beta_1, \beta_2, \mathbf{0}_{1\times(n-5)}, \beta_2, \beta_1] \in \mathbb{R}^n$$

and

$$b(\theta) = \beta_0 + 2\beta_1 \cos(\theta) + 2\beta_2 \cos(2\theta), \tag{3.28}$$

$$c(z) = \beta_0 + 2\beta_1 T_1(z) + 2\beta_2 T_2(z) = \beta_0 - 2\beta_2 + 2\beta_1 z + 4\beta_2 z^2. \tag{3.29}$$

To see how many roots the function $b(\theta)$ has, we analyze the roots of $c(z)$ in $[-1, 1]$. Even if we have a quadratic polynomial, we follow an approach that can provide insights for a general case. First we find its extreme values that is

$$c'(z) = 2\beta_1 + 8\beta_2 z = 0 \Leftrightarrow z = -\frac{\beta_1}{4\beta_2}.$$

Then we observe that this function may have roots in $\left[-1, -\frac{\beta_1}{4\beta_2}\right]$ and $\left[-\frac{\beta_1}{4\beta_2}, 1\right]$ if and only if

$$\left|\frac{\beta_1}{4\beta_2}\right| \leq 1. \tag{3.30}$$

Otherwise, the polynomial has a root in $[-1, 1]$ if and only if

$$c(-1)c(1) = \beta_0 - 2\beta_1 + 2\beta_2 \leq 0.$$

This condition, together with $c(1) = 1$ implies that

$$\beta_1 \geq \tfrac{1}{4}, \qquad \beta_0 + 2\beta_2 \leq \tfrac{1}{2}. \tag{3.31}$$

If the root is $z = -1$, then also $b(\theta)$ has only one root. In this case $\beta_1 = \frac{1}{4}$ and $\beta_0 + 2\beta_2 = \frac{1}{2}$.

The function $b(z)$ has no roots when the polynomial $c(z)$ has no real roots or when it vanishes outside the interval $[-1, 1]$

$$c(-1)c(1) = \beta_0 - 2\beta_1 + 2\beta_2 > 0,$$

which implies

$$\beta_1 < \tfrac{1}{4}, \qquad \beta_0 + 2\beta_2 > \tfrac{1}{2}. \tag{3.32}$$

Lemma 11. *For a primal subdivision scheme with symmetric first limit stencil $\boldsymbol{\beta}^0 = \{\beta_2, \beta_1, \beta_0, \beta_1, \beta_2\}$ with $\beta_2 \neq 0$, if $|\beta_1| > 4|\beta_2|$, $\beta_1 < \frac{1}{4}$ and $\beta_0 + 2\beta_2 > \frac{1}{2}$,then M_n is non-singular.*

If (3.30) holds, then the function vanishes in the interval $\left[-\frac{\beta_1}{4\beta_2}, 1\right]$ when

$$c\left(-\frac{\beta_1}{4\beta_2}\right)c(1) = \beta_0 - 2\beta_2 - \frac{\beta_1^2}{4\beta_2} \leq 0.$$

The latter is equivalent to

$$\beta_2\left(4\beta_0\beta_2 - \beta_1^2 - 8\beta_2^2\right) \leq 0. \tag{3.33}$$

Analogously, to determine whether there is none or one root in the interval $\left[-1, -\frac{\beta_1}{4\beta_2}\right]$, we use the inequality

$$c(-1)c\left(-\frac{\beta_1}{4\beta_2}\right) = \frac{(\beta_0 - 2\beta_1 + 2\beta_2)\left(4\beta_0\beta_2 - \beta_1^2 - 8\beta_2^2\right)}{4\beta_2} \leq 0,$$

that is equivalent to

$$\beta_2(\beta_0 - 2\beta_1 + 2\beta_2)\left(4\beta_0\beta_2 - \beta_1^2 - 8\beta_2^2\right) \leq 0. \tag{3.34}$$

If there is a root in $\left[-\frac{\beta_1}{4\beta_2}, 1\right]$, then this condition reduces to $\beta_0 - 2\beta_1 + 2\beta_2 \geq 0$.

The case $c\left(-\frac{\beta_1}{4\beta_2}\right) = 0$ provides a single root with double multiplicity. In that case, $4\beta_0\beta_2 - \beta_1^2 - 8\beta_2^2 = 0$.

Therefore, recalling *Lemma 8*, the amount of zeros of $b(\theta)$ can be from 0 to 4, as summarized in the Table 3.1.

no roots	$\left\{\lvert\beta_1\rvert > 4\lvert\beta_2\rvert,\ \beta_1 < \frac{1}{4},\ \beta_0 + 2\beta_2 > \frac{1}{2}\right\}$
one root	$\left\{\beta_1 = \frac{1}{4},\ \beta_0 + 2\beta_2 = \frac{1}{2}\right\}$
two roots	$\left\{\lvert\beta_1\rvert > 4\lvert\beta_2\rvert,\ \beta_1 \geq \frac{1}{4},\ \beta_0 + 2\beta_2 \leq \frac{1}{2}\right\}$ or $\{\lvert\beta_1\rvert \leq \lvert 4\beta_2\rvert,\ 4\beta_0\beta_2 - \beta_1^2 - 8\beta_2^2 = 0\}$
three roots	$\left\{\beta_1 = \frac{1}{4},\ \lvert\beta_2\rvert > \frac{1}{16},\ \beta_2\left(4\beta_0\beta_2 - 8\beta_2^2 - \frac{1}{16}\right) \leq 0,\ \beta_0 + 2\beta_2 = \frac{1}{2}\right\}$
four roots	$\{\lvert\beta_1\rvert < 4\lvert\beta_2\rvert,\ \beta_2\left(4\beta_0\beta_2 - \beta_1^2 - 8\beta_2^2\right) \leq 0,\ \beta_0 - 2\beta_1 + 2\beta_2 > 0\}$

Table 3.1: Conditions for the existence of roots of the symbol $b(\theta)$.

Example:

The J-spline subdivision scheme produces curves which are at least C^1 for $\nu \in [-1.7, 5.8]$, C^2 for $\nu \in (0, 4)$, C^3 for $\nu \in (1, 2.8]$ and C^4 for $\nu = 3/2$, which corresponds to the uniform quintic B-spline subdivision scheme [101].

For this family of schemes

$$\beta_0 = \tfrac{72+2(\nu-9)\nu}{12(6+\nu)}, \quad \beta_1 = \tfrac{2\nu(8-\nu)}{12(6+\nu)}, \quad \beta_2 = \tfrac{(\nu-1)\nu}{12(6+\nu)}, \tag{3.35}$$

the constraint (3.30) becomes

$$\left|\frac{2(8-\nu)}{(\nu-1)}\right| \leq 1 \Leftrightarrow \nu \in \left[\tfrac{17}{3}, 15\right]. \tag{3.36}$$

Then, we could only consider $\nu \in \left[\frac{17}{3}, 5.8\right)$ to analyze the existence of more than one root, but this interval would be only of interest for the C^1 continuity and no more than that smoothness. From this, the only possibility for a C^2 scheme in this family to have a singular interpolation matrix M_n is that (3.31) holds, which results in

$$\tfrac{72+2(\nu-9)\nu}{12(6+\nu)} - \tfrac{4\nu(8-\nu)}{12(6+\nu)} + \tfrac{2(\nu-1)\nu}{12(6+\nu)} = \frac{(2\nu-9)(\nu-2)}{3(6+\nu)} \leq 0 \Leftrightarrow \nu \in \left[2, \tfrac{9}{2}\right]. \tag{3.37}$$

The other case $\nu \leq -6$ is out of the convergence interval.

Therefore, as a first conclusion, for $\nu \in (-1.7, 2) \cup \left(\frac{9}{2}, 5.8\right)$, M_n is non-singular.

The condition

$$\frac{(2\nu-9)(\nu-2)}{3(6+\nu)} = 0 \Leftrightarrow \nu = 2, \nu = \tfrac{9}{2} \tag{3.38}$$

implies two different scenarios. For $\nu = 2$, we find $c(z) = \frac{5}{12} + \frac{1}{2}z + \frac{1}{12}z^2 = (z+1)(z+\frac{1}{5})$. As $\cos(\pi\nu) = -\frac{1}{5}$ has no rational solution, the only root of $b(\theta)$ in the grid $\frac{2\pi}{n}\mathbb{Z} \cap [0, \pi]$ is $\theta = \pi$, of even $n \in \mathbb{N}$.

On the other hand, for $\nu = \frac{9}{2}$, we get $c(z) = \frac{1}{2}z + \frac{1}{2}z^2 = \frac{1}{2}z(z+1)$. Then, we obtain the roots of $b(\theta)$, $\theta_1 = \frac{\pi}{2}$ and $\theta_2 = \pi$. If $n \mod (2) = 0$ then $b(\theta)$ vanishes in the grid only at θ_1, but if $n \mod (4) = 0$, then $b(\theta)$ vanishes at $\left\{\frac{\pi}{2}, \pi, 3\frac{\pi}{2}\right\}$.

If $\nu \in \left(2, \frac{9}{2}\right)$, then there is one root of $c(z)$ in $[-1, 1]$. That root should be confirmed to belong to the grid of interest in order to imply the singularity of M_n.

$\square$

Let us suppose that $b\left(\frac{2s\pi}{n}\right) = 0$ for some $s \in \{0, \ldots, n-1\}$ and $b\left(\frac{2t\pi}{n+1}\right) = 0$ for some $t \in \{0, \ldots, n\}$.

Then

$$\begin{aligned} 0 &= b\left(\tfrac{2s\pi}{n}\right) - b\left(\tfrac{2t\pi}{n+1}\right) \\ &= \beta_0 + 2\beta_1 \cos\left(\tfrac{2s\pi}{n}\right) + 2\beta_2 \cos\left(\tfrac{4s\pi}{n}\right) - \left(\beta_0 + 2\beta_1 \cos\left(\tfrac{2t\pi}{n+1}\right) + 2\beta_2 \cos\left(\tfrac{4t\pi}{n+1}\right)\right) \end{aligned}$$

no roots	$\nu \in [0,2) \cup (2, \frac{9}{2}) \cup (\frac{9}{2}, 4]$
one root	$\nu = 2, \quad n \mod (2) = 0$
two roots	$\nu \in (2, \frac{9}{2})$ if $b(\theta)$ has a root in the grid $\frac{2\pi}{n}\mathbb{Z} \cap [0,\pi]$
three roots	$\nu = \frac{9}{2}, \quad n \mod (4) = 0$

Table 3.2: Conditions for the existence of roots of the J-spline symbol $b(\theta)$ in the grid $\frac{2\pi}{n}\mathbb{Z} \cap [0,\pi]$.

$$\begin{aligned}
&= 2\beta_1 \left(\cos\left(\tfrac{2s\pi}{n}\right) - \cos\left(\tfrac{2t\pi}{n+1}\right)\right) + 2\beta_2 \left(\cos\left(\tfrac{4s\pi}{n}\right) - \cos\left(\tfrac{4t\pi}{n+1}\right)\right) \\
&= \left(\cos\left(\tfrac{2s\pi}{n}\right) - \cos\left(\tfrac{2t\pi}{n+1}\right)\right) \left(2\beta_1 + 4\beta_2 \left(\cos\left(\tfrac{2s\pi}{n}\right) + \cos\left(\tfrac{2t\pi}{n+1}\right)\right)\right).
\end{aligned}$$

The first factor is non-zero as it was analyzed before. Therefore we have

$$0 = 2\beta_1 + 4\beta_2 \left(\cos\left(\tfrac{2s\pi}{n}\right) + \cos\left(\tfrac{2t\pi}{n+1}\right)\right). \tag{3.39}$$

With (3.39) we have the linear system of equations

$$\begin{cases}
b(0) = \beta_0 + 2\beta_1 + 2\beta_2 & = 1, \\
b\left(\tfrac{2s\pi}{n}\right) = \beta_0 + 2\beta_1 \cos\left(\tfrac{2s\pi}{n}\right) + 2\beta_2 \cos\left(\tfrac{4s\pi}{n}\right) & = 0, \\
2\beta_1 + 4\beta_2 \left(\cos\left(\tfrac{2s\pi}{n}\right) + \cos\left(\tfrac{2t\pi}{n+1}\right)\right) & = 0.
\end{cases} \tag{3.40}$$

By using standard elimination operations, we arrive to the equivalent system

$$\begin{cases}
\beta_0 + 2\beta_1 + 2\beta_2 & = 1, \\
(2 - 2\cos\left(\tfrac{2s\pi}{n}\right))\beta_1 + (4 - 4\cos^2\left(\tfrac{2s\pi}{n}\right))\beta_2 & = 1, \\
4\left(\cos\left(\tfrac{2s\pi}{n}\right) - 1\right)\left(\cos\left(\tfrac{2t\pi}{n+1}\right) - 1\right)\beta_2 & = 1.
\end{cases} \tag{3.41}$$

For s and t different from zero, the respective coefficients of β_2 in the third row and β_1 in the second row are non zero. Thus, the system has a unique solution once fixed the values of s, t and $n = n_0 \in \mathbb{N}$.

Remark 13. *For those values of $\beta_0, \beta_1, \beta_2$ that define $b(\theta)$ with roots $\frac{2s\pi}{n_0}$ and $\frac{2t\pi}{n_0+1}$, the question is if it is possible that $b(\theta)$ also vanishes in the grids $\frac{2\pi}{n}\mathbb{Z} \cap [0,\pi]$ and $\frac{2\pi}{n+1}\mathbb{Z} \cap [0,\pi]$ for other values of $n \neq n_0 \in \mathbb{N}$.*

The answer is positive and that translates into the fact that

$$\tfrac{s}{n_0} \in \left(\tfrac{\mathbb{Z}}{n} \cap \tfrac{\mathbb{Z}}{n_0}\right) \quad \text{and} \quad \tfrac{t}{n_0+1} \in \left(\tfrac{\mathbb{Z}}{n+1} \cap \tfrac{\mathbb{Z}}{n_0+1}\right) \Leftrightarrow$$

$$n = k\frac{n_0(n_0+1)}{\gcd(s,n_0)\gcd(t,n_0+1)} + n_0, \ k \in \mathbb{N} \tag{3.42}$$

or,

$$\tfrac{t}{n_0+1} \in \left(\tfrac{\mathbb{Z}}{n} \cap \tfrac{\mathbb{Z}}{n_0+1}\right) \quad \text{and} \quad \tfrac{s}{n_0} \in \left(\tfrac{\mathbb{Z}}{n+1} \cap \tfrac{\mathbb{Z}}{n_0}\right) \Leftrightarrow$$

$$n = k\frac{n_0(n_0+1)}{\gcd(s,n_0)\gcd(t,n_0+1)} - (n_0+1), \ k \in \mathbb{N}. \tag{3.43}$$

Then, the following result can be stated.

Proposition 2. *Let $\boldsymbol{\beta}^0 = \{\beta_2, \beta_1, \beta_0, \beta_1, \beta_2\}$ be the first limit stencil of a dual subdivision scheme. If M_n is singular, as well as M_{n+1}, then we deduce that M_{n+2} is non-singular.*

Proof. Let n be such that M_n and M_{n+1} are both singular. Then, if M_{n+2} were also singular, by the previous results and (3.43), we would have

$$n + 2 = \frac{n(n+1)}{\gcd(s,n)\gcd(t,n+1)} - (n+1), \tag{3.44}$$

which leads to a contradiction as no divisor of $n+1$ can also divide $n+2$. □

Arbitrary degree case

Let us consider now the general symbol in (3.9). As we stated in *Lemma 8*, M_n is singular iff $b\left(\frac{2j\pi}{n}\right) = 0$ for some $j = 0, \dots, n-1$, which is equivalent to $c\left(\cos\left(\frac{2j\pi}{n}\right)\right) = 0$, where

$$c(z) = \beta_0 + 2\sum_{k=1}^{p} \beta_k \, T_k(z).$$

As $c(z)$ is a polynomial of degree p, then it can be defined by p roots.

Theorem 3. *Let $b(\theta)$ be such that it vanishes in the set*

$$\left\{\frac{2j_0\pi}{n_0}, \frac{2j_1\pi}{n_0+1}, \dots, \frac{2j_{p-1}\pi}{n_0+p-1}\right\}, \quad n_0, p \in \mathbb{N}^*, \tag{3.45}$$

for some integers $j_s \in \{0, \dots, n_0+s\}$ and $s = 0, \dots, p-1$. Then for the symbol $b(\theta)$ we find that the associated matrices $M_{n_0}, M_{n_0+1}, \dots, M_{n_0+p-1}$ are all singular.

Proof. The roots of $b(\theta)$ are defined by the roots of $c(z) = c(cos(\theta))$. If $b(\theta)$ is a trigonometric polynomial of degree p (3.9), then the associated Chebyshev series $c(z)$ in (3.15) is a polynomial of degree p and is has the roots

$$\left\{\cos\left(\frac{2j_0\pi}{n_0}\right), \cos\left(\frac{2j_1\pi}{n_0+1}\right), \ldots, \cos\left(\frac{2j_{p-1}\pi}{n_0+p-1}\right)\right\}, \quad n_0, p \in \mathbb{N}^*, \tag{3.46}$$

and those are the only roots of $b(\theta)$. From the study of the spectrum of M_n we deduce that all the matrices M_{n_0}, M_{n_0+1}, ..., M_{n_0+p-1} are singular, because they have at least one zero in their spectra. □

It just remains to analyze the case where for some $n \in \mathbb{N}$ exists $j \in \{1, \ldots, n-1\}$, such that

$$\frac{2j\pi}{n} \in \left\{\frac{2j_0\pi}{n_0}, \frac{2j_1\pi}{n_0+1}, \ldots, \frac{2j_{p-1}\pi}{n_0+p-1}\right\}, \quad n_0, p \in \mathbb{N}^*. \tag{3.47}$$

In that case, the matrix M_n associated with the same symbol is also singular.

3.1.2 Dual subdivision schemes

For this kind of subdivision schemes the limit stencil is of the form

$$\boldsymbol{\beta}^0 = \{\beta_p, \ldots, \beta_1, \beta_0, \beta_0, \beta_1, \ldots, \beta_p\}.$$

Let us analyze first what happens for the particular cases of $p = 0$ and $p = 1$.

In the case of first degree trigonometric polynomials, i.e. $p = 0$, from (3.10), we find a symbol of the form $b(\theta) = 2e^{\frac{i\theta}{2}}\beta_0 \cos\left(\frac{\theta}{2}\right)$. Then we have that $b(\theta) = 0$ if and only if $\theta = \pi$, independent of the value of β_0. Besides, because of the $b(0) = 1$, we obtain $\beta_0 = \frac{1}{2}$.

The value π belongs to the grid $\frac{2\pi}{n}\mathbb{Z} \cap [0, \pi]$ if and only if n is even. Therefore, we can state the following result.

Proposition 3. *Let $\boldsymbol{\beta} = \{\beta_0, \beta_0\}$ be the first limit stencil of a dual subdivision scheme. Then, M_n is singular if and only if n is even.*

Second degree trigonometric polynomials

For $p = 1$ we obtain the symbol

$$b(\theta) = 2e^{\frac{i\theta}{2}}\left(\beta_0 \cos\left(\tfrac{\theta}{2}\right) + \beta_1 \cos\left(\tfrac{3\theta}{2}\right)\right) = 2e^{\frac{i\theta}{2}} \cos\left(\tfrac{\theta}{2}\right)\left(\beta_0 + \beta_1\left[4\cos^2\left(\tfrac{\theta}{2}\right) - 3\right]\right)$$

$$= 2e^{\frac{i\theta}{2}} \cos\left(\tfrac{\theta}{2}\right) \left[\beta_0 - 3\beta_1 + 4\beta_1 \cos^2\left(\tfrac{\theta}{2}\right)\right], \tag{3.48}$$

and the associated Chebyshev series

$$c(z) = z\left(\beta_0 - 3\beta_1 + 4\beta_1 z^2\right). \tag{3.49}$$

As in the previous case, we have $b(\frac{\pi}{2}) = 0$ independent of the limit stencil $\boldsymbol{\beta}^0 = \{\beta_1, \beta_0, \beta_0, \beta_1\}$, and then M_n is singular for every even $n \in \mathbb{N}$.

Let us analyze when the factor $c(z)z^{-1} = \beta_0 - 3\beta_1 + 4\beta_1 z^2$ vanishes in the grid $\cos\left(\frac{2\pi}{2n+1}\mathbb{Z}\right) \cap [-1, 1]$ for $n \in \mathbb{N}$. It follows that

$$c\left(z_0\right) = \beta_0 - 3\beta_1 + 4\beta_1 z_0^2 = 0 \quad \Leftrightarrow \quad \frac{3\beta_1 - \beta_0}{4\beta_1} \in [0, 1], \tag{3.50}$$

for $z_0 = \cos\left(\frac{2s\pi}{2n+1}\right)$, $s \in \{1, \ldots, 2n\}$.

As in the primal case, it is possible to provide a stencil $\boldsymbol{\beta}^0$ such that the symbol $b(\theta)$ vanishes at the value $\frac{2s\pi}{2n+1}$, as solution of the linear system of equations

$$\begin{cases} \beta_0 + \beta_1 & = \frac{1}{2}, \\ \beta_0 - 3\beta_1 + 4\beta_1 \cos^2\left(\frac{2s\pi}{2n+1}\right) & = 0. \end{cases} \tag{3.51}$$

The unique solution for β_0, β_1 is guaranteed provided that $\sin\left(\frac{2s\pi}{2n+1}\right) \neq 0$, for $s \in \{1, \ldots, 2n\}$.

Then the values in the limit stencil are

$$\beta_0 = \tfrac{1}{2} - \beta_1, \qquad \beta_1 = \frac{1}{8\sin^2\left(\frac{2s\pi}{2n+1}\right)}, \qquad \text{for some } s \in \{1, \ldots, 2n\}. \tag{3.52}$$

Proposition 4. *Let $\boldsymbol{\beta}^0 = \{\beta_1, \beta_0, \beta_0, \beta_1\}$ be the first limit stencil of a dual subdivision scheme. If M_{2n}, M_{2n+1} and M_{2n+2} are singular, then M_{2n+3} is non-singular.*

Proof. For each $n \in \mathbb{N}$, the odd numbers $2n + 1$ and $2n + 3$ are coprimes, then if $b(\theta)$ vanishes in the grid $\frac{2\pi}{2n+1}\mathbb{Z} \cap [0, \pi]$, it does not vanish in the grid $\frac{2\pi}{2n+3}\mathbb{Z} \cap [0, \pi]$.

□

Quartic degree case

The next considered case is that of $p = 2$ with a symbol of the form

$$\begin{aligned} b(\theta) &= 2e^{\frac{i\theta}{2}} \left(\beta_0 \cos\left(\tfrac{\theta}{2}\right) + \beta_1 \cos\left(\tfrac{3\theta}{2}\right) + \beta_2 \cos\left(\tfrac{5\theta}{2}\right)\right) && (3.53)\\ &= 2e^{\frac{i\theta}{2}} \cos\left(\tfrac{\theta}{2}\right) \left(\beta_0 + \beta_1 \left[4\cos^2\left(\tfrac{\theta}{2}\right) - 3\right] + \beta_2 \left[16\cos^4\left(\tfrac{\theta}{2}\right) - 20\cos^2\left(\tfrac{\theta}{2}\right) + 5\right]\right) \\ &= 2e^{\frac{i\theta}{2}} \cos\left(\tfrac{\theta}{2}\right) \left([\beta_0 - 3\beta_1 + 5\beta_2] + [4\beta_1 - 20\beta_2]\cos^2\left(\tfrac{\theta}{2}\right) + 16\cos^4\left(\tfrac{\theta}{2}\right)\right), && (3.54) \end{aligned}$$

and the associated Chebyshev series

$$c(z) = z\left(\beta_0 - 3\beta_1 + 5\beta_2 + [4\beta_1 - 20\beta_2]\, z^2 + 16z^4\right). \tag{3.55}$$

We can study the second factor in the previous expression with the change of variables $\lambda = z^2$. As a result, the analysis of this case is similar to the one provided for the second degree case for primal schemes, with the additional constraint of non-negativity of the roots.

As an example of this case we have the scheme with symbol (3.23) proposed in [43].

Arbitrary degree case

Let us consider now the general symbol in (3.10). Similar to the case of primal subdivision schemes, from *Lemma 9*, M_n is singular iff $b\left(\frac{2j\pi}{n}\right) = 0$ for some $j = 0, \ldots, n-1$, and that is equivalent to $c\left(\cos\left(\frac{j\pi}{n}\right)\right) = 0$, where

$$c(z) = \sum_{k=0}^{p} \beta_k\, T_{2k+1}(z).$$

As $c(z)$ is a polynomial of degree $2p+1$, it can be defined by its $2p+1$ roots, where one of them is $z = 0$ corresponding to $\theta = \pi$.

Theorem 4. *Let $b(\theta)$ be such that it vanishes in the set*

$$\left\{\frac{2j_0\pi}{2n_0+1}, \frac{2j_1\pi}{2n_0+3}, \ldots, \frac{2j_{p-1}\pi}{2n_0+2p-1}\right\}, \quad n_0, p \in \mathbb{N}^*, \tag{3.56}$$

for some integers $j_s \in \{0, \ldots, 2n_0 + 2s + 1\}$ and $s = 0, \ldots, p-1$. Then we have for the symbol $b(\theta) = 2e^{\frac{i\theta}{2}} c\left(\cos\left(\frac{\theta}{2}\right)\right)$ that the associated matrices $M_{2n_0}, M_{2n_0+1}, \ldots, M_{2n_0+2p-1}$ are all singular. Besides, for any symbol $b(\theta)$ in (3.10) the associated matrix M_{2n} is *singular.*

The proof is analogous to that of the previous theorem for the primal case.

3.2 Hermite interpolation problem

As a natural extension, we are ready to take into account the interpolation problem with a subdivision curve of n points with associated derivatives up to the $d-1$-th order, where the 0 derivative in a point is the point itself.

Let

$$\mathbf{U}^{(d)} = [V_0^0, V_0^1, \ldots, V_0^{d-1}, V_1^0, V_1^1, \ldots, V_1^{d-1}, \ldots, V_{n-1}^0, V_{n-1}^1, \ldots, V_{n-1}^{d-1}]^\top$$

be the data that we want to interpolate.

We can suppose that there exists a parameter sequence $\{t_j, j = 0, \ldots, n-1\}$ such that the subdivision curve $c(t)$ interpolates the data information in $\mathbf{U}^{(d)}$, i.e.,

$$c^{(l)}(t_j) = V_j^l, \qquad \text{for } l = 0, \ldots, d-1, \; j = 0, \ldots, n. \tag{3.57}$$

If the curve $c(t)$ is defined by n control points as in (2.27), then we compute a solution for the point interpolation problem (2.27), which may contradict the values of higher order derivatives.

Thus, in order to be able to interpolate all the information in $\mathbf{U}^{(d)}$, we need to use nd control points $\mathbf{P}^0 = \{P_j^0 \in \mathbb{R}^m, j = 0, \ldots, dn-1\}$ with the periodization $P_j^0 = P_{j+nd}^0$, $j \in \mathbb{Z}$. A natural choice is to consider the parameters in (3.57) to be $t_j = dj$, $j = 0, \ldots, n-1$. Then, from (2.28) we can express the following nd equations

$$c^{(l)}(di) = V_{di}^l = \sum_{j\in\mathbb{Z}} P_j^0 \beta_{di-j}^l, \qquad i = 0, \ldots, n-1, \; l = 0, \ldots, d-1. \tag{3.58}$$

These equations can be represented in a compact matrix form as

$$\mathbf{M}_n \mathbf{P}^0 = \begin{bmatrix} \boldsymbol{B}_0 & \boldsymbol{B}_{-1} & \boldsymbol{B}_{-2} & \ldots & \boldsymbol{B}_2 & \boldsymbol{B}_1 \\ \boldsymbol{B}_1 & \boldsymbol{B}_0 & \boldsymbol{B}_{-1} & \ldots & \boldsymbol{B}_3 & \boldsymbol{B}_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{B}_{-1} & \boldsymbol{B}_{-2} & \boldsymbol{B}_{-3} & \ldots & \boldsymbol{B}_1 & \boldsymbol{B}_0 \end{bmatrix} \begin{bmatrix} P_0^0 \\ P_1^0 \\ \vdots \\ P_{dn-1}^0 \end{bmatrix} = \mathbf{U}^{(d)}, \tag{3.59}$$

where the $d \times d$ blocks of the matrix $\mathbf{M}_n \in \mathbb{R}^{nd \times nd}$ satisfy $\boldsymbol{B}_j = \boldsymbol{B}_{j-n}$ for $j = 1, \ldots, n$ and

$$\boldsymbol{B}_j = \begin{bmatrix} \beta^0_{dj} & \beta^0_{dj-1} & \beta^0_{dj-2} & \cdots & \beta^0_{d(j-1)+1} \\ \beta^1_{dj} & \beta^1_{dj-1} & \beta^1_{dj-2} & \cdots & \beta^1_{d(j-1)+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta^{d-1}_{dj} & \beta^{d-1}_{dj-1} & \beta^{d-1}_{dj-2} & \cdots & \beta^{d-1}_{d(j-1)+1} \end{bmatrix}. \tag{3.60}$$

Example:

Let us consider the cubic B-spline scheme, whose first limit stencil is $\{\frac{1}{6}, \frac{4}{6}, \frac{1}{6}\}$ and the second limit stencil is $\{-\frac{1}{2}, 0, \frac{1}{2}\}$. Then we write the matrix

$$\left[\begin{array}{cc|cc|c|cc|cc} \frac{4}{6} & \frac{1}{6} & 0 & 0 & \cdots & 0 & 0 & 0 & \frac{1}{6} \\ 0 & \frac{1}{2} & 0 & 0 & & 0 & 0 & 0 & -\frac{1}{2} \\ \hline 0 & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} & \ddots & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} & & 0 & 0 & 0 & 0 \\ \hline & \vdots & & \ddots & \ddots & & \ddots & & \vdots \\ \hline 0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \\ 0 & 0 & 0 & 0 & & 0 & -\frac{1}{2} & 0 & \frac{1}{2} \end{array}\right] \tag{3.61}$$

which is singular with a kernel of dimension 1.

□

Since the resulting structure is a block-circulant matrix, we infer that:

$$\mathbf{M}_n = (\mathbf{F}_n \otimes \mathbf{I}_d)\ \mathcal{L}_n\ (\mathbf{F}^*_n \otimes \mathbf{I}_d) \tag{3.62}$$

with $\mathcal{L}_n$ as defined in (2.76). As the limit stencils have compact support on $[-p, q]$, there are at least $\lceil \frac{p}{d} \rceil + \lceil \frac{q}{d} \rceil$ not null blocks in the set $\{\boldsymbol{B}_j, j = 0, \ldots, -n+1\}$. As a consequence, we find the representation

$$(\mathcal{L}_n)_{s,s} = \sum_{j=0}^{\lceil \frac{p}{d} \rceil} \boldsymbol{B}_{-j} e^{-2(s-1)ji\pi/n} + \sum_{j=1}^{\lceil \frac{q}{d} \rceil} \boldsymbol{B}_j e^{2(s-1)ji\pi/n}, \quad s = 1, \ldots, n. \tag{3.63}$$

Lemma 12. *For an odd-symmetric subdivision scheme and $d = 2$, it holds that the matrix $\mathbf{M}_n$ is singular.*

Proof. From (2.34), (2.33), and $\beta^1_0 = 0$ for odd-symmetric schemes, we have that the first block of the matrix $\mathcal{L}_n$ satisfies

$$
\begin{aligned}
(\mathcal{L}_n)_{1,1} &= \sum_{j=0}^{\lceil \frac{p}{2} \rceil} \begin{bmatrix} \beta^0_{-2j} & \beta^0_{-2j-1} \\ \beta^1_{-2j} & \beta^1_{-2j-1} \end{bmatrix} + \sum_{j=1}^{\lceil \frac{q}{2} \rceil} \begin{bmatrix} \beta^0_{2j} & \beta^0_{2j-1} \\ \beta^1_{2j} & \beta^1_{2j-1} \end{bmatrix} \\
&= \begin{bmatrix} \beta^0_0 & \beta^0_1 \\ -\beta^1_0 & -\beta^1_1 \end{bmatrix} + \sum_{j=1}^{\lceil \frac{p}{2} \rceil} \begin{bmatrix} \beta^0_{2j} & \beta^0_{2j+1} \\ -\beta^1_{2j} & -\beta^1_{2j+1} \end{bmatrix} + \sum_{j=1}^{\lceil \frac{q}{2} \rceil} \begin{bmatrix} \beta^0_{2j} & \beta^0_{2j-1} \\ \beta^1_{2j} & \beta^1_{2j-1} \end{bmatrix} \\
&= \begin{bmatrix} \beta^0_0 + 2\sum_{j=1}^{\lceil \frac{p}{2} \rceil} \beta^0_{2j} & 2\sum_{j=1}^{\lceil \frac{p}{2} \rceil} \beta^0_{2j-1} \\ 0 & 0 \end{bmatrix}.
\end{aligned}
\tag{3.64}
$$

Because there is a null row in (3.64), it follows that the first block $(\mathcal{L}_n)_{1,1}$ is singular and then also the matrices $\mathcal{L}_n$ and $\mathbf{M}_n$, by virtue of (3.62). □

Instead, for even-symmetric subdivision schemes, due to (2.34) and (2.33), the first block is equal to

$$
\begin{aligned}
(\mathcal{L}_n)_{1,1} &= \sum_{j=0}^{\lceil \frac{p}{2} \rceil} \begin{bmatrix} \beta^0_{-2j} & \beta^0_{-2j-1} \\ \beta^1_{-2j} & \beta^1_{-2j-1} \end{bmatrix} + \sum_{j=1}^{\lceil \frac{q}{2} \rceil} \begin{bmatrix} \beta^0_{2j} & \beta^0_{2j-1} \\ \beta^1_{2j} & \beta^1_{2j-1} \end{bmatrix} \\
&= \begin{bmatrix} \sum_{j=-\lceil \frac{p}{2} \rceil}^{\lceil \frac{q}{2} \rceil} \beta^0_{2j} & \sum_{j=-\lceil \frac{p}{2} \rceil}^{\lceil \frac{q}{2} \rceil} \beta^0_{2j-1} \\ \sum_{j=-\lceil \frac{p}{2} \rceil}^{\lceil \frac{q}{2} \rceil} \beta^1_{2j} & \sum_{j=-\lceil \frac{p}{2} \rceil}^{\lceil \frac{q}{2} \rceil} \beta^1_{2j-1} \end{bmatrix} = \begin{bmatrix} \Sigma_1 & \Sigma_1 \\ \Sigma_2 & -\Sigma_2 \end{bmatrix},
\end{aligned}
\tag{3.65}
$$

with

$$
\Sigma_1 = \sum_{j=-\lceil \frac{p}{2} \rceil}^{\lceil \frac{q}{2} \rceil} \beta^0_{2j} \quad \text{and} \quad \Sigma_2 = \sum_{j=-\lceil \frac{p}{2} \rceil}^{\lceil \frac{q}{2} \rceil} \beta^1_{2j}.
$$

Therefore, $(\mathcal{L}_n)_{1,1}$ is not singular in general, but for some particular stencils with $\Sigma_1 = 0$ or $\Sigma_2 = 0$.

The previous Lemma suggests to consider pseudo-inverse matrices for the solution of the interpolation problem (3.59).

Proposition 5. *Given the vector* $\mathbf{U}^{(d)}$*, the interpolation problem* $\mathbf{M}_n\mathbf{P}^0 = \mathbf{U}^{(d)}$ *in*

(3.59) *has solution* $\mathbf{P}^0 = \mathbf{M}_n^\dagger \mathbf{U}^{(d)}$*, where:*

$$(\mathbf{M}_n^\dagger)_{s,t} = \frac{1}{n}(\mathcal{L}_n)_{1,1}^\dagger + \frac{1}{n}\sum_{j=2}^{n}(\mathcal{L}_n)_{j,j}^\dagger \, e^{2(t-s)(j-1)\mathrm{i}\pi/n}, \qquad \textit{for } s,t = 1,\dots,n. \tag{3.66}$$

Proof. Given the block-circulant matrix in (3.59) $\mathbf{M}_n = (\mathbf{F}_n \otimes \mathbf{I}_d)\mathcal{L}_n(\mathbf{F}_n^* \otimes \mathbf{I}_d)$ we can write

$$\begin{aligned}\mathbf{P}^0 = \mathbf{M}_n^\dagger \mathbf{U}^{(d)} &= \left[(\mathbf{F}_n \otimes \mathbf{I}_d)\mathcal{L}_n(\mathbf{F}_n^* \otimes \mathbf{I}_d)\right]^\dagger \mathbf{U}^{(d)} \\ &= (\mathbf{F}_n \otimes \mathbf{I}_d)\ \mathcal{L}_n^\dagger\ (\mathbf{F}_n^* \otimes \mathbf{I}_d)\mathbf{U}^{(d)}.\end{aligned} \tag{3.67}$$

Since

$$\begin{bmatrix}(\mathcal{L}_n)_{1,1} & \mathbf{0} & \dots & \mathbf{0}\\ \mathbf{0} & (\mathcal{L}_n)_{2,2} & \ddots & \mathbf{0}\\ \vdots & \ddots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \dots & (\mathcal{L}_n)_{n,n}\end{bmatrix}^\dagger = \begin{bmatrix}(\mathcal{L}_n)_{1,1}^\dagger & \mathbf{0} & \dots & \mathbf{0}\\ \mathbf{0} & (\mathcal{L}_n)_{2,2}^\dagger & \ddots & \mathbf{0}\\ \vdots & \ddots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \dots & (\mathcal{L}_n)_{n,n}^\dagger\end{bmatrix}, \tag{3.68}$$

the $d \times d$ block $(\mathbf{M}_n^\dagger)_{s,t}$ is such that

$$\begin{aligned}(\mathbf{M}_n^\dagger)_{s,t} &= \frac{1}{n}\sum_{j=1}^{n} e^{-(s-1)(j-1)\mathrm{i}\pi/n} e^{(t-1)(j-1)\mathrm{i}\pi/n}(\mathcal{L}_n)_{j,j}^\dagger \\ &= \tfrac{1}{n}(\mathcal{L}_n)_{1,1}^\dagger + \tfrac{1}{n}\sum_{j=2}^{n} e^{2(t-s)(j-1)\mathrm{i}\pi/n}(\mathcal{L}_n)_{j,j}^\dagger.\end{aligned}$$

□

Remark 14. *Even though the representation in* (3.66) *provides a way to compute* $\mathbf{M}_n^\dagger$*, it is more efficient in the implementation to compute* $\mathbf{M}_n^\dagger \mathbf{U}^{(d)}$ *by benefiting from the Fourier factorization in* (3.62)*. In this way, the computational cost of that operation reduces to* $\mathcal{O}(dn\log(n))$*.*

The proof of *Lemma 12* leads to a characterization of a vector in the kernel of the matrix M_n as follows.

Proposition 6. *Let us consider an odd-symmetric subdivision scheme with first limit* *stencil $\boldsymbol{\beta}^0$ and second limit stencil $\boldsymbol{\beta}^1$. The vector w with components*

$$w_{2j+s} = \begin{cases} \Sigma_2 & \text{if } s = 1 \\ -\Sigma_1 & \text{if } s = 2 \end{cases} \qquad j = 0, \ldots, n-1, \tag{3.69}$$

where

$$\Sigma_1 = \sum_{j=-\lceil \frac{p}{2} \rceil}^{\lceil \frac{p}{2} \rceil} \beta^0_{2j} \quad \text{and} \quad \Sigma_2 = \sum_{j=-\lceil \frac{p}{2} \rceil}^{\lceil \frac{p}{2} \rceil} \beta^0_{2j-1},$$

belongs to the kernel of the interpolating matrix M_n for the Hermite problem with $d = 2$.

Proof. Let w be a vector defined like in (3.69). Then, we have that $(\mathbf{F}^* \otimes \mathbf{I}_2)w = \frac{1}{\sqrt{n}}[\Sigma_2, -\Sigma_1, \mathbf{0}_{1\times(2n-2)}]^\top$.

If we compute $M_n\, w$ with (3.62), then we verified that

$$\begin{aligned} M_n\, w &= (\mathbf{F}_n \otimes \mathbf{I}_2)\ \mathcal{L}_n\ (\mathbf{F}_n^* \otimes \mathbf{I}_2)\ w \\ &= \tfrac{1}{\sqrt{n}}(\mathbf{F}_n \otimes \mathbf{I}_2)\ \mathcal{L}_n\ [\Sigma_2, -\Sigma_1, \mathbf{0}_{1\times(2n-2)}]^\top = \tfrac{1}{\sqrt{n}}(\mathbf{F}_n \otimes \mathbf{I}_2)\ \mathbf{0}_{2n\times 1} \\ &= \mathbf{0}_{1\times 2n}, \end{aligned}$$

because the first block of $\mathcal{L}_n$ is equal to $\begin{bmatrix} \Sigma_1 & \Sigma_2 \\ 0 & 0 \end{bmatrix}$. □

If $dim\ Ker(M_n) = 1$, then w is a basis of the kernel. Otherwise, we consider it as a vector in the basis for one of the proposed solutions in the next section (see *Algorithm 3*).

For higher derivatives interpolation the blocks (3.60) composing M_n in (3.59) do not present the symmetry that leads to singular matrix independently on the mask as in (3.64). Hence, the possibility for a singular operator with Hermite interpolation of order 2 or higher depends on the mask and not on the amount of points.

3.3 Solving the interpolation problem

The previous sections provide the characterization for the singularity of the interpolation operator M_n in (3.59). The matrix M_n is a block-circulant matrix with blocks of dimension $d \times d$, with $d-1$ the maximum order of derivatives interpolated as Hermite problem. The results are summarized in the following Table 3.3.

<table>
<tr><th>Mask Symmetry</th><th colspan="2">Point ($d = 1$)</th><th>Points and tangents ($d = 2$)</th><th>Points and higher derivatives ($d \geq 3$)</th></tr>
<tr><td>Odd</td><td colspan="2">Depend on mask</td><td>Always singular</td><td>Depend on mask</td></tr>
<tr><td rowspan="2">Even</td><td>odd n</td><td>Depend on mask</td><td rowspan="2">Depend on mask</td><td rowspan="2">Depend on mask</td></tr>
<tr><td>even n</td><td>Always singular</td></tr>
</table>

Table 3.3: Singularity of the interpolation operator M_n

When M_n is non-singular it is possible to provide an exact solution, as in the case of odd-symmetric subdivision scheme for $d = 1$.

Corollary 2. *Let $\boldsymbol{\beta}^0$ be the first limit stencil of an odd-symmetric subdivision scheme. The subdivision curve that interpolates a given set of points $\mathbf{V}^0$ has the set $\mathbf{P}^0$ as control points, defined by the rule*

$$\mathbf{P}^0 = \mathbf{M}_n^{-1}\mathbf{U}^{(1)}, \tag{3.70}$$

where the elements of matrix $\mathbf{M}_n^{-1}$ are given by

$$(\mathbf{M}_n^{-1})_{s,t} = \begin{cases} \frac{1}{n} + \frac{\cos((s-t)\pi)}{n\left(\beta_0^0 + 2\sum_{j=1}^{p}\beta_j^0\cos(j\pi)\right)} + \frac{2}{n}\sum_{r=1}^{\frac{n}{2}-1}\frac{\cos(2r(s-t)\pi/n)}{\beta_0^0+2\sum_{j=1}^{p}\beta_j^0\cos(2jr\pi/n)}, & n \mod 2 = 0, \\ \frac{1}{n} + \frac{2}{n}\sum_{r=1}^{\lfloor\frac{n}{2}\rfloor}\frac{\cos(2r(s-t)\pi/n)}{\beta_0^0+2\sum_{j=1}^{p}\beta_j^0\cos(2jr\pi/n)}, & n \mod 2 = 1, \end{cases}$$

with $s, t = 1, \ldots, n$.

Proof. Given the first limit stencil $\boldsymbol{\beta}^0$, the matrix $\mathbf{M}_n$ is circulant with the first row equal to $[\beta_0^0, \beta_{-1}^0, \ldots, \beta_1^0]$ as setted in (3.59) for the case $d = 1$. Then, following *Lemma 6* we obtain the factorization $\mathbf{M}_n = \mathbf{F}\mathbf{L}_n\mathbf{F}^*$ with $\mathbf{L}_n = \text{diag}(\mathbf{F}\ \boldsymbol{\beta}^0)$, which is equivalent to write

$$(\mathbf{L}_n)_{s,s} = \beta_0^0 + 2\sum_{j=1}^{n-1}\beta_j^0\cos(2js\pi/n) = (\mathbf{L}_n)_{n+1-s,n+1-s}. \tag{3.71}$$

Therefore, the inverse $\mathbf{M}_n^{-1}$ can be computed as $\mathbf{M}_n^{-1} = \mathbf{F}\mathbf{L}_n^{-1}\mathbf{F}^*$ and then for the (s,t) entry we have

$$(\mathbf{M}_n^{-1})_{s,t} = \frac{1}{n}\sum_{j=0}^{n-1}\frac{e^{-(s-1)ji\pi/n}e^{(t-1)ji\pi/n}}{(\mathbf{L}_n)_{j,j}}$$

$$= \tfrac{1}{n} + \tfrac{1}{n} \sum_{j=1}^{n-1} e^{(t-s)ji\pi/n} (\mathbf{L}_n)^{-1}_{j,j}.$$

From the symmetry of the values in the diagonal of $\mathbf{L}_n$ and the fact that $e^{ai} + e^{-ai} = 2\cos(a)$ we deduce the final expression. $\square$

Remark 15. *Even when we have a way to construct $(M_n)^{-1}$, it is not efficient to compute the inverse and then multiply it by $\mathbf{U}^{(1)}$. In that sense, it is better to look at $(M_n)^{-1}\mathbf{U}^{(1)} = \mathbf{F}\mathbf{L}_n\mathbf{F}^*\mathbf{U}^{(1)}$ as the sequence of operations:*

- *compute the inverse Fourier transform of $\mathbf{U}^{(1)}$: $\mathbf{F}^*\mathbf{U}^{(1)}$,*
- *filter the result with the diagonal of $\mathbf{L}_n^{-1}$: $\mathbf{L}_n^{-1}\mathbf{F}^*\mathbf{U}^{(1)}$,*
- *compute the Fourier transform of the filtered vector: $\mathbf{F}\mathbf{L}_n\mathbf{F}^*\mathbf{U}^{(1)}$.*

In that way the computational cost is just related with the Fourier transformation, which can be done with a Fast Fourier implementation that is $\mathcal{O}(n\log n)$, as the filter in the second step is just $\mathcal{O}(n)$.

The Hermite interpolation follows the same approach, where $(\mathbf{F} \otimes I_d)\mathbf{U}^{(d)}$ is computed by the Fourier transform of the d vectors $\left[U_0^{(j)}, \dots U_{n-1}^{(j)}\right]$, with $j = 0, \dots, d-1$. Therefore, the computational cost is $\mathcal{O}(\ n\log n)$, because the values of d are small.

Algorithm 2 Computing the control points to interpolate the data points and associated tangents

/* Interpolation operator M_n is not singular */

/* circshift(arg,s): shift circularly rows of vector/matrix in argument by s positions, if positive ->, if negative <- */

Input: subdivision mask $\mathbf{a}$, data points $\mathbf{U}^{(d)}$

Output: control points to interpolate data points

1: Compute limit stencils $\boldsymbol{\beta}^s$, $s = 0, \dots, d-1$

2: Define $\boldsymbol{\beta} = \begin{bmatrix} \beta_q^0 & \dots & \beta_0^0 & \dots & \beta_{-p}^{d-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_q^{d-1} & \dots & \beta_0^{d-1} & \dots & \beta_{-p}^{d-1} \end{bmatrix}_{d\times(q+p+1)}$

3: Compute $\mathbf{b}_n = circshift\left([\boldsymbol{\beta}, \mathbf{0}_{d\times(dn-(q+p+1))}], -\lceil\frac{(q+p+1)}{2}\rceil\right)$

4: Compute $\mathcal{L}_n = \mathrm{diag}((\mathbf{F}^* \otimes \mathbf{I}_d)\mathbf{b}_n^\top)$

5: Compute $\mathcal{L}_n^{-1} = \mathrm{diag}((L_n)_1^{-1}, \dots, (L_n)_n^{-1})$ // block inverse in (3.63)

6: **return** $\mathbf{P}^0 = \mathtt{real}((\mathbf{F} \otimes \mathbf{I}_d)\mathbf{L}_n^{-1}(\mathbf{F}^* \otimes \mathbf{I}_d))\ \mathbf{U}^{(d)})$

The Algorithm 2 for non-singular operator M_n does not construct the matrix, instead it uses the Fourier transform of vectors. Such an approach save computational cost and memory.

Nevertheless, if we use that algorithm for a singular operator, then from the *aesthetic point of view*[3] the solution in the least square sense of (3.66) (i.e., for the pseudo-inverse definition) may not be the desired one (see Fig. 3.1).

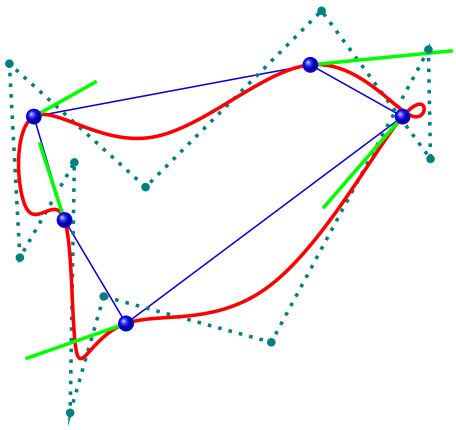


Figure 3.1: Interpolating points and tangents directions with a cubic-Bpline curve, considering the Least Square solution.

The next section covers a perturbation strategy proposed.

3.3.1 Regularization as a potential alternative

When the interpolation matrix M_n is singular, the interpolation problem in (3.59) may have no solution or the solution is not unique. In this case we do not fulfill the Hadamard's definition of *well-posedness* and we can consider different known methods to solve this problem. An alternative is considering a *regularization*, which is an approximation of the *ill-posed* problem by a family of neighbouring *well-posed* problems [52].

One known choice as *regularization method* is Tikhonov regularization

$$\operatorname{argmin}_x \|Ax - b\|^2 + \lambda \|Lx\|^2 , \tag{3.72}$$

[3] We can call that *null space artifacts*!

with the family of solutions depending on the *regularization parameter* λ. The latter controls the weight given to the residual norm and the regularization term. The optimal value for λ can be chosen by *discrepancy principle*, *generalized cross-validation*, or the *L-curve method* [63].

The operator L, which can be taken as the *identity operator* or a *differential operator*[4], looks for altering the least square solution to enforce special features of the regularized approximations [63, 52]. As a necessary condition is required that $Ker(A) \ \cap \ Ker(L) = \{0\}$.

This problem is equivalent to solve the normal equations $(A^\top A + \lambda L^\top L)x = A^\top \ b$ for each λ.

Our problem involves the square matrix M_n and that fact motivates us to explore the regularization $(M_n + \lambda\mathbf{I})\mathbf{P}^0 = \mathbf{U}^{(d)}$ instead. In this case the spectrum of M_n is shifted by λ. Therefore, with a suitable λ each matrix $M_n + \lambda\mathbf{I}$ is non singular and the problem is well-posed.

We can analyze another alternative that consists of solving another regularized problem $M_{n,\omega}\mathbf{P}^0 = \mathbf{V}^0$, exchanging the circulant matrix by a ω-*circulant matrix* [36].

To define our matrix $M_{n,\omega}$ we will use the complex number $\omega = e^{\mathrm{i}\psi}$. Then, by Remark 8, the spectrum of our matrix $M_{n,\omega}$ is given by the symbol $b(\theta + \frac{\psi}{n})$. In this way, the eigenvalues of M_n are shifted. As a consequence, the matrices M_n and $M_{n,\omega}$ are both singular only if there at least two roots of $b(\theta)$ in the grid $\frac{2\pi\mathbb{N}}{n} \cap [0, 2\pi]$ with distance $\frac{\psi}{n}$ among them.

If $M_{n,\omega}$ is not singular, then the system of equations $M_{n,\omega}\mathbf{P}^0 = \mathbf{V}^0$ is a perturbation of the initial problem $M_n\mathbf{P}^0 = \mathbf{V}^0$, that has a unique solution when the former does not. The question is then the existence of a complex solution when the original problem is set in the real domain. By choosing ψ small in modulus the perturbed system is close to the original one. Furthermore, since the imaginary part is then small enough, taking the real part of the solution provides a good approximation.

We can illustrate the performance in the Hermite interpolation scenario by considering the particular case of point and tangent interpolation. If we consider the

[4]For instance, the first or second derivative operator.

corresponding ω-circulant matrix $M_{n,\omega}$ we obtain that the block in (3.64) becomes

$$
\begin{aligned}
(\mathcal{L}_{n,\omega})_{1,1} &= \sum_{j=0}^{\lceil \frac{p}{2} \rceil} \boldsymbol{B}_j \; \omega^{j/n} + \sum_{j=1}^{\lceil \frac{p}{2} \rceil} \boldsymbol{B}_{-j} \; \omega^{(n-j)/n} \\
&= \begin{bmatrix} \beta_0^0 & \beta_1^0 \\ \beta_0^1 & -\beta_1^1 \end{bmatrix} + \sum_{j=1}^{\lceil \frac{p}{2} \rceil} \begin{bmatrix} \beta_{2j}^0 & \beta_{2j-1}^0 \\ \beta_{2j}^1 & \beta_{2j-1}^1 \end{bmatrix} \omega^{j/n} + \sum_{j=1}^{\lceil \frac{p}{2} \rceil} \begin{bmatrix} \beta_{2j}^0 & \beta_{2j+1}^0 \\ -\beta_{2j}^1 & -\beta_{2j+1}^1 \end{bmatrix} \omega^{(n-j)/n} \\
&= \begin{bmatrix} \beta_0^0 + \sum_{j=1}^{\lceil \frac{p}{2} \rceil} \beta_{2j}^0 \left(\omega^{j/n} + \omega^{(n-j)/n}\right) & \beta_1^0 \left(1 + \omega^{(n-1)/n}\right) + \sum_{j=2}^{\lceil \frac{p}{2} \rceil} \beta_{2j-1}^0 \left(\omega^{j/n} + \omega^{(n-j)/n}\right) \\ \sum_{j=1}^{\lceil \frac{p}{2} \rceil} \beta_{2j}^1 \left(\omega^{j/n} - \omega^{(n-j)/n}\right) & \beta_1^1 \left(\omega^{1/n} - 1\right) + \sum_{j=2}^{\lceil \frac{p}{2} \rceil} \beta_{2j-1}^1 \left(\omega^{j/n} - \omega^{(n-j)/n}\right) \end{bmatrix}.
\end{aligned}
$$

Then, the singularity of M_n does not imply the singularity of $M_{n,\omega}$ (as in *Lemma 12*) and the system $M_{n,\omega}\mathbf{P}^0 = \mathbf{U}^{(d)}$ can be solved instead of (3.59).

It is worth noting that in this general case, the symbol does not follow the same structure as in (3.9) and (3.10), rather, it follows a more complex one.

The regularization strategy present as a drawback the need to solve many equations to tune properly the regularization parameter. The tuning for the Tikhonov method is independent of the symbol $b(\theta)$. Meanwhile, for the other two strategies perturbing the spectrum of M_n, the parameter can be chosen by a residual criterion taking into account the roots of the symbol $b(\theta)$.

In what follows, other strategies are proposed avoiding the tuning required by the regularization.

3.3.2 A proposed functional strategy

The previous section showed that the interpolation problem (2.27) is not well defined in general for linear subdivision schemes. If the matrix M_n which defines the interpolation $M_n\mathbf{P}^0 = \mathbf{U}^{(d)}$ is singular, then we have two possibilities regarding the possible solutions. According to the Rouché-Capelli Theorem, a solution does not exist at all or there are an infinite number of them. The second case could be solved by choosing a suitable energy functional which selects the *best* solution in terms of its energy.

The first case is undesirable for the designer who would like to have a curve through those control points. In this case the proposed strategy is based on what follows. As noted earlier, the singularity of M_n depends on the limit stencils and the amount of control points $n \in \mathbb{N}$. Once the roots of the symbol $b(\theta)$ are known, it is possible to increase the amount of control points so that the symbol does not vanish in the grid $\frac{2\pi}{n}\mathbb{Z} \cap [0, \pi]$.

In this section we propose a strategy to perturb the least square solution with vectors in the nullspace of M_n, by using the energies proposed in [116].

Let $\mathbf{W}$ be the matrix with columns $\{w_i,\ i = 1, \dots, \dim(\mathrm{Ker}(\mathbf{M}_n))\}$ a basis of the kernel of $\mathbf{M}_n$, i.e.

$$\mathrm{Ker}(\mathbf{M}_n) = \mathrm{span}\{w_i,\ i = 1, \dots, \dim(\mathrm{Ker}(\mathbf{M}_n))\}. \tag{3.73}$$

If $\widehat{\mathbf{P}}^0$ is a solution in the least square sense of (3.66), then we deduce that

$$\mathbf{P}^0 = \widehat{\mathbf{P}}^0 + \mathbf{W}s = \mathbf{M}_n^{\dagger}\mathbf{U}^{(d)} + \mathbf{W}\mathbf{s}, \qquad \mathbf{s} \in \mathbb{R}^{\dim(\mathrm{Ker}(\mathbf{M}_n))\times d}, \tag{3.74}$$

is also a solution of the interpolation problem in (3.58).

Recalling the combined energy in (2.80)

$$\mathcal{E}(c) \ = \ \alpha_1\, \widetilde{\mathcal{E}}_{stretch}(c) \ + \ \alpha_2\, \widetilde{\mathcal{E}}_{bend}(c) \ + \ \alpha_3\, \widetilde{\mathcal{E}}_{twist}(c),$$

if we consider $\mathbf{P^0} = \widehat{\mathbf{P}}^0 + \mathbf{W}\mathbf{s}$, then we obtain the following expression

$$\begin{aligned}\mathcal{E}(c) \ = \sum_{\ell=1}^{m} \Big[\ & \mathbf{s}(:,\ell)^\top\ \mathbf{W}^\top\ \mathbf{D}\ \mathbf{W}\ \mathbf{s}(:,\ell) \ + \ 2\,\widehat{\mathbf{P}}^0(:,\ell)^\top\ \mathbf{D}\ \mathbf{W}\ \mathbf{s}(:,\ell) \\ & + \widehat{\mathbf{P}}^0(:,\ell)^\top\ \mathbf{D}\ \widehat{\mathbf{P}}^0(:,\ell)\ \Big],\end{aligned}$$

where

$$\mathbf{D} \ := \ \sum_{r=1}^{3} \alpha_r\, \mathbf{D}_r^\top\, \mathbf{G}_r\, \mathbf{D}_r, \tag{3.75}$$

with $\mathbf{D}_r$ and $\mathbf{G}_r$, $r = 1, 2, 3$, defined in (2.82), (2.87), (2.83), and (2.89), respectively.

Thus $\mathcal{E}(c)$ is a component-wise quadratic functional and we can easily find the $\mathbf{s}$ that minimizes it as

$$\mathbf{s}(:,\ell) \ = \ -\,(\ \mathbf{W}^\top\ \mathbf{D}\ \mathbf{W}\ )^{-1}\ \mathbf{W}^\top\ \mathbf{D}^\top\ \widehat{\mathbf{P}}^0(:,\ell), \tag{3.76}$$

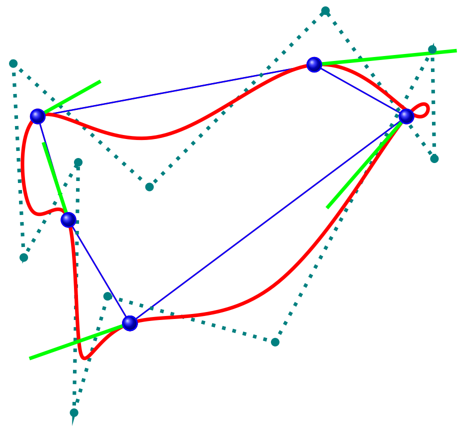
for $\ell = 1, \dots, m$.

With (3.76), the solution (3.74) can be written as

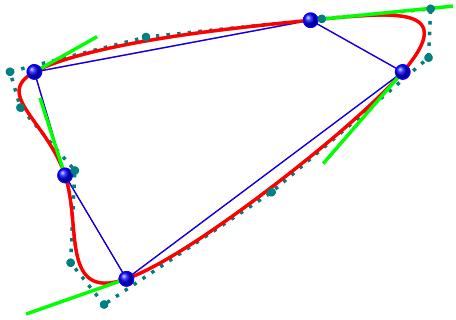
$$\begin{aligned}\mathbf{P}^0 &= \widehat{\mathbf{P}}^0 + \mathbf{W}s = \widehat{\mathbf{P}}^0 - \mathbf{W}(\mathbf{W}^\top\ \mathbf{D}\ \mathbf{W}\ )^{-1}\ \mathbf{W}^\top\ \mathbf{D}^\top\widehat{\mathbf{P}}^0 \\ &= \left(\mathbf{I}_{nd} - \mathbf{W}(\mathbf{W}^\top\ \mathbf{D}\ \mathbf{W}\ )^{-1}\ \mathbf{W}^\top\ \mathbf{D}^\top\right)\ (\mathbf{F}_n \otimes \mathbf{I}_d)\mathcal{L}_n^\dagger(\mathbf{F}_n^* \otimes \mathbf{I}_d)\ \mathbf{U}^{(d)}. \end{aligned} \tag{3.77}$$

Remark 16. *The matrix D in* (3.75) *is a circulant matrix, because D_1, D_2, and D_3 are circulant matrices. Therefore, D can be factorized by the Fast Fourier transform and the computations in* (3.76) *can be performed by using the Fast Fourier transform analogous to* Remark 15.

A family of control points can be provided by perturbing the particular solution with vectors in the $Ker(\mathbf{M}_n)$ (see Fig. 3.2).



(a) Least Square solution



(b) Perturbed solution

Figure 3.2: Interpolating points and tangents directions with a cubic-Bpline curve. The least square solution is perturbed with a vector in the nullspace.

Considering the fact that the stretch energy (based on arc-length) could dominate over the bend energy (based on curvature) and the twist energy (based on torsion), we propose to normalize the first by the arc-length of the polygon with vertices in $\mathbf{U}^{(1)}$.

Figure 3.2b shows the output for the parameters $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, and $\alpha_3 = 0$ in (3.75). In this case the twisting energy is not considered as we deal with a planar curve. For 3D curves we invite the reader to experiment with the codes provided in [35].

The procedure is performed as shown in *Algorithm 3*.

Algorithm 3 Perturbing the least square solution

```
/* Interpolation operator M_n is singular */
/* circshift(arg,s): shift circularly rows of vector/matrix in argument
by s positions, if positive ->, if negative <- */
```

Input: subdivision mask $\mathbf{a}$, data points $\mathbf{U}^{(d)}$

Output: control points to interpolate data points

1: Compute limit stencils $\boldsymbol{\beta}^s$, $s = 0, \ldots, d-1$

2: Define $\boldsymbol{\beta} = \begin{bmatrix} \beta^0_q & \ldots & \beta^0_0 & \ldots & \beta^{d-1}_{-p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta^{d-1}_q & \ldots & \beta^{d-1}_0 & \ldots & \beta^{d-1}_{-p} \end{bmatrix}_{d\times(q+p+1)}$

3: Compute $\mathbf{b}_n = circshift\left([\boldsymbol{\beta}, \mathbf{0}_{d\times(dn-(q+p+1))}], -\lceil\frac{(q+p+1)}{2}\rceil\right)$

4: Compute $\mathcal{L}_n = \mathrm{diag}((\mathbf{F}^* \otimes I_d)\mathbf{b}_n^\top)$

5: Compute $\mathcal{L}_n^\dagger = \mathrm{diag}((L_n)_1^\dagger, \ldots, (L_n)_n^\dagger)$ `// block pseudo-inverse in` (3.63)

6: Compute $\mathbf{W} = [w_1|w_2|\ldots|w_s]$ `//` $\{w_j, j = 1, \ldots, s\}$ `a basis of Ker(`M_n`), independent of` $\mathbf{U}^{(d)}$

`/* Alternatively, it can be used only` $\mathbf{W} = w$ `in` (3.69) `*/`

7: Compute $\mathbf{D}$ in (3.75) `// independent of` $\mathbf{U}^{(d)}$

8: **return** $\mathbf{P}^0 = \mathtt{real}\big(\left(\mathbf{I}_{nd} - \mathbf{W}(\mathbf{W}^\top\ \mathbf{D}\ \mathbf{W}\ )^{-1}\ \mathbf{W}^\top\ \mathbf{D}^\top\right)\ (\mathbf{F}_n \otimes \mathbf{I}_d)\mathcal{L}_n^\dagger(\mathbf{F}_n^* \otimes \mathbf{I}_d)\ \mathbf{U}^{(d)}\big)$

3.3.3 Shifting interpolation parameters: an alternative solution

The previous discussion is based on the supposition that the interpolated points $\mathbf{V}^0$ should be interpolated at integer parameters $c(j) = V_j^0$, for $j \in \{0, \ldots, n-1\}$, as it was stated in (2.27). However, if we change that assumption, then we have a different scenario. This subsection explores that idea and shows how it is possible to find a solution to the interpolation problem with a non-singular operator, so that the solution is unique. The idea is inspired in the works for B-splines in [93].

Let us reconsider the dual subdivision schemes case. Previously, with the interpolation at the parameters $t_j = j$, $j = 0, \ldots, n-1$, we obtained the first limit stencil as coefficients

$$V_j^0 = c(j) = \sum_{s\in\mathbb{Z}} \varphi(j-s)P_s^0 = \sum_{s=\mu}^{\nu} \varphi(s)P_{j-s}^0, \qquad \text{with } \operatorname{supp}\varphi = [\mu, \nu], \tag{3.78}$$

but we can also consider that

$$V_j^0 = c(j+\sigma) = \sum_{s\in\mathbb{Z}} \varphi(j-s+\sigma)P_s^0 = \sum_{s=\mu}^{\nu-1} \varphi(s+\sigma)P_{j-s}^0 \tag{3.79}$$

with $\sigma \in (0,1)$. In particular, we explore the choice $\sigma \in \frac{1}{q}\mathbb{Z}\cap(0,1)$ for a certain positive integer q. The evaluation of the basic function at those parameters is done as in (2.46).

Examples:

Let us analyze the quadratic uniform B-spline scheme for $\sigma = \frac{1}{2}$. In this case, for the interpolation at integer parameter values and from (2.38), we obtain that

$$V_j^0 = c(j) = \varphi(-1)P_{j+1}^0 + \varphi(0)P_j^0 + \varphi(1)P_{j-1}^0 + \varphi(2)P_{j-2}^0 = \tfrac{1}{2}P_j^0 + \tfrac{1}{2}P_{j-1}^0, \tag{3.80}$$

and the interpolation operator M_n is singular for even n (see *Proposition 3*). On the other hand, with the values obtained using (2.46) for $\frac{1}{2}\mathbb{Z}$, we have

$$\begin{aligned} V_j^0 = c(j+\tfrac{1}{2}) &= \varphi(-1+\tfrac{1}{2})P_{j+1}^0 + \varphi(0+\tfrac{1}{2})P_j^0 + \varphi(1+\tfrac{1}{2})P_{j-1}^0 \\ &= \frac{1}{8}P_{j+1}^0 + \frac{3}{4}P_j^0 + \frac{1}{8}P_{j-1}^0. \end{aligned} \tag{3.81}$$

In this way, the interpolation operator is not singular, because its symbol $\frac{3}{4} + \frac{1}{4}\cos(\theta)$ never vanishes in the grid $\frac{2\pi}{n}\mathbb{Z} \cap [0,\pi]$. Therefore, it is possible to interpolate any amount of given points.

In Figure 3.3 an even number of points is interpolated with the dual scheme proposed in [43]. In a first try, the least square solution (with red line) is computed by applying the interpolation at integer parameters. This solution does not interpolate the points, but the one obtained with the interpolation at the parameters $\mathbb{Z} + \frac{1}{2}$ (with blue line).

□

For dual schemes, the strategy shown allows us to obtain an interpolation operator with a symbol as in the primal case (3.13). Thus, the singularity analysis occurs as it did in the section concerning primal schemes. In addition, we recover the advantage of dealing with a symmetric circulant matrix, whose eigenvalues are represented as real trigonometric polynomials in cosines.

The use of shifted parameters for interpolation becomes in this way a degree of freedom for the geometry of the interpolation curve (see Fig. 3.4). However, it is

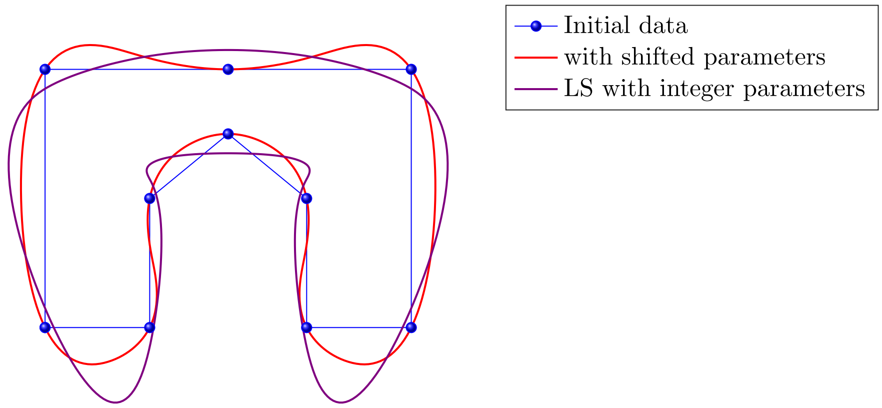


Figure 3.3: Point interpolation with a dual subdivision scheme [43] considering shifted parameters.

possible that we miss the symmetry provided by the subdivision scheme. As pointed out before, if the scheme is symmetric, then the basic function satisfies $\varphi(\mu+t) = \varphi(\nu-t)$, with $\operatorname{supp}\varphi = [\mu, \nu]$. In case of odd-symmetry with $\mu = -\nu$ then the set

$$\{\varphi(-\nu), \dots, \varphi(0), \dots, \varphi(\nu)\}$$

is odd-symmetric, but the set

$$\left\{\varphi(-\nu + \tfrac{1}{2}), \dots, \varphi(\tfrac{1}{2}), \dots, \varphi(\nu - \tfrac{1}{2})\right\}$$

is even-symmetric. On the other hand, for even-symmetric schemes with $\mu = -\nu + 1$, the former has even-symmetry while the latter has odd-symmetry.

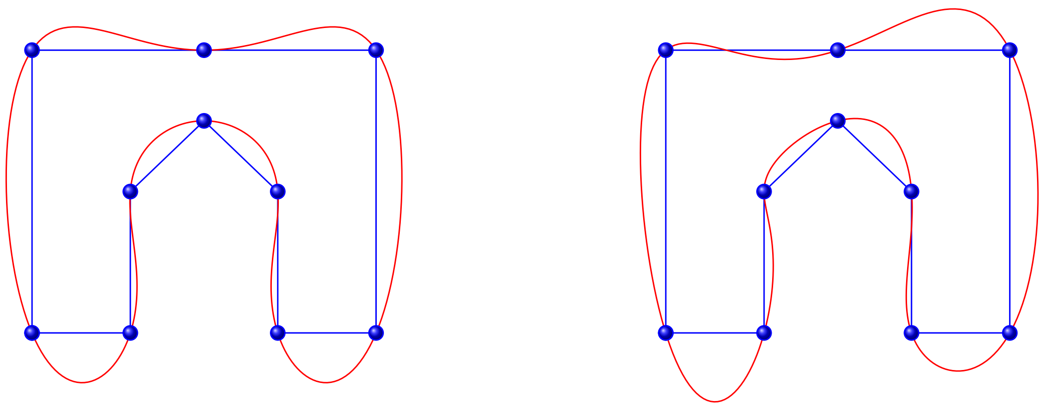
In general, the sets

$$\{\varphi(-\nu + \sigma), \varphi(-\nu + 1 + \sigma), \dots, \varphi(\sigma), \dots, \varphi(\nu - 1 + \sigma)\} \tag{3.82}$$

for odd-symmetric schemes and

$$\{\varphi(-\nu + \sigma), \varphi(-\nu + 1 + \sigma), \dots, \varphi(\sigma), \dots, \varphi(\nu + \sigma)\} \tag{3.83}$$

for even-symmetric schemes do not possess symmetry for all σ. Therefore, the subdivision curve interpolating the points at those parameters does not reproduce the symmetry of the data. That is to say, the obtained control points do not show symmetry, even with symmetric interpolated data (see Fig. 3.4).



(a) Interpolation at $c(\mathbb{Z})$ (b) Interpolation at $c\left(\left\{\mathbb{Z}+\frac{1}{5}\right\}\right)$

Figure 3.4: Interpolation with quintic uniform B-spline at different parameter values.

As a general approach, when the primal case has a singular interpolation operator, we can consider $V_j^0 = c(j+\sigma)$, with $\sigma \in (0,1)$. The given choice leads to an interpolation operator without the odd and even symmetries as the former cases. Consequently, the related symbol is a complex trigonometric polynomial. This fact has to be considered in the numerical implementations when manipulating the matrix M_n, taking into account that the solutions represent points in $\mathbb{R}^m$.

Extension to the Hermite interpolation at shifted parameters

In the point interpolation case with even-symmetric subdivision schemes, we propose shifting the interpolation parameter from integers to half integer parameters (i.e., $\frac{1}{2}+\mathbb{Z}$) to obtain a non singular interpolation operator. In other words, we shift the case from the symbol in the form (3.10) to (3.9). For Hermite interpolation we proceed in an analogous way only for odd-symmetric subdivision schemes, due to the fact that *Lemma 12* leads to a singular case but (3.65) does not.

The general formula for the first block of the matrix $\mathcal{L}_n$ in (3.62) is

$$(\mathcal{L}_n)_{1,1} = \sum_{j=-\lceil\frac{p-1}{2}\rceil}^{\lceil\frac{q}{2}\rceil} \begin{bmatrix} \varphi(2j+\sigma) & \varphi(2j-1+\sigma) \\ \varphi'(2j+\sigma) & \varphi'(2j-1+\sigma) \end{bmatrix} \tag{3.84}$$

$$= \begin{bmatrix} \varphi(\sigma) + \sum_{j=1}^{\lceil \frac{p-1}{2} \rceil} (\varphi(2j-\sigma) + \varphi(2j+\sigma)) & \varphi(1-\sigma) + \sum_{j=1}^{\lceil \frac{p-1}{2} \rceil} (\varphi(2j-1+\sigma) + \varphi(2j+1-\sigma)) \\ \varphi'(\sigma) + \sum_{j=1}^{\lceil \frac{p-1}{2} \rceil} (\varphi'(2j+\sigma) - \varphi'(2j-\sigma)) & -\varphi'(1-\sigma) + \sum_{j=1}^{\lceil \frac{p-1}{2} \rceil} (\varphi'(2j-1+\sigma) - \varphi'(2j+1-\sigma)) \end{bmatrix},$$

because of the symmetry relation $\varphi(t+\sigma) = \varphi(-t-\sigma)$ and $\varphi'(-t-\sigma) = -\varphi'(t+\sigma)$ for $t \geq 0$.

With $\sigma = 0$ and the symmetry relations in (2.34) and (2.33) the cases (3.64) and (3.65) are obtained. However, we swap the properties by considering $\sigma = \frac{1}{2}$. Indeed, the first block for odd-symmetric schemes is then:

$$(\mathcal{L}_n)_{1,1} = \sum_{j=-\lceil \frac{p-1}{2} \rceil}^{\lceil \frac{q}{2} \rceil} \begin{bmatrix} \varphi(2j+\frac{1}{2}) & \varphi(2j-\frac{1}{2}) \\ \varphi'(2j+\frac{1}{2}) & \varphi'(2j-\frac{1}{2}) \end{bmatrix} \tag{3.85}$$

$$= \begin{bmatrix} \varphi(\frac{1}{2}) + \sum_{j=1}^{\lceil \frac{p-1}{2} \rceil} (\varphi(2j-\frac{1}{2}) + \varphi(2j+\frac{1}{2})) & \varphi(\frac{1}{2}) + \sum_{j=1}^{\lceil \frac{p-1}{2} \rceil} (\varphi(2j-\frac{1}{2}) + \varphi(2j+\frac{1}{2})) \\ \varphi'(\frac{1}{2}) + \sum_{j=1}^{\lceil \frac{p-1}{2} \rceil} (\varphi'(2j+\frac{1}{2}) - \varphi'(2j-\frac{1}{2})) & -\varphi'(\frac{1}{2}) + \sum_{j=1}^{\lceil \frac{p-1}{2} \rceil} (\varphi'(2j-\frac{1}{2}) - \varphi'(2j+\frac{1}{2})) \end{bmatrix}$$

$$= \begin{bmatrix} \Sigma_1 & \Sigma_1 \\ \Sigma_2 & -\Sigma_2 \end{bmatrix}. \tag{3.86}$$

with

$$\Sigma_1 = \sum_{j=-\lceil \frac{p-1}{2} \rceil}^{\lceil \frac{q}{2} \rceil} \varphi\left(2j+\tfrac{1}{2}\right) \quad \text{and} \quad \Sigma_2 = \sum_{j=-\lceil \frac{p-1}{2} \rceil}^{\lceil \frac{q}{2} \rceil} \varphi'\left(2j+\tfrac{1}{2}\right). \tag{3.87}$$

This block is not singular for all stencils $\mathbf{b}^0$ and $\mathbf{b}^1$, as was the case when $\sigma = 0$ in *Lemma 12. Therefore, we propose the parameter value $\sigma = \frac{1}{2}$ as strategy for interpolating points and tangent vectors with odd-symmetric schemes. It is possible to use other choices for the parameter σ, although the symmetry of the stencils is lost.*

For even-symmetric schemes we stick to the choice $\sigma = 0$, following what it was said in (3.65). Once more, it is a free choice for the user to consider another parameter value.

It is worth noting that *Algorithm 5* becomes *Algorithm 2* for $\sigma = 0$.

Algorithm 4 Proposed values for σ

Input: subdivision mask $\mathbf{a}$,

Output: parameter σ for shifting the interpolation parameters

```
1:  if σ not provided then
2:      if d = 1 and a even-symmetric then
3:          σ = 1/2
4:      else if d = 1 and a odd-symmetric then
5:          σ = 0
6:      else if d = 2 and a even-symmetric then
7:          σ = 0                                  // see Algorithm 2, 3
8:      else if d = 2 and a odd-symmetric then
9:          σ = 1/2
10:     end if
11: end if
```

Algorithm 5 Computing control points to interpolate data points $c(\sigma + \mathbb{Z})$ and associated derivatives $c^{(d)}(\sigma + \mathbb{Z})$

Input: subdivision mask $\mathbf{a}$, data points $\mathbf{U}^{(d)}$, $\sigma \in (0, 1)$

Output: control points to interpolate data points

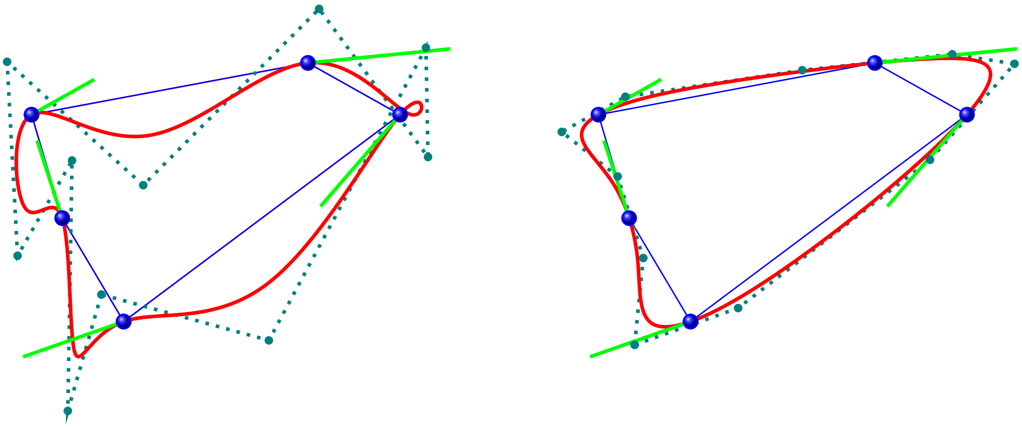
1: Compute $\boldsymbol{\beta}_\sigma = \begin{bmatrix} \varphi(-\nu+\sigma) & \dots & \varphi(\mu-1+\sigma) \\ \vdots & \vdots & \vdots \\ \varphi^{(d-1)}(-\nu+\sigma) & \dots & \varphi^{(d-1)}(\mu-1+\sigma) \end{bmatrix}_{d\times(\nu+\mu)}$

2: Compute $\mathbf{b}_n = circshift\left([\boldsymbol{\beta}_\sigma, \mathbf{0}_{d\times(dn-(\nu+\mu))}], -\lceil \frac{(\nu+\mu)}{2} \rceil\right)$

3: Compute $\mathcal{L}_n = \operatorname{diag}((\mathbf{F}^* \otimes I_d)\mathbf{b}_n^\top)$

4: Compute $\mathcal{L}_n^{-1} = \operatorname{diag}((L_n)_1^{-1}, \dots, (L_n)_n^{-1})$ `// block inverse`

5: **return** $\mathbf{P}^0 = \texttt{real}\left((\mathbf{F}_n \otimes \mathbf{I}_d)\, \mathcal{L}_n^{-1}\, (\mathbf{F}^* \otimes \mathbf{I}_d)\, \mathbf{U}^{(d)}\right)$



(a) Minimal-norm solution (b) Shifted-parameters solution

Figure 3.5: Interpolating points and tangents directions with a cubic-Bpline curve by shifting parameters.

3.4 Existing strategies comparison

In this section we show two iterative methods proposed in the literature to solve the interpolation problem, each one for different scalar subdivision schemes, the J-spline family and the cubic uniform B-splines. We compare their solutions with the one obtained with the direct method proposed in this thesis.

On the other hand, some examples comparing Hermite subdivision schemes with the Hermite interpolation strategy proposed are shown.

3.4.1 Recovery of retrofitting via J-splines

Let us consider the J-spline family of subdivision schemes [101] with subdivision rules already presented in (3.1). In the case of point interpolation, instead of solving the global system of equations (3.59), in [101] the authors proposed to use an iterative retrofitting scheme which converges rapidly to the solution of these equations. When a proof of convergence is not provided, it converges only for values of $\nu \in (-0.86, 2)$.

The proposed *retrofitting* strategy can be posed as the solution with an iterative method to the problem $\mathbf{M}_n\mathbf{P}^0 = \mathbf{U}^{(1)}$ in (3.59) for point interpolation that is

$$\mathbf{Q}^{k+1} = \mathbf{Q}^k + \left(\mathbf{U}^{(1)} - \mathbf{M}_n\mathbf{Q}^k\right) = (\mathbf{I}_n - \mathbf{M}_n)\mathbf{Q}^k + \mathbf{U}^{(1)}, \qquad k \in \mathbb{N} \tag{3.88}$$

with $\mathbf{Q}^0 = \mathbf{U}^{(1)}$ and where $\mathbf{Q}^k$ converges to $\mathbf{P}^0$ as k runs to infinity. Then the convergence depends on the spectral radius of the iteration matrix $\mathbf{I}_n - \mathbf{M}_n$, where $\mathbf{M}_n$ is a circulant matrix with first row defined by $\boldsymbol{\beta}^0$ in (3.2) as in (3.59). Then, the factorization in (3.62) gives us the convergence that depends on the spectral radius of $\mathbf{I}_n - \mathbf{L}_n$, with $\mathbf{L}_n$ defined in (3.63).

This formulation can be extended to the interpolation of higher-order derivatives, changing the matrix $\mathbf{M}_n$ similar to (3.59). Also in the present setting, the spectral radius should be controlled to ensure convergence.

Comparison with direct method

The first point to consider regarding the retrofitting method is the amount of operations to be done in relation to the amount n of control points. At each iteration (3.88) there is only a matrix-vector product with a sparse matrix, whose rows have only 5 non zero entries, and one addition. The computational cost is $\mathcal{O}(n)$, multiplied by the amount of required iterations for the convergence. However, the amount of iterations grows fast as the parameter ν grows towards 2 and beyond, while the spectral radius ρ of the iteration matrix $\mathbf{I} - \mathbf{M}_n$ grows and eventually is greater than 1 for $\nu = 2$ (see Table 3.4). Even for $\nu \geq 2$ the J-spline schemes can generate C^3 limit curves, but this strategy fails.

In Table 3.4 we compare, for a test case, how many iterations of the retrofitting method are needed to have the same *relative residual error (RRE)*

$$RRE := \max_{j=1,\dots,n} \frac{\|(\mathbf{M}_n\mathbf{P}^0 - \mathbf{U}^{(1)})_j\|}{\|\mathbf{U}^{(1)}_j\|},$$

as the direct method in (3.66).

ν	0	0.5	1.0	1.5	1.9	1.99	2.0	2.2
RRE	1.18e-15	1.47e-15	1.55e-15	1.44e-15	2.17e-15	1.29e-14	7.89e-2	1.92e-15
iter	1	34	78	222	1411	14117	19	19283
ρ	1.13e-15	0.38	0.67	0.87	0.98	0.99	1.0	1.04

Table 3.4: Retrofitting method with the threshold equals to the RRE of the direct method in (3.89).

In case of using the solution in *Algorithm 3* with the least square solution in (3.70), we have

$$\begin{aligned}\mathbf{P}^0 &= M_n^{\dagger}\mathbf{U}^{(1)} - \mathbf{W}\left(\mathbf{W}^{\top}D\mathbf{W}\right)^{-1}\left(\mathbf{W}^{\top}D^{\top}\widehat{\mathbf{P}}^0\right) \\ &= \left(\mathbf{I}_n - \mathbf{W}\left(\mathbf{W}^{\top}D\mathbf{W}\right)^{-1}\mathbf{W}^{\top}D^{\top}\right)\mathbf{F}\mathbf{L}_n^{\dagger}\mathbf{F}^*\,\widehat{\mathbf{P}}^0. \qquad (3.89)\end{aligned}$$

When the matrix M_n is not singular, the *filter* $\left(\mathbf{I}_n - \mathbf{W}\left(\mathbf{W}^{\top}D\mathbf{W}\right)^{-1}\mathbf{W}^{\top}D^{\top}\right)$ reduces to the identity matrix and the solution is obtained after three Fourier transformations with computational cost $\mathcal{O}(n\log n)$.

Otherwise, being $\mathbf{W}$ a vector computed in (3.69), the Fourier factorization of matrix $\mathbf{D}$ implies that the computational cost of (3.89) is $\mathcal{O}(n\log n)$. As the values of n considered in real application are not greater than 100, when ν start growing the cost of the retrofitting method $\mathcal{O}(iter\cdot n)$ becomes greater than the cost of the direct method in (3.89).

The use of *Algorithm 5* leads to a similar computational cost with respect to *Algorithm 3*.

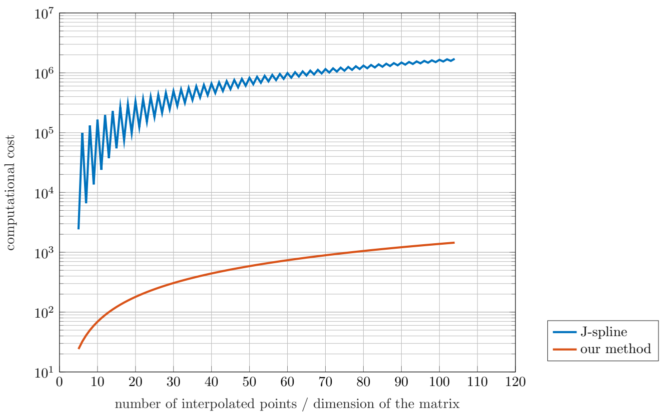


Figure 3.6: Comparing computational cost among the retrofitting method and the direct method proposed: $\mathcal{O}(n\cdot\#iterations)$ vs $\mathcal{O}(n\log n)$.

On the other hand, we can provide the solution for values of the parameter which defines the J-spline subdivision rules, for which the retrofitting strategy fails.

Regarding to the singularity of M_n, we can invoke *Corollary 2* with the solution

(3.70) whenever it exists. Thus, all the eigenvalues of $\mathbf{M}_n$

$$\beta_0^0 + 2\sum_{j=1}^{n-1} \beta_j^0 \cos(2js\pi/n), \quad \text{for } s = 0, \ldots, n-1, \tag{3.90}$$

should be non-zero. In the case $\nu = 2$, we have $\boldsymbol{\beta}^0 = \{1/48,\ 1/4,\ 11/24,\ 1/4,\ 1/48\}$ and for $s = 3$ we note that M_n has one eigenvalue equals to zero.

Finally, we can use this family of subdivision schemes to interpolate associated derivatives to the set of given points. That was not considered in the original contribution [101] and it is an advantage for the Hermite interpolation problem, because of the smoothness provided by some schemes in this family.

3.4.2 Recovery of Okaniwa's results via cubic B-splines

In [89] there are proposed solutions for the problems of points interpolation and also tangent interpolation at those points as well. Both approaches make use of the rules provided in (2.28), transforming them into an iterative method for solving a linear system of equations.

Points interpolation

For the problem of point interpolation the proposed rules are

$$Q_j^{k+1} = \tfrac{6}{4} V_j^0 - \tfrac{1}{4} Q_{j-1}^k - \tfrac{1}{4} Q_{j+1}^k, \qquad \text{with } Q_j^0 = V_j^0,\ j = 1, \ldots, n, \tag{3.91}$$

where $Q_j^k \to P_j^0$ as $k \to \infty$ for all $j \in \mathbb{Z}$.

That expression can be written as the iterative method

$$\mathbf{Q}^{k+1} = \left(\mathbf{I}_n - \tfrac{3}{2}\mathbf{M}_n\right) \mathbf{Q}^k + \tfrac{3}{2}\mathbf{U}^{(1)}, \qquad \text{with } \mathbf{Q}^0 = \mathbf{U}^{(1)}, \tag{3.92}$$

for the solution of the system of equations $\mathbf{M}_n \mathbf{P}^0 = \mathbf{U}^{(1)}$. The iteration matrix $I_n - \frac{3}{2}\mathbf{M}_n$ leads to a better convergence than $\mathbf{I}_n - \mathbf{M}_n$ in (3.88), as it can be verified with the respective spectral radii

$$\rho(\mathbf{I}_n - \tfrac{3}{2}\mathbf{M}_n) = \tfrac{1}{2} < \rho(\mathbf{I}_n - \mathbf{M}_n) = \tfrac{2}{3}. \tag{3.93}$$

Points and tangent directions interpolation

The problem of point and tangent vectors interpolation deals with the difficulty to be overdetermined, as n points P_j^0 uniquely determine by (2.26) another n points V_i^0 and their respective tangent vectors V_i^1. Consequently, providing different values for V_i^1 gives different solutions to (3.91). Then the approach is to solve the point interpolation problem and then to generate more control points through one subdivision iteration (2.12) in order to increase the degrees of freedom of the curve, setting $k = 1$ in (2.13). In this case, the rules (2.28) lead to

$$c(j) = V_j^0 = \tfrac{1}{6}P_{2j-1}^1 + \tfrac{4}{6}P_{2j}^1 + \tfrac{1}{6}P_{2j+1}^1 \quad \text{and} \quad c'(j) = V_i^1 = -P_{2j-1}^1 + P_{2j+1}^1, \tag{3.94}$$

for $j \in \mathbb{Z}$.

Therefore, setting $T_j = \frac{V_j^1}{\|V_j^1\|}$ as unitary vectors for all $j \in \mathbb{Z}$, the rules in (3.95) yield to the iterations

$$\begin{cases} Q_{2j}^{k+1} &= \frac{6}{4}V_j^0 - \frac{1}{4}Q_{2j-1}^k - \frac{1}{4}Q_{2j+1}^k, \\ Q_{2j+1}^{k+1} &= Q_{2j-1}^k + \|Q_{2j+1}^k - Q_{2j-1}^k\| T_j, \end{cases} \qquad j \in \mathbb{Z}, k \in \mathbb{N}, \tag{3.95}$$

where $Q_j^k \to P_j^1$ as $k \to \infty$ for all $j \in \mathbb{Z}$.

As initial values, we define $\mathbf{Q}^0 = \{Q_j^0 = P_j^1, j \in \mathbb{Z}\}$, where the set P^1 is computed from P^0 obtained previously in (3.91). In other words, we solve the point interpolation problem and the solution $\mathbf{P}^0$ provides the initial values $\mathbf{Q}^0 = \mathbf{P}^1$ (by a subdivision step) for the iterations in (3.95).

The formulation in (3.95) cannot be written as in (3.92) in terms of a linear iterative operator because the non-linear nature of (3.95). Nevertheless, the solution obtained by (3.95) can be obtained as solution of $\mathbf{M}_n\mathbf{P}^0 = \mathbf{U}^{(2)}$ for $\mathbf{M}_n$ as in (3.59) for $d = 2$ (i.e., point and tangent interpolation) and $\mathbf{U}^{(2)} = [V_0^0, V_0^1, V_1^0, V_1^1, \ldots, V_{n-1}^0, V_{n-1}^1]^\top$ with $V_j^1 = \lambda_j T_j$ and $\lambda_j \in \mathbb{R}$ unknown values, which can affect the geometry of the limit curve (see Fig. 1.7).

Remark 17. *The idea in [89] to perform a subdivision step in order to have enough information supposes that there is a curve with an initial polygon of n points. This is not always true when we are interpolating tangents, as the initial points determine uniquely the tangents vectors of the subdivision curve. In this case, the iterations in* (3.95) *are taking those points as initial approximations.*

Remark 18. *How can we obtain the same results as the Okaniwa et al. algorithm?*

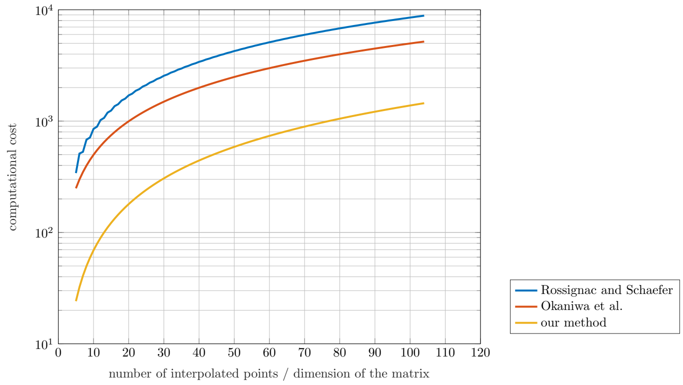


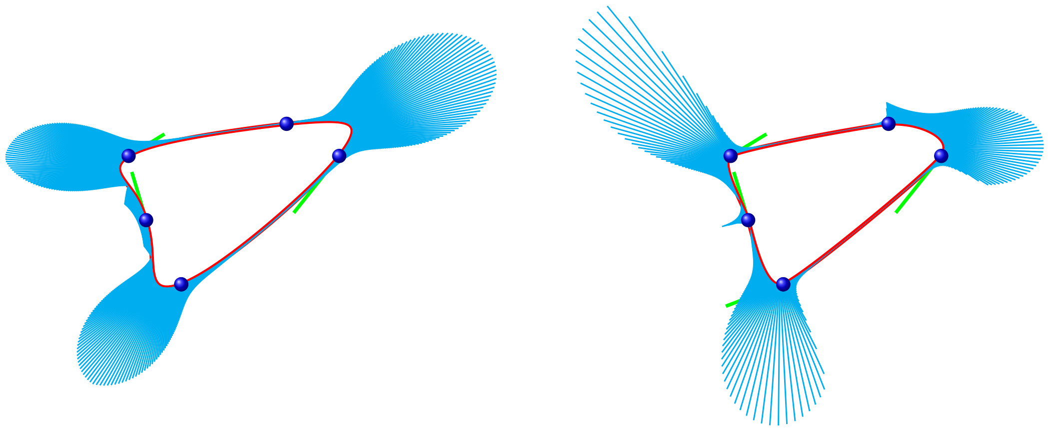
Figure 3.7: Comparing computational cost among the iterative methods related and the direct method proposed: $\mathcal{O}(n \cdot \#iterations)$ vs $\mathcal{O}(n \log n)$.

3.4.3 Selected Hermite subdivision schemes comparisons

Let us consider as a first example the cubic Hermite subdivision scheme proposed by [84], compared to the scalar scheme generating cubic B-spline curves. In the first case, only two points with their tangents are used each time for the Hermite subdivision rule and three in the scalar case. However, the Hermite scheme generates a spline curve with only C^1 continuity on the control points, while it is C^2 for each segment as a cubic curve. Meanwhile the scalar scheme is C^2 everywhere.

The curvature plot (see Fig. 3.8) makes evident the discontinuity jumps in curvature for the Merrien's interpolant at the control points, as expected from a C^1 spline curve. On the other hand, the scalar subdivision scheme with the same degree as a piecewise polynomial curve exhibit a smoother variation of the curvature. Considering the support, the subdivision rules for the Hermite scheme use only two control points, whereas the B-spline scheme uses three.

A possible drawback for scalar subdivision schemes is the need to compute control points to interpolate the data (points and associated derivatives). An interpolatory Hermite scheme has the advantage to generate in each iteration a denser sampling of the curve and its derivatives. Nevertheless, the Fourier factorization of the interpolating block-circulant matrix pays off the trade-off of smoothness versus computational cost, which is $\mathcal{O}(n)$ in the Hermite case and $\mathcal{O}(n \log n)$ in the scalar case.



(a) Cubic B-spline Hermite interpolant (b) Merrien's cubic Hermite scheme

Figure 3.8: Curvature comb comparison between a scalar subdivision scheme (cubic B-spline) and an Hermite subdivision scheme (Merrien's cubic spline).

Limit curves with C^2 continuity can be produced interpolating higher derivatives or with masks of larger support. However, in [95, 99] they generate spline curves of only C^2 continuity on the interpolated points even though locally the continuity is C^4 as quintic polynomials. The proposals in [73, 87] increase the support which is greater than the support for the one-parameter J-spline family, which can generate limit curves C^4 everywhere.

A comparison with an Hermite interpolant generating cubic B-splines with double knots [93], which are C^1 curves, produces the same trade-off. The vector subdivision scheme

$$V_s^{k+1} = \sum_{j\in\mathbb{Z}} A_{s-2j} V_j^k, \quad \text{with} A_{-1} = \begin{bmatrix} \frac{1}{8} & 0 \\ \frac{5}{16} & \frac{1}{16} \end{bmatrix}, A_0 = \begin{bmatrix} \frac{3}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{3}{8} \end{bmatrix}, A_1 = \begin{bmatrix} \frac{1}{16} & \frac{5}{16} \\ 0 & \frac{1}{8} \end{bmatrix}, \tag{3.96}$$

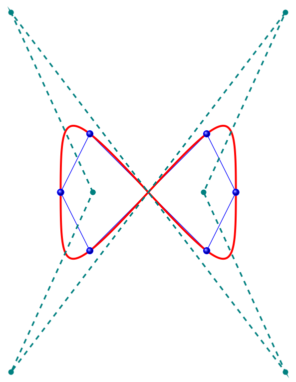
and $A_j = 0 \in \mathbb{R}^{2\times 2}\ \forall\ j \notin \{-1, 0, 1\}$, also generates double-knot cubic B-spline [49] which are C^1 curves as well and does not interpolate the initial data. We conclude that an inverse problem for the interpolation with that vector subdivision scheme does not benefit with respect to the scalar cubic B-spline subdivision scheme.

Chapter 4

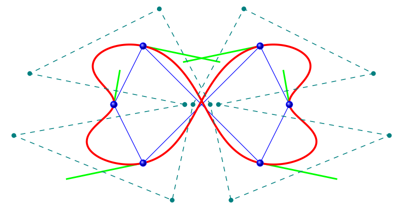
Interpolation in real-world applications

In the previous chapter a solution to the point interpolation and Hermite interpolation problem was proposed with any scalar linear uniform subdivision scheme once the exact evaluation of the basic function at certain parameters is known. For non stationary schemes, if the limit basic function in (2.53) is known, then the same procedure applies to define the interpolation operator. Then the analysis of the spectrum and the computation of the solution remain the same. With the choice of the parameters at which the curve interpolates the required data, a degree of freedom is added, although we lose the symmetry in the general case.

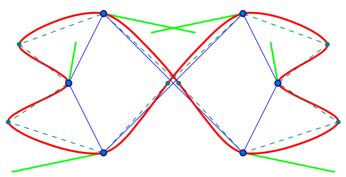
The following figures provide examples of free-form design with different schemes and the *Algorithm 5*, covering the cases of point interpolation and Hermite interpolation with primal and dual schemes. The same data points are used in each case with the freedom for the designer to change the tangent vectors if the shape of the point interpolant curve is not the desired one. The Fig. 4.0c shows an interpolant generated by an interpolatory scheme (for points interpolation) [50]. That scheme is not interpolatory for the Hermite problem as the rules in (1.11) do not consider the tangent information. However, we can solve the Hermite interpolation problem with the proposed algorithm.



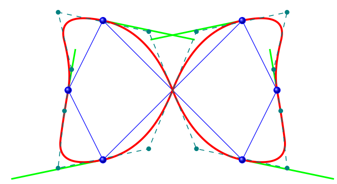
(a) Point interpolation with a primal scheme: quintic B-spline



(b) Hermite interpolation with a primal scheme: quintic B-spline



(c) Hermite interpolation with the interpolatory 4-point scheme in (1.11)



(d) Hermite interpolation with a dual scheme: Dyn, Floater, Horman in (3.21) for $\nu = \frac{1}{16}$

Figure 4.0: Free-form design of curve for given points and tangent vectors.

4.1 Offset curves

The computation of *offset curves* is one of the fundamental operations in CAD/CAM[1], *numerical controlled* (NC) machining, manufacturing, and designing in robotics. Specifically, for laser cutting machining and 2.5D pocket machining, the cutting tool path generation requires an offset (i.e., constant distance from the design contour) for the definition of tolerance regions.

The *offset* of a curve $c(t)$ is the set of all points that lie at constant perpendicular distance ρ from $c(t)$. We can represent this set by the curve $\Omega(\rho, c(t)) = c(t) + \rho n(t)$

[1] *Computer Aided Design/Computer Aided Manufacturing*

with $n(t)$ the unitary normal vector to the curve $c(t)$ for all t. For example, for a curve $c : \mathbb{R} \to \mathbb{R}^2$, parameterized as $c(t) = (x(t), y(t))$ the offset curve is defined as

$$\Omega(\rho, c(t)) = c(t) + \rho \frac{(y'(t), -x'(t))}{\sqrt{(x'(t)^2 + (y'(t)^2)}}. \tag{4.1}$$

Traditional techniques of NC tool path description use the so-called *G-code*, which uses piecewise linear (`G1` instruction), circular (`G2/G3` instructions) and cubic Bézier curve segments (`G5` instructions). The first two descriptions are inherit by their offsets. On the contrary, the offset of any cubic Bézier curve is not a curve of the same class[2], the same happens for many classes of curves. To overcome this situation, *Pythagorean Hodograph* (PH) curves have been proposed in the literature [57] as a closed class with respect to the offset operator. They form a subset of odd-degree polynomial Bézier curves.

Computing the offset for any curve is more challenging computational problem and there is an extensive literature on numerical approximation techniques for the offset of curves such as B-splines curves [91, 92, 51, 82, 109, 120]. Some of these methods aim to approximate the offset with a curve of the same class, such as Bézier and B-spline curves, with a different amount of control points. For subdivision schemes is not known in general a parameterization of the generated curve[3]. Therefore, the offset has to be computed from the generated set of points $c\left(\frac{1}{q}\mathbb{Z}\right)$, $q \in \mathbb{N}$.

An advantage of subdivision schemes for geometric modeling systems is the possibility to generate different *levels of resolution* of the subdivision curve (see Fig. 1.2). That *multiresolution* representation benefits the rendering of the curve for specified level-of-detail[4]. Even though the offset cannot be described by a set of control points, similarly to the subdivision curve, its exact computation can be done once the values $c\left(\frac{1}{q}\mathbb{Z}\right)$ and $c'\left(\frac{1}{q}\mathbb{Z}\right)$ are known.

Interpolatory Hermite subdivision schemes (see Fig. 1.6) benefit from their construction for the offset computation. We must observe that the control polygon $\widetilde{\mathbf{P}^0}$ with vertices at constant distance from an initial control polygon $\mathbf{P}^0$ does not provide the offset of the subdivision curve obtained from $\mathbf{P}^0$. Instead, for scalar subdivision

[2] *Tschirnhausen's* cubics, modulo rigid motions, scaling, and linear reparameterization, are the only cases to have a cubic Bézier offset.

[3] Indeed, this thesis uses the quadratic, cubic and quintic B-splines as examples because the results can be verified by using the known parametric representation in (1.2)

[4] This point is useful as well in the next section for image segmentation.

schemes, if the sampling of the basic function $\varphi\left(\frac{1}{q}\mathbb{Z}\right)$ is pre-computed, then the rendering of a subdivision curve (2.2) is an $\mathcal{O}(n)$ operation, with n the amount of control points. The offset computation is independent of the interpolatory or approximating nature of the scheme.

Nonetheless, it is possible to compute the exact offset for curves designed interpolating points and associated tangents (see Fig. 4.1). The parameterization of the offset curve is inherited from the subdivision curve. Even more, the computation of the sampling of the basic function allows to compute the exact radius of curvature. This information is useful for detecting singularities of the offset. For example, whether the offset radius ρ is less than the minimum radius of curvature of the curve[5].

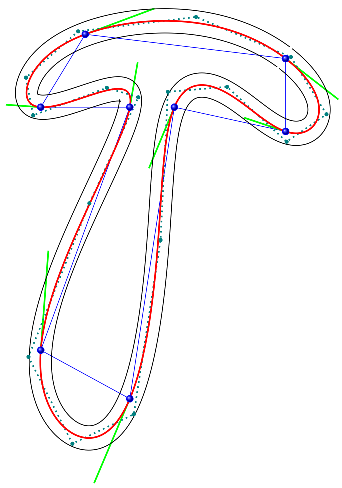
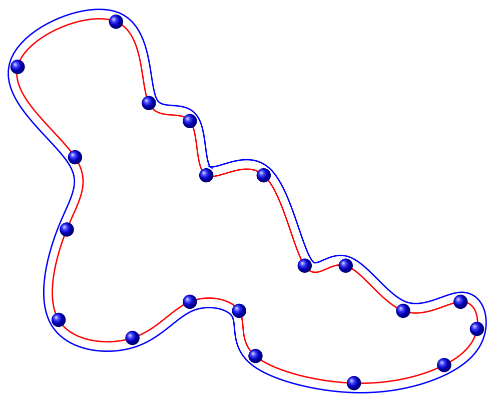


Figure 4.1: Offset of a C^3 subdivision curve generated with Tan-Zhuang-Zhang scheme [112] for the parameters $A = 1, B = \frac{1}{32}$.

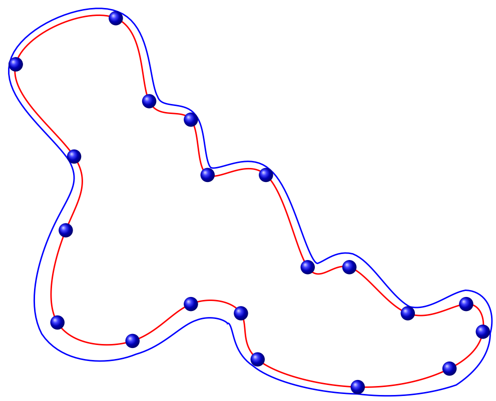
It is also possible to define an offset with variable distance $\rho(t)$ from the curve. In the cases of constant and variable distance, the outcome is not the same as to interpolate the set of points

$$\left\{ c(t_j) + \rho(t_j)\frac{(y'(t_j), -x'(t_j))}{\sqrt{(x'(t_j)^2 + (y'(t_j)^2)}}, j = 0, \ldots, n-1 \right\}. \tag{4.2}$$

[5]When this occurs in NC applications, the cutting tool performs unintended gouging.



(a) Constant offset



(b) Variable offset

Figure 4.2: Constant and variable offset of a subdivision curve (Varese's lake shape).

4.2 Snakes curves based on subdivision

This section is devoted to show another application of subdivision curves, beyond the free-form design of curves. The considered application is the image segmentation, where the boundary of an object is represented by a curve. Therefore, the curves used for that purpose are defined by using the already studied class of subdivision schemes.

The contour of an object in an image can be represented by a curve, instead of merely a set of digital points (or pixels). There are different approaches to accomplish this task. Among them, there are methods where a *snake*, that is a sequence of curves, converges to that boundary. Each curve in the sequence can be represented as a subdivision curve. The evolution of the snake is driven by its control points which are computed minimizing an energy that pushes the snake towards the boundary of the region of interest.

Active contours, or snakes, were introduced by Kass et al. in [74] as curves that slither within an image from some initial position towards the contour of the object of interest. Snakes have become popular in segmentation and tracking applications [8], [30] since they are very flexible and efficient.

The evolution of the snake is formulated as a minimization problem and the corresponding objective functions is usually known as snake energy. During the optimization process, the snake is iteratively updated from a starting position until it reaches the minimum of the energy function. This energy measures the distance between the snake and the boundary of the object. It also controls some desirable properties of the final snake, such as the smoothness and the interpolation of distinguished points. The quality of segmentation is determined by the choice of the energy terms.

Kass et al. [74] originally formulated the snake energy as a linear combination of three terms: the *image energy*, which only depends on the image, the *internal energy*, which ensures the smoothness of the snake, and the *constraint energy*, which allows that the user interacts with the snake. The specific definition of these energies depends on the application, on the nature of the image, and on the representation of the snake. The image energy guides the snake to the boundary of the interest object and is the most important energy. It is usually defined as a weighted sum of a gradient based energy [74], [110], that provides a good approximation of the contour of the object, and a region based energy [58], [113], that distinguishes different homogeneous regions within the image. Gradient based energies have a narrow zone of attraction in comparison with region based energies. Hence, the success of the segmentation depends on the selection of the weight.

Snakes differ not only in the choice of the energy function, but also in the representation of the curve. According to the representation, snakes may be classified as point snakes [74], geodesic snakes [10, 119, 2, 119], and parametric snakes [58, 7, 108, 28]. Point-snakes simply consist of an ordered collection of points. This representation depends on a large number of parameters (the snake points) which makes the optimization expensive. Geodesic snakes are described as the zero level set of a higher-dimensional manifold. This type of active contours is very flexible topologically. As a result, it is suitable for segmenting objects that have variable shapes. A drawback of geodesic snakes is that they are expensive from a computational point of view. Parametric snakes are smooth curves written as a linear combination of a basis of functions. The coefficients in this representation, known as control points, are few. This speeds up the optimization process. The downside of parametric snakes is that the parametrization restricts the shapes that can be approximated.

In this section we focus on a particular class of parametric snakes: those generated from a subdivision scheme. Subdivision curves describe a contour by an initial discrete and finite set of control points which, by the iterative application of refinement rules, becomes continuous in the limit. Depending on the choice of the subdivision mask, the continuous limit curve may have different degrees of smoothness. The main advantages of subdivision schemes are their simplicity of implementation, the possibility to control their order of approximation, and their multiresolution property which provides representations of the contour of a shape with varying resolutions.

4.2.1 Related work

The use of subdivision curves for segmentation was first proposed in [66], where the so-called *tamed snake* is introduced. This snake is generated by the classical four-point subdivision scheme [50]. The method incorporates image information considering the control points of the subdivision curve as mass points attracted by edges of the image. The four point subdivision scheme is also used in [108], in combination with the gradient vector flow. After every step of subdivision, the region energy of the subdivision polygon is reversely computed and a local adaptive compensation is carried out in such a way that regions with high curvature are further subdivided, while flat regions remain unrefined.

In the context of image segmentation the most common snakes based on subdivision schemes are those producing B-spline type curves [7], [58], [69], [31]. In [7] the snake is represented by cubic B-spline basis functions. The initial B-spline is specified choosing node points instead of the B-spline control points to provide a more intuitive user-interaction. To improve optimization speed and robustness, a multiresolution approach is selected. This approach, based on an image pyramid, starts applying the optimization procedure at the coarsest level on a small version of the image. After convergence, this solution is then used as the starting condition for the next finer level.

In [83] a segmentation method called *SketchSnakes* is proposed. The method combines a general subdivision curve snake with an initialization process based on a few sketch lines drawn by the user across the width of the target object. External image forces are computed at the points of the finer level curve and then distributed using weights derived from the original subdivision rules among the points of the coarse level. The positions of the control points are updated, new external forces are calculated, and the process is repeated until an accurate solution is reached.

As of late, exponential B-spline have been introduced for constructing snakes that reproduce circular and elliptical shapes [28], [29], [31]. In [3] subdivision snakes are obtained in a generic way using a multiscale approach to speed up the optimization process and improve robustness. Depending on the selected admissible subdivision mask, the snake may be interpolatory or reproduce trigonometric or polynomial curves. The multiscale approach facilitates the increase of the number of points describing the curve as the algorithm progresses to the solution and, at each step, the scale of the image feature is matched to the density of the sample of the curve.

4.2.2 Snake Energies

In the literature, the evolution of the snake is driven by the minimization of several energies. One measures the proximity between the snake and the boundary $\partial\Gamma$ of a bounded region Γ in a digital image. Others measure desirable properties of the final curve including the smoothness and the interpolation of distinguished points.

Since our snake is a subdivision curve, the total energy E_{snake}, depends on the initial control polygon $\mathbf{P}^0$. The control polygon $\mathbf{P}^0_*$ of the optimal snake is computed as:

$$\mathbf{P}^0_* = arg \min_{\mathbf{P}^0} E_{\text{snake}}(\mathbf{P}^0). \tag{4.3}$$

In the current approach, we assume that the region of interest Γ to be segmented is dark in comparison to the background. Hence, the energy functionals related with the image are designed to detect dark objects on a brighter background. All the energies are defined by integrals of functions which are computed approximately. To obtain good approximations, we use a large sample of points on the subdivision curve. In the following sections we develop the expressions for each energy.

Remark 19. *The images are represented in system of coordinates defined by rows and columns, like the indexing of a matrix. Thus, as a convention when we say* (x, y)*, the* x*-coordinate refers to the row and the* y*-coordinate refers to the column (see for example Fig. 4.3). This does not affect the definition and use of the subdivision scheme, as each coordinate in* (2.2) *works independently.*

Gradient energy

Let $I(x, y)$ denotes the image intensity at a pixel with coordinates (x, y). If $\mathbf{r}(t) = (x(t), y(t))$ with $t \in [0, n]$ is a subdivision curve defined by n control points, then the simplest image energy considered is the *gradient magnitude energy* E_{mag} given by

$$E_{mag}(\mathbf{r}(t)) = -\int_0^M \|\nabla I(\mathbf{r}(t))\|^2 \ dt, \tag{4.4}$$

where $\|\nabla I(\mathbf{r}(t))\|^2 = \left(\frac{\partial I}{\partial x}(x(t), y(t))\right)^2 + \left(\frac{\partial I}{\partial y}(x(t), y(t))\right)^2$. Since the gradient magnitude energy only depends on the magnitude of the gradient vector, the minimization of (4.4)) can misguide the snake to a neighboring object if the initial approximation is not very close to the boundary of interest. To overcome this limitation, several alternatives energies have been proposed, like balloon forces [16], gradient vector-fields [117, 68, 69], or multiresolution approaches [7].

In the following we use the gradient-based image energy E_{grad} proposed in [68]. The idea behind this approach is described in the subsequent lines. If we travel around the ground truth boundary curve $\partial\Gamma$ in a counterclockwise direction, then Γ is always on the "left", i.e in the direction of $-\nabla I$. Hence, we pull the snake in the direction of $\partial\Gamma$, requiring the normal to the snake at any point to be parallel to $-\nabla I$ at the same point. Specifically, if we denote by $\mathbf{n}(t)$ the inward unit normal to snake at the point $\mathbf{r}(t)$, then the new energy E_{grad}, which takes into account not only the magnitude of the image gradient but also its direction, is given by

$$E_{grad}(\mathbf{r}(t)) = -\int_0^M \langle \nabla I(\mathbf{r}(t)), \left\| \frac{\mathrm{d}\mathbf{r}(t)}{\mathrm{d}t} \right\| \mathbf{n}(t) \rangle \mathrm{d}t, \tag{4.5}$$

where $\langle \cdot, \cdot \rangle$ is the usual scalar product and $\frac{\mathrm{d}\mathbf{r}(t)}{\mathrm{d}t}$ denotes the tangent to $\mathbf{r}(t)$. Expanding (4.5) we obtain

$$E_{grad}(\mathbf{r}(t)) = -\int_0^M \left(\frac{\partial I}{\partial x}(x(t), y(t)) \frac{\mathrm{d}y(t)}{\mathrm{d}t} - \frac{\partial I}{\partial y}(x(t), y(t)) \frac{\mathrm{d}x(t)}{dt} \right) \mathrm{d}t. \tag{4.6}$$

To compute good approximations of the energies (and their derivatives with respect to the coordinates of control points), we use a large sample of points on the subdivision curve. More precisely, given the initial polygon $\mathbf{P}^0 = \{P_0^0, \ldots, P_{M-1}^0\}$, we select q (in our experiments we take $q \in [2^4, 2^5]$) and we use (2.2) to generate $q\ M$ points $\mathbf{r}(j/q)$, $j = 0, \ldots, q\ M - 1$ on the subdivision curve. We apply bilinear interpolation on the gradient of the image to compute $\nabla I(\mathbf{r}(j/q))$. Finally, we approximate the energy substituting the integral in (4.6) by the average of values of the integrand over the sample of $q\ M$ points on the subdivision curve corresponding to parameter values $\frac{j}{q}$, $i = 0, \ldots, q\ M - 1$.

Considering (2.2), we obtain[6] the following approximation of (4.6)

$$E_{grad}(\mathbf{P}^0) \approx \frac{1}{q\,M} \sum_{i=0}^{q\,M-1} \left[\frac{\partial I}{\partial y} \left(\sum_{j\in\mathbb{Z}} P_j^0 \varphi(\tfrac{i}{q} - j) \right) \mathbf{t}_{x}{}_i^q - \frac{\partial I}{\partial x} \left(\sum_{j\in\mathbb{Z}} P_j^0 \varphi(\tfrac{i}{q} - j) \right) \mathbf{t}_{y}{}_i^q \right], \tag{4.7}$$

where $\mathbf{t}_{x}{}_i^q = \frac{\mathrm{d}x}{\mathrm{d}t}(i/q)$ and $\mathbf{t}_{y}{}_i^q = \frac{\mathrm{d}y}{\mathrm{d}t}(i/q)$, so that $(\mathbf{t}_{x}{}_i^q, \mathbf{t}_{y}{}_i^q) = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}(i/q)$.

It should be noticed that the right-hand side of (4.7) is a function of the coordinates of the initial control points $\mathbf{P}^0$. The sampling $\varphi(\mathbb{Z}/q)$ is precomputed, as well as the sampling $\frac{\mathrm{d}}{\mathrm{d}t}\varphi(\mathbb{Z}/q)$ used to computed the gradient for each energy.

Region energy

The main limitation of gradient based energy (4.5) is that its zone of attraction is limited, since the gradient is small as long as we move away from $\partial\Gamma$. To address this problem several region energies have been introduced in the literature [110, 13, 12, 113, 28]. Some use statistical information to identify different regions [68, 69, 113]. Inspired by the energies proposed in [28] and [113], we introduce simple region energy E_{reg} in our setting. The considered energy is designed to maximize the contrast between the average intensity of the pixels within the snake and the average intensity in the region outside the snake and inside a given bounding box.

Assuming that Ω, the region enclosed by the snake $(x(t), y(t))$, $t \in [0, M]$, is contained in a rectangular region R, we denote by $|R|$ the area of R (which is a constant) and by $|\Omega|$ the area of Ω (which may vary, while the snake evolves). The new region energy, E_{reg}, to be minimized is reported below

$$E_{reg}(\mathbf{P}^0) := -\left(\frac{\int\int_\Omega I(x,y)\mathrm{d}x\mathrm{d}y}{|\Omega|} - \frac{\int\int_{R\setminus\Omega} I(x,y)\mathrm{d}x\mathrm{d}y}{|R| - |\Omega|} \right)^2. \tag{4.8}$$

We observe that minimizing E_{reg} is equivalent to maximize the difference between the average intensity inside Ω and the average intensity in the complement of Ω in R.

Let introduce the following notation

$$I_\Omega := \int\int_\Omega I(x,y)\mathrm{d}x\mathrm{d}y \qquad \text{and} \qquad I_R := \int\int_R I(x,y)\mathrm{d}x\mathrm{d}y.$$

[6]Recall that the indices of the inner summations depend on the choice of the subdivision scheme and its support.

In accordance with the definition of I_Ω and I_R, the region energy may be written as

$$E_{reg}(\mathbf{P}^0) = -\left(\frac{I_\Omega}{|\Omega|} - \frac{I_R - I_\Omega}{|R| - |\Omega|}\right)^2. \qquad (4.9)$$

Since region energies are usually expressed as integrals of a function over the domain Ω enclosed by the snake, some authors propose the use of Green's theorem to rewrite the 2D integrals as a line integral along the snake [28, 31, 69]. If we apply it to the function $I(x, y)$, then we obtain

$$I_\Omega = \int\int_\Omega I(x,y)\mathrm{d}x\mathrm{d}y = \int_{\partial\Omega} I_1(x,y)\mathrm{d}y = -\int_{\partial\Omega} I_2(x,y)\mathrm{d}x, \qquad (4.10)$$

where

$$I_1(x,y) = \int_{-\infty}^{x} I(\tau, y)\mathrm{d}\tau \quad \text{and} \quad I_2(x,y) = \int_{-\infty}^{y} I(x,\tau)\mathrm{d}\tau. \qquad (4.11)$$

Thus, if $\partial\Omega$ is parametrized by $\mathbf{r}(t) = (x(t), y(t)),\ 0 \le t \le M$, then from (4.10) we deduce

$$I_\Omega = \int_0^M I_1(x(t), y(t))\,\frac{\mathrm{d}y(t)}{\mathrm{d}t}\mathrm{d}t = -\int_0^M I_2(x(t), y(t))\,\frac{\mathrm{d}x(t)}{\mathrm{d}t}\mathrm{d}t. \qquad (4.12)$$

This approach reduces significantly the computational cost, but in our experiments we have found that large errors may be introduced when we use it to compute the integrals in (4.8), in the context of digital images.

In those works, the line integrals (4.11) are approximated using a sample of points on the snake and sum up the contributions of column or row image pixel strips corresponding to each point on the snake. However, even if the snake is parametrized by a multiple of the arc length, the distribution on the image of the sample of points may be very irregular. For instance, if the image has low resolution then some points may belong to the same pixel overestimating the value of the integral. On the contrary, if the image has high resolution then those rows or columns of Ω without any point of the sample do not contribute to the computation and produce an underestimate of the integral.

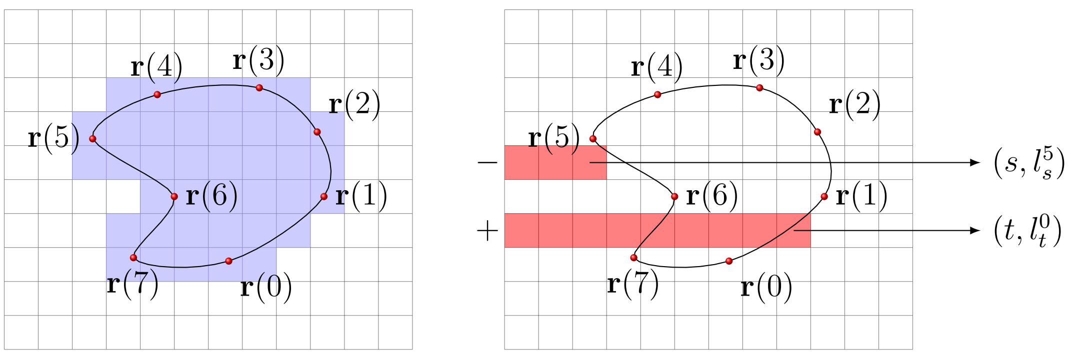
Nevertheless, we use (4.12) to compute later the partial derivatives (in *Section 4.2.3*).

Instead, we propose a sort of *rasterization* of $\partial\Omega$ to describe it and then we compute (4.8) with the pixels in Ω and their values of intensity (see Fig. 4.7). It should be noted that the subdivision curve $\mathbf{r}(t)$ is represented as a polygon with vertices in $\{\mathbf{r}(i/q), i = 0, \ldots, q\ M - 1\}$ living on the image. Then, it can be observed that $\mathbf{r}(i/q) = (x_i^q, y_i^q)$ is represented on the image by the pixel with coordinates $(\lceil x_i^q \rceil, \lceil y_i^q \rceil)$.

According to the previous analysis (detailed in *Section 4.2.4*), the integral of the intensity may be computed approximately summing up (with sign) the contribution of each horizontal image strip intersected by Ω, see Figure 4.3. The value l_j^i is the index of the column of the pixel that results from the intersection of the edge $[\mathbf{r}(i/q), \mathbf{r}((i+1)/q)]$ with the j-th row of the image. Therefore,

$$I_\Omega = \int\int_\Omega I(x,y)\ dx\ dy \approx \sum_{i=0}^{q\ M-1} \operatorname{sign}(x_i^q - x_{i+1}^q) \sum_{j=\lceil x_i^q \rceil}^{\lceil x_{i+1}^q \rceil} \sum_{l=1}^{l_j^i} I(l,j). \tag{4.13}$$

The values of $\sum_{l=1}^{l_j^i} I(l,j)$ can be pre-computed in a cumulative table for speeding up the implementation.



(a) Representing Ω (b) The strips used to approximate the integral in E_{reg}

Figure 4.3: Pixels considered to compute the *region-based energy*. Observe that pixel's coordinates are in the coordinate system of the image *(row, column)*.

In particular, the approximation of the area of Ω enclosed by the subdivision curve is

$$|\Omega| = \int\int_\Omega\ dx\ dy \approx \sum_{i=0}^{q\ M-1} \operatorname{sign}(x_i^q - x_{i+1}^q) \sum_{j=\lceil x_i^q \rceil}^{\lceil x_{i+1}^q \rceil} l_j^i. \tag{4.14}$$

Finally, both approximations (4.13) and (4.14) are substituted in (4.9) in view of providing the approximation of the region energy.

Curvilinear parametrization energy

In the literature [74, 69], the so-called *internal energy* is introduced to guarantee the smoothness of the snake. It usually combines the length of the snake and its curvature. Since the curves generated by some schemes do have not continuous curvature, our internal energy is reduced to the *curvilinear parametrization energy* defined as

$$E_{curv}(\mathbf{r}(t)) = \int_0^M \left(\left\| \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{r}(t) \right\|^2 - c \right)^2 \mathrm{d}t \tag{4.15}$$

where $c = \frac{L^2}{M^2}$ and L denotes the arc lengths of the snake for $t \in [0, M]$.

Recall that minimizing (4.15) is equivalent to requiring that the parametrization of the snake is close to being a multiple of the arc length, but our snake is a subdivision curve parametrized uniformly. Hence, when we are minimizing (4.15), we are in fact looking for the control points of a subdivision curve for which the uniform parametrization is approximately a multiple of the arc length parametrization. In the optimum, the arc length of the snake between any pair of consecutive points $\mathbf{r}(i), \mathbf{r}(i+1)$ would be constant. The same happen with points $\mathbf{r}(i/q), i = 0, \ldots, q\,M-1$, since these points correspond to uniform parameters values $\frac{i}{q}$. The uniform distribution of points $\mathbf{r}(i/q)$ ensures that the polygonal curve defined by them is a good approximation of the subdivision curve. This is very convenient, because points $\mathbf{r}(i/q)$ are used to approximate the gradient and region energies, respectively, along with their partial derivatives. Finally, it is important to point out that the parametrization energy avoids that control points P_j^0 could be accumulated at some regions producing corners and other artifacts on the subdivision curve.

To compute approximately the parametrization energy E_{curv} we use the tangent vectors at the parameter values $\frac{i}{q},\ i = 0, \ldots, q\ M-1$. From (2.14) we obtain the following *approximate* expressions for E_{curv} as function of the control polygon $\mathbf{P}^0$

$$E_{curv}(\mathbf{P}^0) \approx \sum_{i=0}^{q\ M-1} ((\mathbf{t}_{x_i}^{\ q})^2 + (\mathbf{t}_{y_i}^{\ q})^2 - c)^2, \tag{4.16}$$

where $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{r}(i/q) = (\mathbf{t}_{x_i}^{\ q}, \mathbf{t}_{y_i}^{\ q})$.

4.2.3 Optimization

To obtain the optimal position of the control points of the snake, we minimize the total energy given by

$$E_{\text{snake}}(\mathbf{P}^0) = \alpha_1 E_{grad}(\mathbf{P}^0) + \alpha_2 E_{reg}(\mathbf{P}^0) + \alpha_3 E_{curv}(\mathbf{P}^0), \quad , \alpha_j \in \mathbb{R}, j \in \{1,2,3\}. \tag{4.17}$$

The optimization problem is solved using the BFGS Quasi-Newton method with a cubic line search procedure. This method requires the gradient of the snake energy with respect to the variables of our problem which are the coordinates (x_j^0, y_j^0) of the control points $\mathbf{P}_j^0$, $j = 0, \dots, M-1$. In this section we give the expressions of the approximations of partial derivatives of each energy with respect to each coordinate x_j^0 and y_j^0.

For further details see [37].

Derivatives of gradient energy

Deriving directly in (4.6) with respect to x_j^0 we obtain

$$\begin{aligned} \frac{\partial E_{grad}}{\partial x_j^0} = & -\int_0^M \left(\left(\frac{\partial^2 I}{\partial x^2} \frac{\partial x}{\partial x_j^0} + \frac{\partial^2 I}{\partial x \partial y} \frac{\partial y}{\partial x_j^0} \right) \frac{\mathrm{d}y(t)}{\mathrm{d}t} + \frac{\partial I}{\partial x} \frac{\partial \left(\frac{\mathrm{d}y}{\mathrm{d}t} \right)}{\partial x_j^0} \right) \mathrm{d}t \\ & + \int_0^M \left(\left(\frac{\partial^2 I}{\partial x \partial y} \frac{\partial x}{\partial x_j^0} + \frac{\partial^2 I}{\partial y^2} \frac{\partial y}{\partial x_j^0} \right) \frac{\mathrm{d}x(t)}{\mathrm{d}t} + \frac{\partial I}{\partial y} \frac{\partial \left(\frac{\mathrm{d}x}{\mathrm{d}t} \right)}{\partial x_j^0} \right) \mathrm{d}t. \end{aligned} \tag{4.18}$$

Taking into account that $y(t)$ and $\frac{\mathrm{d}y(t)}{\mathrm{d}t}$ do not depend on x_j^0, from (4.18), we obtain

$$\frac{\partial E_{grad}}{\partial x_j^0} = \int_0^M \left(\left[\frac{\partial^2 I}{\partial x \partial y} \frac{\partial x}{\partial t} - \frac{\partial^2 I}{\partial x^2} \frac{\mathrm{d}y}{\mathrm{d}t} \right] \frac{\partial x}{\partial x_j^0} + \frac{\partial I}{\partial y} \frac{\partial \left(\frac{\mathrm{d}x}{\mathrm{d}t} \right)}{\partial x_j^0} \right) \mathrm{d}t. \tag{4.19}$$

From (2.2) and (2.14) it follows that:

$$\frac{\partial x(t)}{\partial x_j^0} = \varphi(t-j), \quad \frac{\partial y(t)}{\partial y_j^0} = \varphi(t-j), \tag{4.20}$$

$$\frac{dx(t)}{dt} = \sum_{j \in \mathbb{Z}} x_j^0 \varphi'(t-j), \quad \frac{\mathrm{d}y(t)}{\mathrm{d}t} = \sum_{j \in \mathbb{Z}} y_j^0 \varphi'(t-j), \tag{4.21}$$

where φ' denotes $\frac{\mathrm{d}\varphi}{\mathrm{d}t}$. Evaluating the last expressions at $t = \frac{i}{q}$, $i = 0, \dots, q\,M-1$, we deduce

$$\left.\frac{\partial x(t)}{\partial x_j^0}\right|_{t=\frac{i}{q}} = \varphi\left(\tfrac{i}{q}-j\right) = \left.\frac{\partial y(t)}{\partial y_j^0}\right|_{t=\frac{i}{q}}, \tag{4.22}$$

$$\left.\frac{\partial\left(\frac{dx}{dt}\right)}{\partial x_j^0}\right|_{t=\frac{i}{q}} = \varphi'\left(\tfrac{i}{q}-j\right) = \left.\frac{\partial\left(\frac{dy}{dt}\right)}{\partial y_j^0}\right|_{t=\frac{i}{q}}. \tag{4.23}$$

Substituting the integral in (4.19) by the average of the integrand evaluated in the parameter values $i/q, \quad i = 0, \ldots, q\ M-1$ and using (4.22) and (4.23), we obtain the following approximation for the partial derivative of gradient energy with respect to x_j^0, that is

$$\frac{\partial E_{grad}}{\partial x_j^0} \approx \frac{1}{q\ M}\sum_{i=0}^{q\ M-1}\left(\frac{\partial^2 I}{\partial x \partial y}\left(\mathbf{r}\left(\tfrac{i}{q}\right)\right)t_{ix}^q - \frac{\partial^2 I}{\partial x^2}\left(\mathbf{r}\left(\tfrac{i}{q}\right)\right)t_{iy}^q\right)\varphi\left(\tfrac{i}{q}-j\right) + \frac{1}{qM}\sum_{i=0}^{q\ M-1}\frac{\partial I}{\partial y}\left(\mathbf{r}\left(\tfrac{i}{q}\right)\right)\varphi'\left(\tfrac{i}{q}-j\right). \tag{4.24}$$

Proceeding in a similar way, if we derive (4.6) with respect to y_j^0 and we take into account that $x(t)$ and $\frac{dx(t)}{dt}$ do not depend on x_j^0, then we obtain the expression for $\frac{\partial E_{grad}}{\partial y_j^0}$

$$\frac{\partial E_{grad}}{\partial y_j^0} = \int_0^M\left(\left(\frac{\partial^2 I}{\partial y^2}\frac{\partial x}{\partial t} - \frac{\partial^2 I}{\partial x \partial y}\frac{dy}{dt}\right)\frac{\partial y}{\partial y_j^q} + \frac{\partial I}{\partial x}\frac{\partial\left(\frac{dy}{dt}\right)}{\partial y_j^q}\right)\mathrm{d}t. \tag{4.25}$$

Discretizing the integral with the same procedure, from (4.25), we obtain the following approximation for the partial derivative of gradient energy with respect to y_j^0

$$\frac{\partial E_{grad}}{\partial y_j^0} \approx \frac{1}{q\ M}\sum_{i=0}^{q\ M-1}\left(\frac{\partial^2 I}{\partial y^2}\left(\mathbf{r}\left(\tfrac{i}{q}\right)\right)t_{ix}^q - \frac{\partial^2 I}{\partial x \partial y}\left(\mathbf{r}\left(\tfrac{i}{q}\right)\right)t_{iy}^q\right)\varphi\left(\tfrac{i}{q}-j\right) + \frac{1}{qM}\sum_{i=0}^{q\ M-1}\frac{\partial I}{\partial x}\left(\mathbf{r}\left(\tfrac{i}{q}\right)\right)\varphi'\left(\tfrac{i}{q}-j\right). \tag{4.26}$$

Derivatives of region energy

In order to find the optimal control polygon we have to compute the partial derivatives of E_{reg} with respect to the coordinates (x_j^0, y_j^0) of the control points $\{\mathbf{P}_j^0,\ j = 0, \ldots, M-$

1}. Since I_R is constant, from (4.9) we obtain

$$\frac{\partial E_{reg}}{\partial x_j^0} = -2D\left(\frac{\partial A}{\partial x_j^0} - \frac{\partial B}{\partial x_j^0}\right), \tag{4.27}$$

where

$$A := \frac{I_\Omega}{|\Omega|}, \quad B := \frac{I_R - I_\Omega}{|R| - |\Omega|} \quad \text{and} \quad D := A - B.$$

Recalling that

$$\frac{\partial E_{reg}}{\partial x_j^0} = -2D\left(\frac{\partial A}{\partial x_j^0} - \frac{\partial B}{\partial x_j^0}\right), \tag{4.28}$$

where

$$A := \frac{I_\Omega}{|\Omega|}, \tag{4.29}$$

$$B := \frac{I_R - I_\Omega}{|R| - |\Omega|}, \tag{4.30}$$

$$D := A - B, \tag{4.31}$$

and deriving directly in (4.29) and (4.30), we find

$$\frac{\partial A}{\partial x_j^0} = \frac{1}{|\Omega|}\frac{\partial I_\Omega}{\partial x_j^0} - \frac{I_\Omega}{|\Omega|^2}\frac{\partial |\Omega|}{\partial x_j^0}, \tag{4.32}$$

$$\frac{\partial B}{\partial x_j^0} = \frac{-1}{|R| - |\Omega|}\frac{\partial I_\Omega}{\partial x_j^0} + \frac{I_R - I_\Omega}{(|R| - ||\Omega|)^2}\frac{\partial |\Omega|}{\partial x_j^0}. \tag{4.33}$$

Substituting (4.32) and (4.33) in (4.28), we have

$$\frac{\partial E_{reg}}{\partial x_j^0} = -2D\left[\left(\frac{1}{|\Omega|} + \frac{1}{|R| - |\Omega|}\right)\frac{\partial I_\Omega}{\partial x_j^0} - \left(\frac{I_\Omega}{|\Omega|^2} + \frac{I_R - I_\Omega}{(|R| - |\Omega|)^2}\right)\frac{\partial |\Omega|}{\partial x_j^0}\right]. \tag{4.34}$$

Now we compute the partial derivatives involved in (4.34) using the Green Theorem as stated in (4.10) and (4.11).

Deriving in the first equality of (4.10) it follows that

$$\frac{\partial I_\Omega}{\partial x_j^0} = -\int_0^M \frac{\partial I_1}{\partial x}\frac{\partial x(t)}{\partial x_j^0} y'(t)\mathrm{d}t.$$

According to Leibniz's rule in (4.11) (for differentiation under the integral sign), taking into account that $\frac{\partial I_1}{\partial x} = I(x(t), y(t))$ and $\frac{\partial x(t)}{\partial x_j^0} = \varphi(t - j)$, from the previous expression we obtain

$$\frac{\partial I_\Omega}{\partial x_j^0} = -\int_0^M I(\mathbf{r}(t))\varphi(t-j)y'(t)\,\mathrm{d}t. \tag{4.35}$$

Since $|\Omega| = \int\int_\Omega \mathrm{d}x\mathrm{d}y$ from (4.35) it is clear that

$$\frac{\partial|\Omega|}{\partial x_j^0} = -\int_0^M \varphi(t-j)y'(t)\,\mathrm{d}t. \tag{4.36}$$

Finally, substituting (4.35) and (4.36) in (4.34) and grouping similar terms, we obtain

$$\frac{\partial E_{reg}}{\partial x_j^0} = -2D\int_0^M \left[G - H\,I(\mathbf{r}(t))\right]\varphi(t-j)y'(t)\mathrm{d}t,$$

where

$$G := \frac{I_\Omega}{|\Omega|^2} + \frac{I_R - I_\Omega}{(|R|-|\Omega|)^2} \quad \text{and} \quad H := \frac{1}{|\Omega|} + \frac{1}{|R|-|\Omega|}.$$

Proceeding in a similar way and deriving in the second equality of (4.10), it is easy to check that

$$\frac{\partial E_{reg}}{\partial y_j^0} = 2D\int_0^M \left(G - H\,I(\mathbf{r}(t))\right)\varphi(t-j)x'(t)\mathrm{d}t. \tag{4.37}$$

In practice, we approximate (4.2.3) and (4.37) by

$$\frac{\partial E_{reg}}{\partial x_j^0} \approx -\frac{2\widetilde{D}}{q\;M}\sum_{i=0}^{q\;M-1}\left[\widetilde{G} - \widetilde{H}I\left(\mathbf{r}\left(\tfrac{i}{q}\right)\right)\right]\varphi\left(\tfrac{i}{q}-j\right)y'\left(\tfrac{i}{q}\right),$$

$$\frac{\partial E_{reg}}{\partial y_j^0} \approx \frac{2\widetilde{D}}{q\;M}\sum_{i=0}^{q\;M-1}\left[\widetilde{G} - \widetilde{H}I\left(\mathbf{r}\left(\tfrac{i}{q}\right)\right)\right]\varphi\left(\tfrac{i}{q}-j\right)x'\left(\tfrac{i}{q}\right),$$

where $\widetilde{D}, \widetilde{G}$, and $\widetilde{H}$ denote the approximations of D, G, and H respectively, obtained from the approximated values of I_R, I_Ω, $|\Omega|$, and $|R|$ in (4.13) and (4.14).

Derivatives of the curvilinear parametrization energy

Deriving (4.15) with respect to x_i^q, we obtain

$$\frac{\partial E_{curv}}{\partial x_j^q} = 4\int_0^M \left(\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2 - \;c\right)\left(\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)\frac{\partial\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)}{\partial x_j^q} + \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)\frac{\partial\left(\frac{dy}{\mathrm{d}t}\right)}{\partial x_j^q}\right).$$

Taking into account that $y(t)$ does not depend on x_j^q and consequently $\frac{\partial\left(\frac{\mathrm{d}y(t)}{\mathrm{d}t}\right)}{\partial x_j^q} = 0$, the previous expression is reduced to

$$\frac{\partial E_{curv}}{\partial x_j^q} = 4 \int_0^M \left(\left(\frac{\mathrm{d}x}{\mathrm{d}t} \right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}t} \right)^2 - c \right) \left(\left(\frac{\mathrm{d}x}{\mathrm{d}t} \right) \frac{\partial \left(\frac{\mathrm{d}x}{\mathrm{d}t} \right)}{\partial x_j^q} \right). \tag{4.38}$$

To approximate (4.38) we evaluate the integrand in the parameter values i/q, $i = 0, \ldots, q\ M - 1$. Also taking into account (4.23) and (2.14), we obtain the following approximation for the partial derivative of curvilinear parametrization energy with respect to x_j^q, that is,

$$\frac{\partial E_{curv}}{\partial x_j^q} \approx \frac{4}{q\ M} \sum_{i=0}^{q\ M-1} ((\mathbf{t}_{x_i^q})^2 + (\mathbf{t}_{y_i^q})^2 - c)(\mathbf{t}_{x_i^q} \varphi_\omega'(i-j)). \tag{4.39}$$

Proceeding in a similar way, it is straightforward to check that the following approximation of the partial derivative with respect to y_j^q is obtained

$$\frac{\partial E_{curv}}{\partial y_j^q} \approx \frac{4}{q\ M} \sum_{i=0}^{q\ M-1} ((\mathbf{t}_{x_i^q})^2 + (\mathbf{t}_{y_i^q})^2 - c)(\mathbf{t}_{y_i^q} \varphi_\omega'(i-j)). \tag{4.40}$$

4.2.4 Implementation

In this section we give some details concerning the computation of the energies previously introduced. Additionally, we describe the main features of the application SubdivisionSnake which to compute the subdivision snakes produced by linear subdivision curves.

For the implementation of energies, it is necessary to define how to compute the gradient of an image in a point in an image, the area enclosed by a curve, and other details. In the following section we cover these topics.

Gradient energy

An image I is a discretization of the continuous space with integral coordinates. The gradient $\nabla I(x, y)$, of the image in the point $(x, y) \in \mathbb{R}^2$ in (4.7), is approximated using bilinear interpolation

$$\begin{aligned} \nabla I(x, y) := {} & \nabla I(\lfloor x \rfloor, \lfloor y \rfloor)\,(1 - \{x\})(1 - \{y\}) + \nabla I(\lfloor x+1 \rfloor, \lfloor y \rfloor)\,\{x\}(1 - \{y\}) + \\ & \nabla I(\lfloor x \rfloor, \lfloor y+1 \rfloor)\,(1 - \{x\})\{y\} + \nabla I(\lfloor x+1 \rfloor, \lfloor y+1 \rfloor)\{x\}\{y\}, \end{aligned} \tag{4.41}$$

where $\{x\} = x - \lfloor x \rfloor$ is known as fractional part of x.

The gradient of the image in a pixel can be approximated using different filters such as Prewitt and Sobel [61] (see Fig. 4.4). Since we evaluate the gradient in points that belong to the snake, it is convenient to extend the width of the filter to increase the region of attraction of gradient energy (see Fig. 4.5). Consequently, we use a generalization of the Prewitt filter of $2q + 1 \times 2q + 1$ to compute the gradient in those pixels with distance greater or equal to $q > 0$ (see Fig. 4.6) to the boundary of the image. For the rest of the pixels, we use Sobel filter to approximate the gradient. The constant value q depends on the image dimensions.

-1	-1	-1
0	0	0
1	1	1

$\frac{\partial I}{\partial x}$

-1	0	1
-1	0	1
-1	0	1

$\frac{\partial I}{\partial y}$

(a) Prewit: for $\frac{\partial I}{\partial x}$ (left) and $\frac{\partial I}{\partial y}$ (right)

-1	-2	-1
0	0	0
1	2	1

$\frac{\partial I}{\partial x}$

-1	0	1
-2	0	2
-1	0	1

$\frac{\partial I}{\partial y}$

(b) Sobel: for $\frac{\partial I}{\partial x}$ (left) and $\frac{\partial I}{\partial y}$ (right)

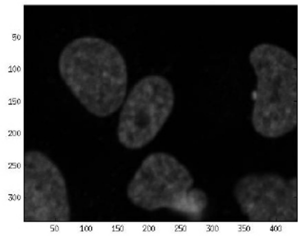
Figure 4.4: Some filters known to compute approximations of the gradient in a pixel.

The gradient of the image in each pixel is precomputed, so that the evaluations in (4.7) and its derivatives [37] use the stored values.

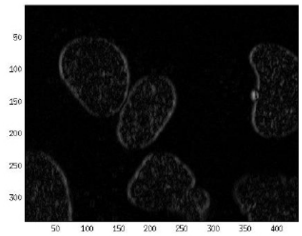
Region energy

The first step to compute the integrals (4.8) defining the region energy is to obtain a sequence of pixels that approximates the snake, that is represented by the polygon with vertices $\{\mathbf{r}(i/q) = (x_i^q, y_i^q), i = 0, \ldots, q\ M - 1\}$. The problem is reduced to the rasterization of each edge of that polygon. Rasterization algorithms provide the pixels that are intersected by a straight line (see Figure 4.7a). However, these are more pixels than the ones needed to describe the region Ω enclosed by a closed polygon. We instead select only one pixel per edge of the polygon for each horizontal line[7]. To obtain these pixels, called *boundary pixels*, we determine for the horizontal line j the pixels (j, l_j^i)

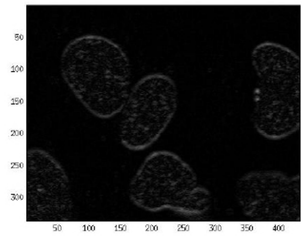
[7] We choose the horizontal direction without loss of generality, the same result is obtained if the vertical direction is chosen.



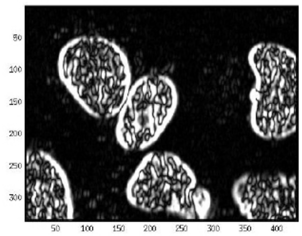
(a) Image I



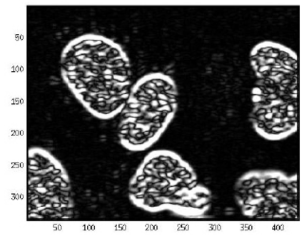
(b) Filter Sobel: $\frac{\partial I}{\partial x}$



(c) Filter Sobel: $\frac{\partial I}{\partial y}$



(d) Filter 7×7: $\frac{\partial I}{\partial x}$



(e) Filter 7×7: $\frac{\partial I}{\partial y}$

Figure 4.5: Comparison of different sizes of the proposed filter to compute the gradient of an image: Sobel (3×3) and the proposed for 7×7.

which are simultaneously on the line and on the edge $[\mathbf{r}(i/q), \mathbf{r}((i+1)/q)]$. If the result of the previous operation is more than one pixel, then we select the outer pixel with respect to the region enclosed by the subdivision curve (see Fig. 4.7b). We explain the reason why we proceed as follows.

We classify the edge $[\mathbf{r}(i/q), \mathbf{r}((i+1)/q)]$ as *downhill, horizontal*, or *uphill* if the sign of $x_i^q - x_{i+1}^q$ is negative, zero, or positive, respectively [8]. To compute approximately the integrals in (4.8) it is necessary to chose, for a given edge $[\mathbf{r}(i/q), \mathbf{r}((i+1)/q)]$, one pixel with coordinates (j, l_j^i) for each image row j, with $\min\{\lceil x_i^q \rceil, \lceil x_{i+1}^q \rceil\} \leq j \leq \max\{\lceil x_i^q \rceil, \lceil x_{i+1}^q \rceil\}$. The value of l_j^i depends on the previous edge classification as follows. Let $r_i(x)$ be the equation of the line passing through the pixels $(\lceil x_i^q \rceil, \lceil y_i^q \rceil)$ and $(\lceil x_{i+1}^q \rceil, \lceil y_{i+1}^q \rceil)$. Then

$$r_i(x) = \lceil y_i^q \rceil + \frac{\lceil y_{i+1}^q \rceil - \lceil y_i^q \rceil}{\lceil x_{i+1}^q \rceil - \lceil x_i^q \rceil}(x - \lceil x_i^q \rceil).$$

If $[\mathbf{r}(i/q), \mathbf{r}((i+1)/q)]$ is a downhill edge (see Figure 4.7b, left: edge $[\mathbf{r}(4), \mathbf{r}(5)]$),

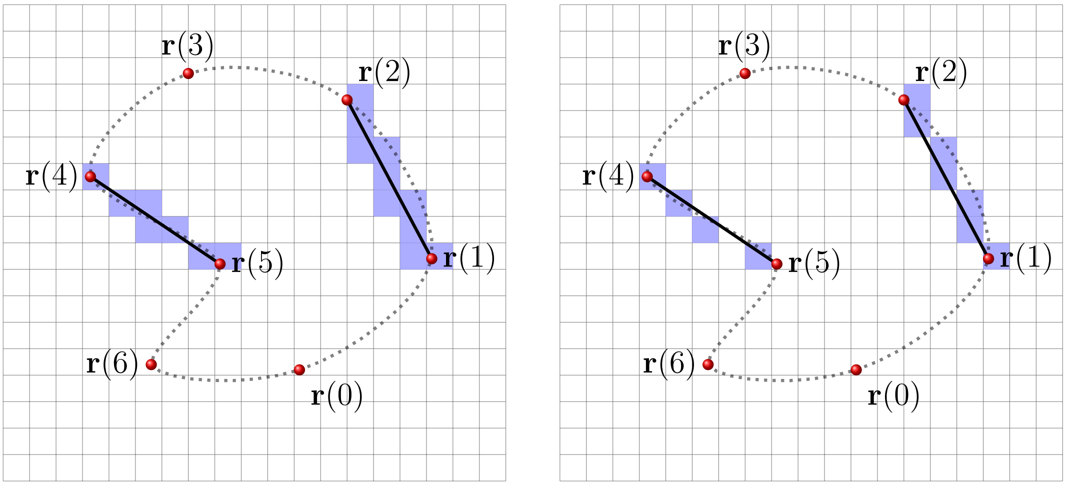
[8]Remember that we are using the system of coordinates defined by $(row, column)$.

$$
\begin{array}{c}
\overbrace{\hphantom{-1\;-1\;\cdots\;-1\;-1}}^{2q+1}\\
q\left\{\begin{array}{ccccc}
-1 & -1 & \cdots & -1 & -1\\
 & \vdots & \ddots & \vdots & \\
-1 & -1 & & -1 & -1
\end{array}\right.\\
\begin{array}{ccccc}
0 & 0 & \cdots & 0 & 0
\end{array}\\
q\left\{\begin{array}{ccccc}
1 & 1 & & 1 & 1\\
 & \vdots & \ddots & \vdots & \\
1 & 1 & \cdots & 1 & 1
\end{array}\right.
\end{array}
\qquad
2q+1\left\{\begin{array}{ccccccc}
\multicolumn{3}{c}{\overbrace{\hphantom{-1\;\cdots\;-1}}^{q}} & & \multicolumn{3}{c}{\overbrace{\hphantom{1\;\cdots\;1}}^{q}}\\
-1 & \cdots & -1 & 0 & 1 & \cdots & 1\\
-1 & \cdots & -1 & 0 & 1 & \cdots & 1\\
\vdots & \ddots & & \vdots & & \ddots & \vdots\\
-1 & \cdots & -1 & 0 & 1 & \cdots & 1\\
-1 & \cdots & -1 & 0 & 1 & \cdots & 1
\end{array}\right.
$$

Figure 4.6: Proposed filter to compute approximations of the gradient of an image: for $\frac{\partial I}{\partial x}$ (left) and $\frac{\partial I}{\partial y}$ (right).

then

$$l_j^i = \min\left\{\lceil r_i(j)\rceil, \lceil r_i(j+1)\rceil\right\}, \qquad j \in \left[\lceil x_i^q\rceil, \lceil x_{i+1}^q\rceil\right]. \tag{4.42}$$



(a) Rasterization of straight lines (b) Boundary pixels to describe the boundary

Figure 4.7: Pixel discretization of a straight line for a left edge and a right edge describing the boundary of a region.

If $[\mathbf{r}(i/q), \mathbf{r}((i+1)/q)]$ is a uphill edge (see Figure 4.7b, right: edge $[\mathbf{r}(1), \mathbf{r}(2)]$), then

$$l_j^i = \max\left\{\lceil r_i(j)\rceil, \lceil r_i(j+1)\rceil\right\}, \qquad j \in \left[\lceil x_{i+1}^q\rceil, \lceil x_i^q\rceil\right]. \tag{4.43}$$

Finally, if $[\mathbf{r}(i/q), \mathbf{r}((i+1)/q)]$ is a horizontal edge, then there is no need to define the value of l_j^i as $\operatorname{sign}(x_i^q - x_{i+1}^q) = 0$ in (4.13) and (4.14). In this case, the description of the boundary makes use of the neighbor edges.

To describe the boundary of the region enclosed by the subdivision curve, we store pairs of *boundary pixels* with respect to each horizontal. The amount of pairs on each horizontal line depends on the convexity of the curve (see Fig. 4.8). It should be noticed that may exist pixels that define left and right boundary at the same time[9].

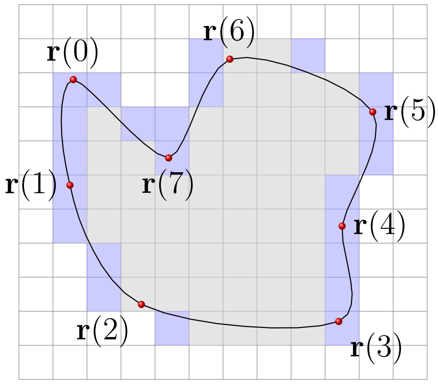


Figure 4.8: Description of the boundary of a region with pairs of *boundary pixels*.

Quantitative evaluation of results

When we work with synthetic images the ground-truth region, composed by pixels belonging to the object Γ, is known. In some real images of Berkeley database, the ground-truth is also given. In all these cases it is possible to validate quantitatively the quality of the results using the Jaccard distance J between Γ and the region Ω enclosed by the snake, given by

$$J = 1 - \frac{|\Omega \cap \Gamma|}{|\Omega \cup \Gamma|},$$

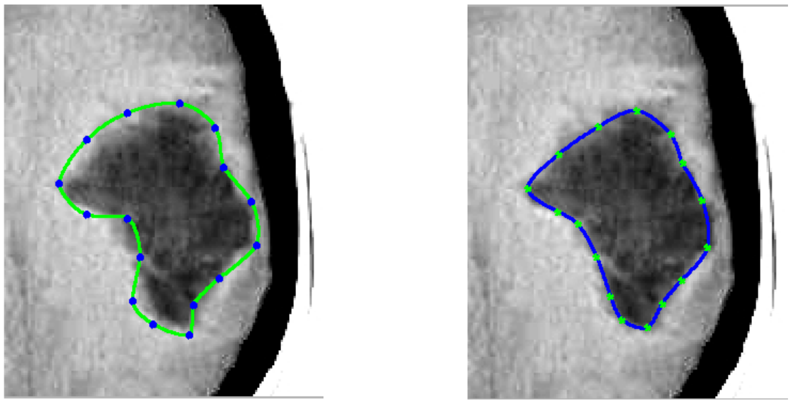
where $|G|$ denotes the area of region G. Observe that $0 \leq J \leq 1$ and a value of J close to 0 indicates a good segmentation. In [37] a table with few selected tests for the cubic B-spline subdivision curves and the 4-points subdivision scheme is shown.

4.2.5 Final remarks

The experiments in [37] using synthetic and real images (see Fig. 4.9) confirm that the proposed method is fast and robust. Our flexible computational framework facilitates

[9]For example, the pixel corresponding to $\mathbf{r}(7)$ in 4.8

the interaction with the snake by letting the user directly move the control points with the mouse and to control the weights associated to the combination of both energy functionals. Different subdivision schemes may be selected, and the evaluation of the subdivision curves and their derivatives can be computed as in the previous chapters.



(a) initial snake (b) optimal snake

Figure 4.9: Contour description of an intracerebral hemorrhage with a quintic B-spline after a swap of color to make the background brighter than the object.

The choice of the parameter q for the sampling at q-adic values (i.e., $c\left(\frac{1}{q}\mathbb{Z}\right)$) could be done in a non-uniform way for every parametric interval $[j, j+1]$. According to this idea, the sampling of the curve $c(t)$ becomes

$$\left\{c\left(j+\frac{s}{q_j}\right),\ s=0,\ldots,q_j-1,\ j=0,\ldots,M-1\right\}. \tag{4.44}$$

In this setting, each integral can be split into M separated definite integrals for the quadrature of every energy. The advantage of this approach is to have a similar density of samples along the arc-length for each parametric interval, as the Euclidean distance between consecutive samples $c(\frac{s}{q})$ and $c(\frac{s+1}{q})$ is not uniform in every instance. The Fig. 1.2 illustrates this fact, where some segments of the curve have more points clustered than others. In the free-form curve design, this feature is controlled by using arc-length based subdivision rules, like chordal and centripetal parametrization in [44]. However, that choice leads to non-linear and non-uniform subdivision schemes, which are more difficult for analyzing theoretical convergence.

The main contribution of our approach is a new region energy designed to maximize the contrast between the average intensity of the image within the snake and over the complement of the snake in a bounding box. This energy is simpler and computationally cheaper than other similar energies proposed in the literature [12],[113],[28]. In our region energy the bounding box containing the object to be segmented does not change during the optimization process. Moreover, the average intensity inside and outside the snake needs neither to be estimated a priori nor to be included among the optimization parameters. Finally, in comparison with other methods, we can compute a better and more robust approximation of the region energy using a method for obtaining a pixel-level discretization of the snake. Our method produces good approximations of the region energy for images of either low or high resolution.

The proposed method can be extended in several directions. In particular, the new region energy may be generalized from one channel to three channels and different approaches of multiresolution optimization may be applied.

Chapter 5

Conclusions

Scalar linear subdivision schemes were the first ones treated in the literature. With the evolution of the field, more complex schemes such as non-stationary schemes and Hermite schemes along with their properties have been studied. The interpolation problem is one of the topics that have been studied and improved on over time.

Interpolatory scalar schemes and interpolatory Hermite schemes provide the interpolation directly. On the other hand, approximating schemes require a preprocessing for computing a suitable set of control points so that the subdivision curve interpolates the data. This problem has been addressed partially for some primal schemes, such as cubic B-splines curves and surfaces in the literature. However, we did not find a general study regarding linear approximating schemes, while trying to use them for some real-world applications, such as the ones analyzed in the last chapter.

Approximating subdivision schemes provides higher continuity degree than interpolatory schemes for the same support of the basic function. Hermite subdivision schemes demand more complex and subtle tools for the study of convergence and some of their properties. Meanwhile, approximating schemes are still used for curves and surface generation in several applications. In addition, Hermite schemes generate a spline curve that can present lower continuity than scalar schemes at the data points.

Hence, in this thesis, we covered some gaps concerning the use of scalar linear subdivision schemes for solving the Hermite interpolation with point interpolation as a particular case. Using our strategies, we showed an application to free-form curve design, exact offset computation and image segmentation by using snake curves.

The use of Chebyshev series to analyze the spectrum of the defined interpolation operator allowed us to distinguish the primal and dual cases, and to propose a proper solution for each one. Even when similar strategies can be found in the literature for the primal case, we have not found in the literature such an approach for the dual case.

By dealing with a direct solution to the interpolation problem, we avoided some problems encountered by other iterative solutions in the literature related to the spectral radius of the iteration operator, which loses convergence even when the problem is well defined.

The presence of singular interpolation operators was tackled by three approaches. The first one is the possibility of using regularization methods. In particular, an ω-circulants perturbation has been proposed, even if more work is needed in order to solve the problem satisfactorily [36].

The second was a suitable perturbation with a filter for the pseudo-inverse operator by using certain fairness energy functionals. Those fairness functionals are usually non linear and in some cases even non convex. To deal with that situation without presenting expensive computational cost, but also without losing quality, selected discrete approximations are analyzed. In that way the chosen interpolating curves are solutions of a quadratic optimization problem which translates into a linear system of equations. The Fourier factorization of the matrices used simplified the solution of those equations. Moreover, the dimensions of such problems are not enough to consider them as large problems.

The last one was a shifting of parameters for the interpolation. As a result, we were able to add another degree of freedom for a family of curve interpolating points (see Fig. 3.4), by means of interpolating at shifted parameters from the uniform parameterization of the subdivision curve. The solutions obtained in this way can be appealing for the free-form design of curves.

Although this thesis has been written with a formulation based on binary schemes, the model and results can be extended straightforward to higher arities. The use of a greater dilation factor for the basic function definition scales the involved matrices, but their structure remains unchanged. Therefore, the techniques used regarding circulant and Toeplitz matrices are the same.

Additionally, it is possible to apply the same procedures for non stationary schemes once the sampling of the limit basic function in (2.53) is computed.

Many open problems remain, for instance the extension to mesh interpolation. At least, the interpolation of quad-meshes vertices with tensor-product bivariate linear subdivision schemes should be straightforward as many similar proposal in the literature. Some of the techniques used in this thesis are general cases of specific solutions to that problem as shown in [62, 106].

Bibliography

[1] G. Albrecht. Invariante Gütekriterien im Kurvendesign - Einige neuere Entwicklungen. In H. Hagen et al., editor, *Effiziente Methoden der geometrischen Modellierung und der wissenschaftlichen Visualisierung / Dagstuhl-Seminar 1997*, pages 134–148. Springer, 1999.

[2] B. Appleton and H. Talbot. Globally optimal geodesic active contours. *Journal of Mathematical Imaging and Vision*, 23(1):67–86, 2005.

[3] A. Badoual, D. Schmitter, V. Uhlmann, and M. Unser. Multiresolution subdivision snakes. *IEEE Transactions on Image Processing*, 26(3):1188–1201, 2017.

[4] C. Beccari, Giulio Casciola, and Lucia Romani. A non-stationary uniform tension controlled interpolating 4-point scheme reproducing conics. *Computer Aided Geometric Design*, pages 1–9, 2007.

[5] D. Bini. Parallel solutions of certain Toeplitz linear systems. *SIAM Journal on Computing*, 13(2):268–276, 1984.

[6] J. P. Boyd. Computing Zeros on a Real Interval through Chebyshev Expansion and Polynomial Rootfinding. *SIAM Journal on Numerical Analysis*, 40(5):1666–1682, 2003.

[7] P. Brigger, J. Hoeg, and M. Unser. B-spline snakes: A flexible tool for parametric contour detection. *IEEE Transaction on Image Processing*, 9:1484–1496, 2000.

[8] A. P. Britto and G. Ravindran. Review of deformable curves - a retro analysis. *Information Technology Journal*, 6(1):26–36, 2007.

[9] G. Casciola and L. Romani. A piecewise rational quintic Hermite interpolant for use in CAGD. *Mathematical Methods for Curves and Surfaces, Nashboro Press, Brentwood, TN*, pages 39–49, 2005.

[10] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[11] G. M. Chaikin. An algorithm for high-speed curve generation. *Computer Graphics and Image Processing*, 3(4):346–349, 1974.

[12] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10:266–276, 2001.

[13] C. Chesnaud, P. Refregier, and V. Boulet. Statistical region snake-based segmentation adapted to different physical noise models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:1145–1157, 1999.

[14] C. Chui and J. de Villiers. *Wavelet Subdivision Methods: Gems for rendering curves and surfaces.* CRC Press, 2010.

[15] R. E. Cline, R. J. Plemmons, and G. Worm. Generalized inverses of certain Toeplitz matrices. *Linear Algebra and its Applications*, 8(1):25–33, 1974.

[16] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53:211–218, 1991.

[17] C. Conti, M. Cotronei, and T. Sauer. Convergence of level-dependent Hermite subdivision schemes. *Applied Numerical Mathematics*, 116(C):119–128, June 2017.

[18] C. Conti and N. Dyn. Non-stationary subdivision schemes: State of the art and perspectives. In Gregory E. Fasshauer, Marian Neamtu, and Larry L. Schumaker, editors, *Approximation Theory XVI*, pages 39–71, Cham, 2021. Springer International Publishing.

[19] C. Conti, N. Dyn, C. Manni, and M.-L. Mazure. Convergence of univariate non-stationary subdivision schemes via asymptotic similarity. *Computer Aided Geometric Design*, 37:1 – 8, 2015.

[20] C. Conti, L. Gemignani, and L. Romani. From symmetric subdivision masks of Hurwitz type to interpolatory subdivision masks. *Linear Algebra and its Applications*, 431(10):1971–1987, 2009.

[21] C. Conti, L. Gemignani, and L. Romani. From approximating to interpolatory non-stationary subdivision schemes with the same generation properties. *Advances in Computational Mathematics*, pages 217–241, 2011.

[22] C. Conti and L. Romani. Algebraic conditions on non-stationary subdivision symbols for exponential polynomial reproduction. *Journal of Computational and Applied Mathematics*, pages 543–556, 2011.

[23] C. Conti and L. Romani. Dual univariate m-ary subdivision schemes of de Rham-type. *Journal of Mathematical Analysis and Applications*, 407:443–456, 2013.

[24] J. A. Cottrell and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley Publishing, 1st edition, 2009.

[25] P. Davis. *Circulant Matrices*. J. Wiley and Sons, New York, 1979.

[26] C. De Boor. *A practical guide to splines; rev. ed.* Applied mathematical sciences. Springer, Berlin, 2001.

[27] G. de Rham. Un peu de mathématique à propos d'une courbe plane. *Elemente der Mathematik*, 2:73–76, 1947.

[28] R. Delgado-Gonzalo, P. Thevenaz, C. Seelamantula, and M. Unser. Snakes with an ellipse-reproducing property. *IEEE Transactions on Image Processing*, 21:1258–1271, 2012.

[29] R. Delgado-Gonzalo, P. Thévenaz, and M. Unser. Exponential splines and minimal-support bases for curve representation. *Computer Aided Geometric Design*, 29:109–128, 2012.

[30] R. Delgado-Gonzalo, V. Uhlmann, D. Schmitter, and M. Unser. Snakes on a Plane: A perfect snap for bioimage analysis. *IEEE Signal Processing Magazine*, 32(1):41–48, 2015.

[31] R. Delgado-Gonzalo and M. Unser. Spline-based framework for interactive segmentation in biomedical imaging. *IRBM*, 34(3):235–243, 2013.

[32] C. Deng, H. Xu, W. Ma, and Y. Li. Repeated local operations and associated interpolation properties of dual 2n-point subdivision schemes. *Journal of Computational and Applied Mathematics*, 349:344–353, 2019.

[33] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constructive Approximation*, 5:49–68, 1989.

[34] R. Díaz-Fuentes. Perturbación de los esquemas de Dubuc-Deslauriers para cualquier aridad. Master's thesis, Universidad de La Habana, 2015.

[35] R. Díaz-Fuentes. Subdiv_HerInterpol. `https://github.com/RafaelDieF/Subdiv_HerInterpol`, 2021.

[36] R. Díaz-Fuentes, M. Mazza, and S. Serra-Capizzano. A ω-circulant regularization for linear systems arising in interpolation with subdivision schemes. Technical Report 2021-002, Department of Information Technology, Uppsala University, March 2021.

[37] R. Díaz-Fuentes, J. Pino-Torres, V. Hernández-Mederos, and J. Estrada-Sarlabous. Stationary subdivision snakes for contour detection. *Revista Investigación Operacional*, 42(2):238–266, 2021.

[38] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[39] S. Dubuc. Scalar and Hermite subdivision schemes. *Applied and Computational Harmonic Analysis*, 21:376–394, 2006.

[40] S. Dubuc and J.L. Merrien. Convergent vector and Hermite subdivision schemes. *Constructive Approximation*, 23:1–22, 2005.

[41] S. Dubuc and J.L. Merrien. Hermite subdivision schemes and Taylor polynomials. *Constructive Approximation*, 29:219–245, 2009.

[42] N. Dyn. Linear and nonlinear subdivision schemes in Geometric Modeling, 2008.

[43] N. Dyn, M. S. Floater, and K. Hormann. A C^2 four-point subdivision scheme with fourth order accuracy and its extensions. In *Mathematical methods for curves and surfaces: TROMSØ 2004, Modern methods in Mathematics*, pages 145–156. Nashboro Press, 2005.

[44] N. Dyn, M. S. Floater, and K. Hormann. Four-point curve subdivision based on iterated chordal and centripetal parameterizations. *Computer Aided Geometric Design*, 26(3):279 – 286, 2009.

[45] N. Dyn, J. A. Gregory, and D. Levin. Analysis of uniform binary subdivision schemes for curve design. *Constructive Approximation*, 7(1):127–147, 1991.

[46] N. Dyn and D. Levin. Analysis of asymptotically equivalent binary subdivision schemes. *Journal of Mathematical Analysis and Applications*, 193(2):594–621, 1995.

[47] N. Dyn and D. Levin. Analysis of asymptotically equivalent binary subdivision schemes. *Journal of Mathematical Analysis and Applications*, 193(2):594–621, 1995.

[48] N. Dyn and D. Levin. Analysis of Hermite-interpolatory subdivision schemes. In S. Dubuc and G. Deslauriers, editors, *Spline Functions and the Theory of Wavelets*, volume 18, pages 105–113. CRM, 1999.

[49] N. Dyn and D. Levin. Subdivision schemes in Geometric Modelling. *Acta Numerica*, 11:73–144, 2002.

[50] N. Dyn, D. Levin, and J. A. Gregory. A 4-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design*, 4:257–268, 1987.

[51] G. Elber, I-K. Lee, and M-S. Kim. Comparing offset curve approximation methods. *IEEE Computer Graphics and Applications*, 17:62–71, 1998.

[52] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Springer, 1996.

[53] G. Farin. Chapter 1 - A history of curves and surfaces in CAGD. In G. Farin, J. Hoschek, and M.-S. Kim, editors, *Handbook of Computer Aided Geometric Design*, pages 1 – 21. North-Holland, Amsterdam, 2002.

[54] G. Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufman Publishers Inc., 5 edition, 2002.

[55] G. Farin. Geometric Hermite interpolation with circular precision. *Computer Aided Design*, 40(4):476–479, 2008.

[56] G. Farin and N. Sapidis. Curvature and the fairness of curves and surfaces. *IEEE Computer Graphics and Applications*, 9(2):52–57, March 1989.

[57] R. T. Farouki. *Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable*. Springer, 2008.

[58] M. A. Figueiredo, J. M. Leitão, and A. K. Jain. Unsupervised contour representation and estimation using B-splines and a minimum description length criterion. *IEEE Transactions on Image Processing*, 9(6):1075–1087, June 2000.

[59] C. Garoni and S. Serra-Capizzano. *Generalized Locally Toeplitz Sequences: Theory and Applications*, volume I. Springer, 2017.

[60] C. Garoni, S. Serra-Capizzano, and D. Sesana. Spectral analysis and spectral symbol of d-variate $\mathbb{Q}_p$ Lagrangian FEM stiffness matrices. *SIAM Journal on Matrix Analysis and Applications*, 36(3):1100–1128, 2015.

[61] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing using MATLAB*. Gatesmark, 2nd edition, 2009.

[62] M. A. Halstead, M. Kass, and T. DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. In *SIGGRAPH'93*, pages 35–44, 1993.

[63] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Society for Industrial and Applied Mathematics, 1999.

[64] M. F. Hassan. *Multiresolution in Geometric Modelling: Subdivision Mark Points and Ternary Subdivision*. PhD thesis, University of Cambridge, Computer Laboratory, April 2005.

[65] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *SIGGRAPH 94*, pages 295–302, 1994.

[66] J. Hug, C. Brechbühler, and G. Szèkely. Tamed snake: A particle system for robust semi-automatic segmentation. In *MICCAI '99 Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 106–115. Springer-Verlag London, 1999.

[67] D. Lasser J. Hoschek. *Grundlagen der geometrischen Datenverarbeitung*. Vieweg+Teubner Verlag, 2 edition, 1992.

[68] M. Jacob, T. Blu, and M. Unser. Unifying approach and interface for spline-based snakes. In Milan Sonka and Kenneth M. Hanson, editors, *Medical Imaging 2001: Image Processing*, volume 4322, pages 340 – 347. International Society for Optics and Photonics, SPIE, 2001.

[69] M. Jacob, T. Blu, and M. Unser. Efficient energies and algorithms for parametric snakes. *IEEE Transactions on Image Processing*, 13:1231–1244, 2004.

[70] G. Jaklič and E. Žagar. Curvature variation minimizing cubic Hermite interpolants. *pplied Mathematics and Computation*, 218:3918–3924, 2011.

[71] R. Q. Jia. Interpolatory subdivision schemes induced by box splines. *Applied and Computational Harmonic Analysis*, 8(3):286–292, 2000.

[72] I. Juhász and Á. Róth. Adjusting the energies of curves defined by control points. *Computer-Aided Design*, 107:77–88, 2019.

[73] B. Jüttler and U. Schwanecke. Analysis and design of Hermite subdivision schemes. *The Visual Computer*, 18:326–342, 2002.

[74] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–332, January 1988.

[75] I. Kra and S. Simanca. On circulant matrices. *Notices of the American Mathematical Society*, 59(3):368–377, 2012.

[76] A. Kunoth. On the fast evaluation of integrals of refinable functions. In *An International Conference on Curves and Surfaces on Wavelets, Images, and Surface Fitting*, page 327–334, USA, 1994.

[77] J. Lane and R. Riesenfeld. A theoretical development for the computer generation and display of piecewise polynomial surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:25–46, 1980.

[78] G. Li and W. Ma. A method for constructing interpolatory subdivision schemes and blending subdivisions. In *Computer Graphics Forum*, volume 26, pages 185–201. Wiley Online Library, 2007.

[79] J. Li. A class of quintic Hermite interpolation curve and the free parameters selection. *Journal of Advanced Mechanical Design, Systems and Manufacturing*, 13(1), 2019.

[80] L. Lu, C. Jiang, and Q. Hu. Planar cubic G^1 and quintic G^2 Hermite interpolations via curvature variation minimization. *Computers & Graphics*, 70:92–98, 2018.

[81] M. Lutti. Offset di curve di suddivisione piane. Master's thesis, University of Milano-Bicocca, 2017.

[82] T. Maekawa. An overview of offset curves and surfaces. *Computer-Aided Design*, 31:165–73, 1999.

[83] T. McInerney. Sketchsnakes: Sketch-line initialized snakes for efficient interactive medical image segmentation. *Computerized medical imaging and graphics*, 32(5):331–352, 2008.

[84] J. L. Merrien. A family of Hermite interpolants by bisection algorithms. *Numerical Algorithms*, 2:187–200, 1992.

[85] J. L. Merrien. Dyadic Hermite interpolation on a triangulation. *Numerical Algorithms*, 7:391–410, 1994.

[86] J. L. Merrien and T. Sauer. From Hermite to stationary subdivision schemes in one and several variables. *Advances in Computational Mathematics*, 36:547–579, 2012.

[87] C. Moosmüller and N. Dyn. Increasing the smoothness of vector and Hermite subdivision schemes. *IMA Journal of Numerical Analysis*, 39(2):579–606, 2019.

[88] E. Ngondiep, S. Serra-Capizzano, and D. Sesana. Spectral features and asymptotic properties for g-circulants and g-Toeplitz sequences. *SIAM Journal on Matrix Analysis and Applications*, 31:1663–1687, January 2010.

[89] S. Okaniwa, A. Nasri, H. Lin, A. Abbas, Y. Kineri, and T. Maekawa. Uniform B-spline curve interpolation with prescribed tangent and curvature vectors. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1474–1487, September 2012.

[90] B. J. Olson, S. W. Shaw, C. Shi, C. Pierre, and R. G. Parker. Circulant matrices and their application to vibration analysis. *Applied Mechanics Reviews*, 66(4), 06 2014. 040803.

[91] B. Pham. Offset approximation of uniform B-splines. *Computer-Aided Design*, 20(8):471–474, 1988.

[92] B. Pham. Offset curves and surfaces: A brief survey. *Computer-AidedDesign*, 24:223–9, 1992.

[93] G. Plonka. An efficient algorithm for periodic Hermite spline interpolation with shifted nodes. *Numerical Algorithms*, 5:51–62, 1993.

[94] G. Plonka and M. Tasche. Efficient algorithms for periodic Hermite spline interpolation. *Mathematics of Computation*, 58:693–703, 1992.

[95] D. Ranirina and J. de Villiers. On Hermite vector splines and multi-wavelets. *Journal of Computational and Applied Mathematics*, 349:366–378, 2019.

[96] L. Romani. Interpolating m-refinable functions with compact support: The second generation class. *Applied Mathematics and Computation*, 361:735–746, 2019.

[97] L. Romani, V. Hernández-Mederos, and J. Estrada-Sarlabous. Exact evaluation of a class of nonstationary approximating subdivision algorithms and related applications. *IMA Journal of Numerical Analysis*, 36(1):380–399, 2015.

[98] L. Romani and A. Viscardi. Dual univariate interpolatory subdivision of every arity: Algebraic characterization and construction. *Journal of Mathematical Analysis and Applications*, 484(1):123713, 2020.

[99] L. Romani and A. Viscardi. On the refinement matrix mask of interpolating Hermite splines. *Applied Mathematics Letters*, 109:106524, 2020.

[100] J. Rossignac. Education-driven research in CAD. *Computer Aided Design*, 36(14):1461–1469, 2004.

[101] J. Rossignac and S. Schaefer. J-splines. *Computer Aided Design*, 40(10-11):1024–1032, 2008.

[102] M. Sabin. Recent progress in subdivision: a survey. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, pages 203–230, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[103] M. Sabin. *Analysis and Design of Univariate Subdivision Schemes*. Springer, 2010.

[104] P. Sablonnière. Bernstein-type bases and corner cutting algorithms for C^1 Merrien's curves. *Advances in Computational Mathematics*, 20:229–2436, 2004.

[105] T. Sakkalis, R. T. Farouki, and L. Vaserstein. Non-existence of rational arc length parameterizations for curves in r^n. *Journal of Computational and Applied Mathematics*, 228(1):494–497, June 2009.

[106] S. Schaefer and J. Warren. Exact evaluation of limits and tangents for non-polynomial subdivision schemes. *Computer Aided Geometric Design*, 25(8):607 – 620, 2008. Computer Graphics and Applications.

[107] S. Serra-Capizzano. A Korovkin-type theory for finite Toeplitz operators via matrix algebras. *Numerische Mathematik*, 82:117–142, March 1999.

[108] Z. Shi and F. Li. Active subdivision snake scheme. *Applied Mechanics and Materials*, 44-47:3917–3921, 2011.

[109] Z. Šír, R. Feichtinger, and B. Jüttler. Approximating curves and their offsetsusing biarcs and Pythagorean hodograph quintics. *Computer-Aided Design*, 38:608–18, 2006.

[110] L. H. Staib and J. S. Duncan. Boundary fitting with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:1061–1075, November 1992.

[111] E. J. Stollnitz, A. D. DeRose, and D. H. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann, 1996.

[112] J. Tan, X. Zhuang, and L. Zhang. A new four-point shape-preserving C^3 subdivision scheme. *Computer Aided Geometric Design*, 31(1):57 – 62, 2014.

[113] P. Thevenaz, R. Delgado-Gonzalo, and M. Unser. The ovuscule. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:382–393, 2011.

[114] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, extended edition, 2019.

[115] M. D. van der Walt. Ternary interpolatory subdivision. Master's thesis, Stellenbosch University, 2012.

[116] R. C. Veltkamp and W. Wesselink. Modeling 3*D* curves of minimal energy. *Computer Graphics Forum*, 14:97–110, 1995.

[117] C. Xu and J. L. Prince. Snakes, Shapes, and Gradient vector Flow. *IEEE Transactions on Image Processing*, 7:359–369, 1998.

[118] F. Yamaguchi. *Curves and Surfaces in Computer Aided Geometric Design*. Springer-Verlag, 1988.

[119] H. Zhang, Z. Bian, Y. Guo, B. Fei, and M. Ye. An efficient multiscale approach to level set evolution. In *25th Annual International Conference of the IEEE Engineering in Medicine and Biological Society*, pages 694–697, September 2003.

[120] H-Y. Zhao and G-J. Wang. Error analysis of reparametrization based approaches for curve offsetting. *Computer-Aided Design*, 39:142–8, 2007.