Geometric Modeling

Vorlesung, zuerst gehalten im Sommersemester 2015

Tomas Sauer

Version 0.0
Letzte Änderung: 12.7.2015

0 Statt einer Leerseite ...

Denn die Doktrin der geschlechtergerechten Sprache macht das Lesen solchermassen "'gerechter"' Texte nicht nur fast unerträglich. Sie basiert auch auf einem linguistischen Grundirrtum, weil es das biologische Geschlecht mit dem grammatischen Genus gleichsetzt.

C. Wirz, "'Neusprech für Fortgeschrittene"', *NZZ Online, 12.7.2013*

Die wahren Analphabeten sind schließlich diejenigen, die zwar lesen können, es aber nicht tun. Weil sie gerade fernsehen.

L. Volkert, *SZ–Online, 11.7.2009*

$$\frac{(a+b)!}{a!b!} \ge \sqrt{\frac{(a+b)^{a+b}}{a^a b^b}}.$$

And it didn't stop being magic just because you found out how it was done.

T. Pratchett, *Wee Free Men*

Oh mein Gott! Ein englischsprachiges Skript, nicht mehr die guten alten Splines oder Geometrische Modellierung, nein, Geometric Modeling muss es sein. Aber im Zeitalter der Internationalisierung von Studiengängen bleibt einem nichts anderes übrig und letztendlich gilt halt doch: *The official language of science is broken English*. In diesem Sinne: Viel Spass damit.

Tomas Sauer
Lehrstuhl für Mathematik mit Schwerpunkt Digitale Bildverabeitung
Universität Passau
Innstr. 43
94032 Passau

Contents | 0

What are the digits that encode beauty, the number-fingers that enclose, transform, transmit, decode, and somehow, in the process, fail to trap or choke the soul of it? Not because of the technology but in spite of it, beauty, that ghost, that treasure, passes undiminished through the new machines.

S. Rushdie, *Fury*

1 Introduction

Geometric Modeling is a task that gains more and more importance as the world, especially the industrial world, is getting more an more digital. At times where 3D printers are available to practically everyone, a digital model of the objects one wants to deal with is unavoidable. This becomes more and more important

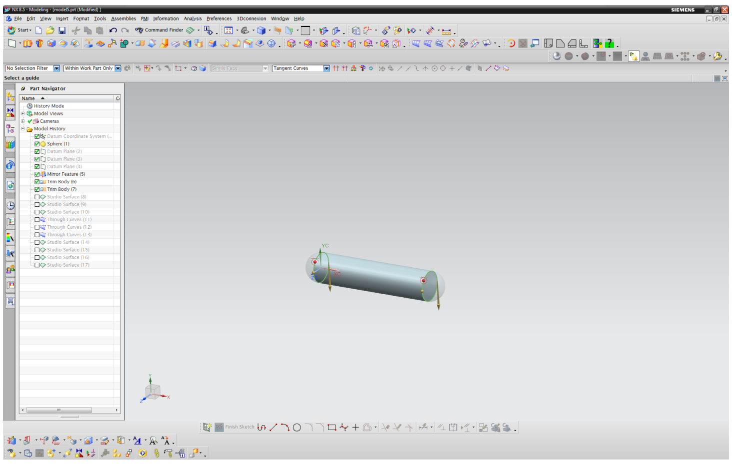

Figure 1.1: Screenshot of the Siemens PLM program **NX**. This is the first step, modeling the geometry. Needless to say that the system also supports significantly more complex parts. (F. Lorenz, Siemens AG)

if not only functional issues, like in in the old days of CAD[1], but also aesthetic

[1]Pieces of simple geometry that had to fit somewhere and work as requested. Bolts and nuts …

desires have to be satisfied, for example in architectural geometry or 3D animated movies or special effects. All these are questions of mathematical nature and require a richer and richer toolbox of curves and surfaces as well as methods to adapt these objects to the needs of an application.

This is the world of *Geometric Modeling*. How can we work with **shapes**: curves and surfaces that describe objects, how can we manipulate them, how can we measure their quality[2] and how can we force them to what we want.

Figure 1.2: A CNC test workpiece for testing how well the geometric model is really generated physically.

Geometric modeling is about creating objects. But that is only the beginning. Noone buys expensive computers and even more expensive programs just to get nice pictures. In the end, something has to be created: a movie, a real world object produced by a CNC machine or a 3D printer, for example or even a building. Modeling is quite pointless without **manufacturing**. Even if we will only learn about the modeling issue in this lecture, i.e., the CAD[3] in the CAD/CAM system, it surely influences the CAM[4] part of the system and the design process must be aware of the type of "manufacturing" that is done afterwards.

Of course, we mainly do mathematics here and fairly much ignore the technical aspects of the manufacturing process, so we end up in CAGD: **C**omputer **A**ided **G**eometric **D**esign which is the mathematical area that is interested in the math behind CAD. Let's have fun.

[2]And, even trickier: What **is** quality? How can distinguish nice objects from not so nice ones, how can we measure beauty in numbers?

[3]**C**omputer **A**ided **D**esign

[4]**C**omputer **A**ided **M**anufacturing

Division and multiplication were discovered. Algebra was invented and provided in interesting diversion for a minute or two. And then he felt the fog of numbers drift away, and looked up and saw the sparkling, distant mountains of calculus.

T. Pratchett, *Men at arms*

2 Coordinates

To model the geometric objects by numbers in a reasonable way, the first step is to fix a proper reference system.

2.1 Cartesian coordinates and vector spaces

The classical and most common way to attach coordinates to our environment is by embedding it into $\mathbb{R}^d$, where d stands for "'**dimension**"'. Realistic cases are $d = 2$ if we draw on a piece of paper or on a computer screen or $d = 3$ if we consider the 3D space around us. Nevertheless it can be a good idea[5] to keep d variable and even consider higher dimensional spaces.

Definition 2.1 (Cartesian space) *The **Cartesian space** $\mathbb{R}^d$ is the usual vector space of all points $\mathbf{x} = (x_1, \dots, x_d)$ with componentwise addition and multiplication:*

$$\mathbf{x} + \mathbf{x}' = \begin{bmatrix} x_1 + x_1' \\ \vdots \\ x_d + x_d' \end{bmatrix}, \qquad \lambda\mathbf{x} = \begin{bmatrix} \lambda x_1 \\ \vdots \\ \lambda x_d \end{bmatrix}, \quad \lambda \in \mathbb{R}. \tag{2.1}$$

A word on notation and terminology: Formally we would have to distinguish between the **tuple** x and the **vector** x. A tuple is just a finite sequence of values while a vector is a member of a **vector space** and therefore certain *operations*, addition and multiplication by scalars, have to be defined for x. Since we will also deal with matrices, we always write a vector as a **column vector** like in (2.1), but will omit transposition in a "'tuple notation"' like $\mathbf{x} = (x_1, \dots, x_d)$ where row or column is simply irrelevant.

If $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ are vectors, we can align them as column vectors into a **matrix**

$$\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_n] = \begin{bmatrix} x_{11} & \dots & x_{n1} \\ \vdots & \ddots & \vdots \\ x_{1d} & \dots & x_{nd} \end{bmatrix} \in \mathbb{R}^{d \times n}. \tag{2.2}$$

[5] At least mathematically, but it is mathematics we are here for.

We can even see $\mathbf{X}$ as the **ordered set** of $x_1, \ldots, x_n$ or, more precisely, as the ordered **multiset** as there can be repetitions among the $\mathbf{x}_j$.

Example 2.2 *With* $d = 1$ *and* $\mathbf{x}_1 = \mathbf{x}_2 = 1$, $\mathbf{x}_3 = 0$, *the multiset consists of*

$$\mathbf{X} = [110],$$

i.e., a "double" 1.

In Cartesian space, a point $\mathbf{x} \in \mathbb{R}^d$ corresponds to the vector that connects the origin $0 = (0, \ldots, 0)$ to this point in space. Addition of two points is defined as addition of these two vectors. This means that there always must be a particular point, namely the **origin**, in the coordinate system.

Definition 2.3 (Unit vectors) *The* j*–th* ***unit vector*** $\mathbf{e}_j$ *in* $\mathbb{R}^d$ *is defined by*

$$e_{jk} = (\mathbf{e}_j)_k = \delta_{jk} = \begin{cases} 1, & j = k, \\ 0, & j \neq k \end{cases}$$

Definition 2.4 (Products) *The* ***inner product*** *or* ***scalar product*** *of* $\mathbf{x}, \mathbf{x}'$ *is the number*

$$\mathbf{x} \cdot \mathbf{x}' = \mathbf{x}^T \mathbf{x} = \sum_{j=1}^{d} x_j x_j', \tag{2.3}$$

the ***outer product*** *or* ***tensor product*** *is the matrix*

$$\mathbf{x} \otimes \mathbf{x}' = \mathbf{x}\mathbf{x}'^T := \left[x_j x_k' : \begin{array}{c} j = 1, \ldots, d \\ k = 1, \ldots, d \end{array} \right] = \begin{bmatrix} x_1 x_1' & \ldots & x_1 x_d' \\ \vdots & \ddots & \vdots \\ x_d x_1' & \ldots & x_d x_d' \end{bmatrix}. \tag{2.4}$$

The ***euclidean length*** $\|\mathbf{x}\|$ *of* $\mathbf{x}$ *is defined as*

$$\|\mathbf{x}\| := \sqrt{\mathbf{x}^T \mathbf{x}} = \left(\sum_{j=1}^{d} x_j \right)^{\frac{1}{2}}. \tag{2.5}$$

The ***angle*** *between two vectors* $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ *is defined as*[6]

$$\angle(\mathbf{x}, \mathbf{x}') := \cos^{-1} \frac{\mathbf{x}^T \mathbf{x}'}{\|\mathbf{x}\| \, \|\mathbf{x}'\|} \in \mathbb{T} := \mathbb{R}/2\pi\mathbb{Z} \simeq [-\pi, \pi). \tag{2.6}$$

Exercise 2.1 Show that $\mathbf{x} \otimes \mathbf{x}'$ has rank 1 and prove the identity

$$\mathbf{x}' \otimes \mathbf{x} = (\mathbf{x} \otimes \mathbf{x}')^T.$$

◊

Since $\mathbf{e}_j \otimes \mathbf{e}_k$ is the matrix with 1 at position j, k and zero everywhere else we get the[7] formula

$$\mathbf{A} = \left[a_{jk} : \begin{array}{c} j = 1, \ldots, d \\ k = 1, \ldots, d \end{array} \right] = \sum_{j,k=1}^{d} a_{jk} \mathbf{e}_j \otimes \mathbf{e}_k. \tag{2.7}$$

[6]This is a consequence of the inner product formula $\mathbf{x}^T \mathbf{x}' = \|\mathbf{x}\| \, \|\mathbf{x}'\| \cos\theta$ where then θ is defined to be the angle between the two vectors.

[7]Impressive but quite trivial

Definition 2.5 (Affine transformation) *Given* $\mathbf{y} \in \mathbb{R}^d$ *and a matrix* $\mathbf{A} \in \mathbb{R}^{d\times d}$, *the mapping*

$$\mathbf{x} \mapsto \mathbf{A}\mathbf{x} + \mathbf{y} \tag{2.8}$$

is called an ***affine transformation****. Special cases are*

1. $\mathbf{A} = \mathbf{I}$: ***translation*** $T_{\mathbf{y}} : \mathbf{x} \mapsto \mathbf{x} + \mathbf{y}$.

2. $\mathbf{y} = 0$, $\mathbf{A}$ *diagonal:* ***scaling****, where different coordinates can be scaled differently.*

3. $\mathbf{y} = 0$, $\mathbf{A} = \mathbf{I} + (\cos\alpha - 1)\,(\mathbf{e}_j \otimes \mathbf{e}_j + \mathbf{e}_k \otimes \mathbf{e}_k) + \sin\alpha(\mathbf{e}_k\mathbf{e}_j^T - \mathbf{e}_j\mathbf{e}_k^T)$: ***rotation****. A rotation has the form*

$$\mathbf{A} = \begin{bmatrix} 1 &&&&&&&&&& \\ & \ddots &&&&&&&&& \\ && 1 &&&&&&&& \\ &&& \cos\alpha &&&& -\sin\alpha &&& \\ &&&& 1 &&&&&& \\ &&&&& \ddots &&&&& \\ &&&&&& 1 &&&& \\ &&& \sin\alpha &&&& \cos\alpha &&& \\ &&&&&&&& 1 && \\ &&&&&&&&& \ddots & \\ &&&&&&&&&& 1 \end{bmatrix}$$

where the coordinates j, k *are the ones with the* $\cos\alpha$ *and* $\sin\alpha$ *terms. Geometrically, this corresponds to a rotation in the plane spanned by* $\mathbf{e}_j$ *and* $\mathbf{e}_k$ *of an angle of* α. *For* $d = 3$ *this can also be seen as the rotataion around the "remaining" axis.*

4. $\mathbf{y} = 0$, $\mathbf{A} = \mathbf{S}(\mathbf{W}) = \begin{bmatrix} \mathbf{I}_k & \mathbf{W} \\ & \mathbf{I}_{d-k} \end{bmatrix}$: ***shear****, where* $\mathbf{W} \in \mathbb{R}^{k\times d-k}$, $k < d$.

Exercise 2.2 Show that $\mathbf{S}(\mathbf{W})\,\mathbf{S}(\mathbf{W}') = \mathbf{S}(\mathbf{W} + \mathbf{W}')$ and prove that shears form an abelian subgroup of matrices. ◇

Definition 2.6 (Orthogonal matrices) *A matrix* $\mathbf{Q} \in \mathbb{R}^{d\times d}$ *is called* ***orthogonal*** *if* $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. *An affine transformation is called* ***euclidean map*** *if* $\mathbf{A}$ *is orthogonal. Normally, we will use the letter* $\mathbf{Q}$ *to denote orthogonal matrices.*

Exercise 2.3 Show that a matrix is orthogonal if and only if[8] its column vectors are orthonormal[9] ◇

Euclidean maps are the most relevant transformations Cartesian spaces as they provide a remarkable amount of structure.

[8]There is the abbreviation *iff* for "if and only if", invented by Halmos, see (Halmos, 1988).

[9]Which is the reason why orthogonal matrices are sometimes also called *orthonormal* matrices, but the obvious ambiguity persists nevertheless.

Theorem 2.7 (Euclidean maps) *Euclidean maps* $E_{\mathbf{Q},\mathbf{y}} : \mathbf{x} \mapsto \mathbf{Q}\mathbf{x} + \mathbf{y}$

1. *preserve distances.*
2. *form a (noncommuting) group.*

Proof: For 1) we choose $\mathbf{x}, \mathbf{x}'$ and consider

$$\begin{aligned}\left\| E^T_{\mathbf{Q},\mathbf{y}}(\mathbf{x}) - E^T_{\mathbf{Q},\mathbf{y}}(\mathbf{x}')\right\| &= (\mathbf{Q}\mathbf{x} + \mathbf{y} - \mathbf{Q}\mathbf{x}' - \mathbf{y})^T (\mathbf{Q}\mathbf{x} + \mathbf{y} - \mathbf{Q}\mathbf{x}' - \mathbf{y}) \\ &= (\mathbf{x} - \mathbf{x}')^T \underbrace{\mathbf{Q}^T\mathbf{Q}}_{=\mathbf{I}}(\mathbf{x} - \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|.\end{aligned}$$

For 2) we first have to show that any concatenation of two euclidean maps is euclidean again:

$$E_{\mathbf{Q}_2,\mathbf{y}_2}\left(E_{\mathbf{Q}_1,\mathbf{y}_1}(\mathbf{x})\right) = \mathbf{Q}_2(\mathbf{Q}_1\mathbf{x} + \mathbf{y}_1) + \mathbf{y}_2 = \underbrace{\mathbf{Q}_2\mathbf{Q}_1}_{:=\mathbf{Q}}\mathbf{x} + \underbrace{\mathbf{Q}_2\mathbf{y}_1 + \mathbf{y}_2}_{=:\mathbf{y}} \tag{2.9}$$

and $\mathbf{Q}$ is orthogonal since[10]

$$\mathbf{Q}^T\mathbf{Q} = (\mathbf{Q}_2\mathbf{Q}_1)^T(\mathbf{Q}_2\mathbf{Q}_1) = \mathbf{Q}_1^T\mathbf{Q}_2^T\mathbf{Q}_2\mathbf{Q}_1 = \mathbf{I}.$$

Moreover, we have to show that any euclidean map has an inverse which is, for $E_{\mathbf{Q},\mathbf{y}}$ the map $E_{\mathbf{Q}^T,-\mathbf{Q}^T\mathbf{y}}$ as simple substitution into (2.9) shows:

$$E_{\mathbf{Q}^T,-\mathbf{Q}^T\mathbf{y}}\left(E_{\mathbf{Q},\mathbf{y}}(\mathbf{x})\right) = \underbrace{\mathbf{Q}^T\mathbf{Q}}_{=\mathbf{I}}\mathbf{x} + \mathbf{Q}^T\mathbf{y} - \mathbf{Q}^T\mathbf{y} = \mathbf{x}.$$

The group is not **abelian**[11] since orthogonal matrices do not commute in general. □

Remark 2.8 *The group of euclidean maps is constructed from the interaction of two other groups: the nonabelian* $SO(n)$ *of orthogonal matrices and the abelian* $\mathbb{R}^d$ *of transformations. This way of combining two groups where one acts on the elements of the other, is called a* ***semidirect product*** *and is a construction that is used, for example, also in the context of wavelets.*

Exercise 2.4 Give an example of two orthogonal $d \times d$ matrices that do not commute. ◇

Remark 2.9 *Euclidean maps are a superset of what is called rigid motions, i.e., what we can do to some object when moving it around in 3D space around us. Indeed, a* ***rigid motion*** *is a euclidean map with* $\det \mathbf{Q} = 1$*, i.e, a map without reflections. Standard examples are euclidean maps based on rotations.*

[10] Orthogonal matrices form another important subgroup of $d \times d$ matrices, called $SO(d)$.

[11] The greatest honor for a mathematician is to have his/her name written in lowercase letters as a property. From that point of view, "Continuous" might be a name to go for.

Exercise 2.5 Show that the rigid motions form a subgroup of the euclidean maps. ◇

Remark 2.10 *Theorem 2.7 says that the distance between two points is an* ***invariant*** *under euclidean maps. This can be used to classify objects given as finite sets*[12] $\mathscr{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ *of points, often called a* ***point cloud****. Such data is collected, for example, by laser scanners. To classify objects, one can compute the* ***density***

$$\phi(d_-, d_+) = \frac{1}{n^2} \#\{1 \le j, k \le n : d_- \le \|\mathbf{x}_j - \mathbf{x}_k\| \le d_+\} \tag{2.10}$$

of distances in the interval $[d_-, d_+]$*. For sufficiently large point clouds, i.e.*[13]*, we can make these intervals very small and thus approximately get a density function* $\phi : \mathbb{R}_+ \to \mathbb{R}$ *that describes the object in a way that is invariant under rigid motions. In practical applications, one directly one uses* (2.10) *to compute a* ***histogram*** *based on threshold values* $d_0, d_1, \dots$*. If the thresholds are chosen relative to*

$$\max_{1 \le j,k \le n} \|\mathbf{x}_j - \mathbf{x}_k\|$$

then the method is even invariant under uniform scaling. Note also, that multiple points only contribute to $\phi(0)$ *which is a rather irrelevant number anyway.*

Affine transformations can also be used to do an affine change of coordinates. To that end, let $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{d \times d}$ be any nonsingular[14] matrix and $\mathbf{v}_0 \in \mathbb{R}^d$, then we can define a coordinate vector

$$\mathbf{v} = E_{\mathbf{V}^{-1}, -\mathbf{V}^{-1}\mathbf{v}_0}(\mathbf{x}) = \mathbf{V}^{-1}(\mathbf{x} - \mathbf{v}_0)$$

as another reference system for which $\mathbf{x} = E_{\mathbf{V},\mathbf{v}_0}$. With respect to tuple $\mathbf{v}$, the vectors $\mathbf{v}_j$ take the role of the unit vectors $\mathbf{e}_j$ in the standard coordinate system. This shows that any change between Cartesian coordinate systems can be seen as a euclidean map and vice versa.

There is one more operation that becomes particularly important for $d = 3$ but can be defined in a more general context.

Definition 2.11 (Vector product) *For* $\mathbf{x}_1, \dots, \mathbf{x}_{d-1} \in \mathbb{R}^d$ *the* ***vector product*** *or* ***cross product*** $\mathbf{x} \times \dots \times \mathbf{x}_{d-1}$ *is defined as the formal determinant*[15]

$$\mathbf{x}_1 \times \dots \times \mathbf{x}_{d-1} = \begin{vmatrix} \mathbf{e}_1 & x_{11} & \dots & x_{d-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e}_d & x_{1d} & \dots & x_{d-1,d} \end{vmatrix}, \tag{2.11}$$

[12] In this context, sets make more sense than multisets though we will see soon that also multisets can be handled that way.

[13] Explanation: *i.e.* stands for the latin phrase *id est*, meaning "that is", but in a more educated fashion.

[14] **Nonsingular** and **invertible** are synonymous though the first is a little bit more common than the latter.

[15] This means that in $\mathbb{R}^d$ a cross product always must have $d - 1$ factors.

which yields for $d = 3$ *the formula*

$$\begin{aligned}\mathbf{x} \times \mathbf{x}' &= (x_2 x_3' - x_3 x_2')\,\mathbf{e}_1 + (x_3 x_1' - x_1 x_3')\,\mathbf{e}_2 + (x_1 x_2' - x_2 x_1')\,\mathbf{e}_3 \\ &= \begin{bmatrix} x_2 x_3' - x_3 x_2' \\ x_3 x_1' - x_1 x_3' \\ x_1 x_2' - x_2 x_1' \end{bmatrix}. \end{aligned} \tag{2.12}$$

It follows from the standard rules for the determinant[16] that

$$\mathbf{x}_1 \times \cdots \times \mathbf{x}_j \times \cdots \times \mathbf{x}_k \times \cdots \times \mathbf{x}_{d-1} = -\,\mathbf{x}_1 \times \cdots \times \mathbf{x}_k \times \cdots \times \mathbf{x}_j \times \cdots \times \mathbf{x}_{d-1}$$

as well as

$$\mathbf{x}_1 \times \cdots \times \mathbf{x}_j \times \cdots \times (\lambda \mathbf{x}_j) \times \cdots \times \mathbf{x}_{d-1} = 0, \qquad \lambda \in \mathbb{R},$$

but we also have that[17]

$$\begin{aligned}\mathbf{x} \cdot (\mathbf{x}_1 \times \cdots \times \mathbf{x}_{d-1}) &= \left(\sum_{j=1}^{d} x_j \mathbf{e}_j \right)^T \left(\sum_{k=1}^{d} (-1)^k \mathbf{e}_k \det \mathbf{X}_k \right) = \sum_{j,k=1}^{d} (-1)^k x_j \underbrace{\mathbf{e}_j^T \mathbf{e}_k}_{=\delta_{jk}} \det \mathbf{X} \\ &= \det [\mathbf{x}\, \mathbf{x}_1 \dots \mathbf{x}_{d-1}] \end{aligned}$$

which is also called the **mixed product** of $\mathbf{x}$ and $\mathbf{x}_1, \dots, \mathbf{x}_{d-1}$, see (Kreyszig, 1959). This implies that

$$\mathbf{x}_1 \times \cdots \times \mathbf{x}_{d-1} \perp [\mathbf{x}_1 \dots \mathbf{x}_{d-1}]\, \mathbb{R}^{d-1} = \left\{ \sum_{j=1}^{d-1} \alpha_j\, \mathbf{x}_j : \alpha_j \in \mathbb{R} \right\} =: \operatorname{span}\{x_1, \dots, x_{d-1}\},$$

hence, the cross product provides a **normal** to the linear subspace of $\mathbb{R}^d$ spanned by $\mathbf{x}_1, \dots, \mathbf{x}_{d-1}$:

$$\mathbf{n} := \frac{\mathbf{x}_1 \times \cdots \times \mathbf{x}_{d-1}}{\|\mathbf{x}_1 \times \cdots \times \mathbf{x}_{d-1}\|} \tag{2.13}$$

satisfies $\mathbf{n}^T \mathbf{x} = 0$ for $\mathbf{x} \in \operatorname{span}\{x_1, \dots, x_{d-1}\}$. In particular we have for $d = 2$ the famous identity

$$\mathbf{n} := \frac{\mathbf{x} \times \mathbf{x}'}{\|\mathbf{x}_1 \mathbf{x} \times \mathbf{x}'\|}. \tag{2.14}$$

[16] Exchange to columns and the sign flips; the determinant is zero if two columns are multiples of each other.

[17] With the matrices

$$\mathbf{X}_k = \begin{bmatrix} x_{11} & \dots & x_{d-1,1} \\ \vdots & \ddots & \vdots \\ x_{1,k-1} & \dots & x_{d-1,k-1} \\ x_{1,k+1} & \dots & x_{d-1,k+1} \\ \vdots & \ddots & \vdots \\ x_{1d} & \dots & x_{d-1,d} \end{bmatrix}, \qquad k = 1, \dots, d$$

from the good old **Leibniz expansion** of the determinant.

This defines *the* normal to $\mathbf{x}$ and $\mathbf{x}'$ since

$$\mathbf{n}^T\mathbf{x} = \mathbf{n}^T\mathbf{x}' = 0 \qquad \text{and} \qquad \|\mathbf{n}\| = 1$$

only determines $\mathbf{n}$ up to sign while (2.14) fixes that as well and gives a preference to a certain direction. This can be used to define left-handed and right-handed coordinate systems.

Exercise 2.6 In (Pogorelov, 1987), the mixed product is called **scalar triple product** $(\mathbf{xyz}) : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, defined as

$$(\mathbf{xyz}) = \mathbf{x} \cdot (\mathbf{y} \times \mathbf{z}).$$

Show that

$$\mathbf{y} \cdot (\mathbf{x} \times \mathbf{z}) = -\mathbf{x} \cdot (\mathbf{y} \times \mathbf{z}) \qquad \text{and} \qquad \mathbf{z} \cdot (\mathbf{y} \times \mathbf{x}) = \mathbf{x} \cdot (\mathbf{y} \times \mathbf{z}).$$

$\diamond$

2.2 Projective coordinates and efficient affine operations

A somewhat disappointing thing about affine maps is that the two operations, multiplication by a matrix and addition of a vector, apparently play two different roles. This can be cured by a slightly more general approach.

Definition 2.12 *The **projective space** $\mathbb{P}^d$ is defined as all equivalence classes in $\mathbb{R}_* \times \mathbb{R}^n$, $\mathbb{R}_* = \mathbb{R} \setminus \{0\}$ being the set of all **units** in $\mathbb{R}$, cf. (Gathen & Gerhard, 1999; Sauer, 2001), with the equivalence*

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \equiv \begin{bmatrix} x_0' \\ x_1' \\ \vdots \\ x_d' \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} x_1/x_0 \\ \vdots \\ x_d/x_0 \end{bmatrix} = \begin{bmatrix} x_1'/x_0' \\ \vdots \\ x_d'/x_0' \end{bmatrix}. \tag{2.15}$$

The euclidean space $\mathbb{R}^d$ is embedded into $\mathbb{P}^d$ by

$$\mathbf{x} \hookrightarrow \hat{\mathbf{x}} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}. \tag{2.16}$$

Lemma 2.13 *In the projective setting, affine operations work as follows*

$$(\mathbf{A}\mathbf{x} + \mathbf{y})^\wedge = \begin{bmatrix} 1 & \\ \mathbf{y} & \mathbf{A} \end{bmatrix} \hat{\mathbf{x}}. \tag{2.17}$$

Proof: A simple[18] computation with block matrices[19] yields

$$\begin{bmatrix} 1 & \\ \mathbf{y} & \mathbf{A} \end{bmatrix} \hat{\mathbf{x}} = \begin{bmatrix} 1 & \\ \mathbf{y} & \mathbf{A} \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{y} + \mathbf{A}\mathbf{x} \end{bmatrix},$$

which completes the proof. □

Hence, in projective space any affine map can be expressed as a matrix multiplication with a $d+1 \times d+1$ matrix and these operations map the embedding of $\mathbb{R}^d$ back to $\mathbb{R}^d$. This way, we can easily represent and realize all our rigid motions which is actually how it is done on graphics cards or in computer graphics in general. In addition, this way of handling the data is also compatible with projective maps that enter the scenery as soon as perspective has to be computed, i.e., when 3D objects have to be represented on a 2D screen. Also, this is relevant in computer vision where pinhole cameras have to be modeled mathematically.

2.3 Affine geometry and barycentric coordinates

In this chapter we introduce a more geometric and intuitive coordinate system which will play a fundamental role for the definition of our geometric primitives, like free form curves and surfaces, later.

Let $\mathbf{x}, \mathbf{x}'$ be two points in $\mathbb{R}^d$, then the **line segment** connecting these two points can be written as

$$\ell(\mathbf{x}, \mathbf{x}') = \{(1-\alpha)\mathbf{x} + \alpha\mathbf{x}' : \alpha \in [0,1]\}. \tag{2.18}$$

This is called **linear interpolation** between $\mathbf{x}$ and $\mathbf{x}'$ or an **affine combination** of the two points. Let us write this in a more fancy way[20] as $u_0 := (1-\alpha)$, $u_1 := \alpha$, $\mathbf{u} = (u_0, u_1)$, then $\mathbf{u}$ always has the property that $\sum u_j = 1$ and

$$\ell(\mathbf{x}, \mathbf{x}') = \{[\mathbf{x}\,\mathbf{x}']\mathbf{u} : \mathbf{u} \in \mathbb{S}_1\} = [\mathbf{x}\,\mathbf{x}']\,\mathbb{S}_1, \qquad \mathbb{S}_1 := \{(u_0, u_1) : u_0, u_1 \ge 0, u_0 + u_1 = 1\}. \tag{2.19}$$

Moreover, the complete infinite **straight line** through $\mathbf{x}$ and $\mathbf{x}'$ can be written as

$$\ell_*(\mathbf{x}, \mathbf{x}') = [\mathbf{x}\,\mathbf{x}']\mathbb{A}_1, \qquad \mathbb{A}_1 = \{(u_0, u_1) : u_0 + u_1 = 0\} \tag{2.20}$$

and a point belongs to the connecting line iff $u_0, u_1 \ge 0$. For points "outside" one of the values has to be negative while the other one exceeds 1.

The notation from (2.19) and (2.20) can now be easily extended to arbitrarily many points $\mathbf{x}_0, \dots, \mathbf{x}_k$ with

$$\mathbb{A}_k := \left\{ \mathbf{u} = (u_0, \dots, u_k) \in \mathbb{R}^{k+1} : \sum_{j=1}^{k} u_j = 1 \right\}, \qquad \mathbb{S}_k := \{\mathbf{u} \in \mathbb{A}_k : \mathbf{u} \ge 0\} \tag{2.21}$$

[18] Yes, this proof is very simple. It is so simple because we make good use of notation which is something that's not only not forbidden but even required in mathematics. At least in serious math. And who would be interested in anything else?

[19] Computations with block matrices work just like with regular matrices, the only difference is that the "coefficient products" are also matrix-matrix or matrix-vector products. If the dimensions do not match – then something is wrong.

[20] Sometimes the shortest way of writing things may be the shortest but can be hard to generalize. So even fancy notations can be extremely helpful. Only when done right, of course.

and considering

$$\ell_*(\mathbf{X}) = \mathbf{X}\mathbb{A}_k, \qquad \ell(\mathbf{X}) = \mathbf{X}\mathbb{S}_k, \qquad \mathbf{X} := [\mathbf{x}_0 \dots \mathbf{x}_k]. \tag{2.22}$$

Definition 2.14 *The k–dimensional **plane** through $\mathbf{X}$ is $\ell_*(\mathbf{X})$ and the k–dimensional **simplex** $\ell(\mathbf{X})$. We also speak of the **affine hull***

$$[\![\mathbf{X}]\!]_* := \mathbf{X}\mathbb{A}_k \tag{2.23}$$

*and the **convex hull***

$$[\![\mathbf{X}]\!] := \mathbf{X}\mathbb{S}_k.$$

By definition, any point $\mathbf{x} \in \ell_*(\mathbf{X})$ can be written as $\mathbf{x} = \mathbf{X}\mathbf{u}$ for some $\mathbf{u} = \mathbf{u}(\mathbf{x}) \in \mathbb{A}_k$. We want to determine this value which is the solution of the linear system

$$\mathbf{x} = \mathbf{X}\mathbf{u}, \qquad 1 = u_0 + \cdots + u_k = \mathbf{1}_k^T \mathbf{u}$$

that can be conveniently combined into

$$\begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{1}^T \\ \mathbf{X} \end{bmatrix} \mathbf{u} = \begin{bmatrix} 1 & \dots & 1 \\ \mathbf{x}_0 & \dots & \mathbf{x}_k \end{bmatrix} \mathbf{u}$$

or, in projective terms,

$$\widehat{\mathbf{x}} = [\widehat{\mathbf{x}}_0 \dots \widehat{\mathbf{x}}_k]\mathbf{u} =: \widehat{\mathbf{X}}\,\mathbf{u}. \tag{2.24}$$

Definition 2.15 *The points $\mathbf{x}_0, \dots, \mathbf{x}_k$ are said to be **in general position** if the matrix $\widehat{\mathbf{X}}$ has rank $k+1$. Under these circumstances the **simplex** $\mathbf{X}\mathbb{S}_k$ is called **nondegenerate**. The tuple $\mathbf{u}(\mathbf{x}) \in \mathbb{A}_k$ such that $\mathbf{x} = \mathbf{X}\mathbf{u}(\mathbf{x})$ is called the **barycentric coordinates** of $\mathbf{x}$ with respect to $\mathbf{x}$.*

Barycentric coordinates are a rather classical concept and have already been studied in (Möbius, 1827) long before the time of CAD or CAGD.

Lemma 2.16 *If and only if the points $\mathbf{X}$ are in general position, then the barycentric coordinates $\mathbf{u}(\mathbf{x}) \in \mathbb{A}_k$ are unique for any $\mathbf{x} \in \ell_*(\mathbf{X})$.*

Proof: If the points are in general position, then there exists a square $k+1 \times k+1$ submatrix $\mathbf{Y}$ of $\mathbf{X}$ that $\mathbf{x}' = \mathbf{Y}\mathbf{u}$, where $\mathbf{x}' \in \mathbb{R}^{k+1}$ is the subvector of $\mathbf{x}$ that corresponds to $k+1$ linearly independent rows of $\widehat{\mathbf{X}}$. But this uniquely defines $\mathbf{u} = \mathbf{Y}^{-1}\mathbf{x}'$.

If, on the other hand, $\widehat{\mathbf{X}}$ does *not* have rank $k+1$, there exists $\mathbf{u}^* \neq 0$ such that

$$0 = \widehat{\mathbf{X}}\mathbf{u}^* = \begin{bmatrix} \mathbf{1}^T \\ \mathbf{X} \end{bmatrix} \mathbf{u}^* = \begin{bmatrix} \mathbf{1}^T\mathbf{u}^* \\ \mathbf{X}\mathbf{u}^* \end{bmatrix},$$

hence, in particular $u_0^* + \cdots + u_k^* = 0$ and therefore

$$\mathbf{u} + \alpha\mathbf{u}^* \in \mathbb{A}_k, \qquad \mathbf{u} \in \mathbb{A}_k,\ \alpha \in \mathbb{R}.$$

Since

$$\mathbf{X}\left(\mathbf{u}(\mathbf{x})+\alpha\mathbf{u}^*\right)=\mathbf{X}\mathbf{u}+\alpha\underbrace{\mathbf{X}\mathbf{u}^*}_{=0}=\mathbf{X}\mathbf{u}(\mathbf{x})=\mathbf{x},\qquad \mathbf{x}\in\ell_*(\mathbf{X}),\ \alpha\in\mathbb{R},$$

no point has unique barycentric coordinates. □

Next, we want to explain the name *barycentric*[21] of these coordinates as this actually gives some quite nice and iteresting insight. To that end, we assume that $k=d$ and that we have $d+1$ points $\mathbf{x}_0,\dots,\mathbf{x}_d$ in general position.

Example 2.17 (Unit simplex) *The intuition behind $d+1$ points in $\mathbb{R}^d$ in general position should always be the following: Imagine $\mathbf{x}_0$ as the corner point of the coordinate system and $\mathbf{x}_j$, $j=1,\dots,d$, as the points that sit at the end of the coordinate vectors $\mathbf{v}_j$, i.e.*

$$\mathbf{x}_j=\mathbf{x}_0+\mathbf{v}_j,\qquad j=1,\dots,d.$$

The standard coordinate system in this terminology gives

$$\mathbf{X}=[\mathbf{0}\,\mathbf{e}_1\dots\mathbf{e}_d]=[\mathbf{0}\,\mathbf{I}]=\begin{bmatrix}0&1&0&\dots&0\\0&0&\ddots&\ddots&\vdots\\\vdots&\vdots&\ddots&\ddots&0\\0&0&\dots&0&1\end{bmatrix}\in\mathbb{R}^{d\times d+1},$$

which is not invertible[22]*, but of course*

$$\widehat{\mathbf{X}}=\begin{bmatrix}1&1&\dots&1\\&1&&\\&&\ddots&\\&&&1\end{bmatrix}$$

is invertible, hence the origin and the unit vectors are indeed in general position[23]*. And any point $\mathbf{x}\in\mathbb{R}^d$ can then indeed be written as*

$$\mathbf{x}=\sum_{j=1}^d x_j\mathbf{e}_j=\underbrace{\left(1-\sum_{j=1}^d x_j\right)}_{u_0}\mathbf{0}+\sum_{j=1}^d\underbrace{x_j}_{=u_j}\mathbf{e}_j=\mathbf{X}\mathbf{u}(\mathbf{x}).$$

We call $[\![\mathbf{X}]\!]$ for this specific choice the ***unit simplex*** *in $\mathbb{R}^d$.*

Using **Cramer's rule**[24], cf. (Fischer, 1984; Marcus & Minc, 1965; Schneider & Barker, 1973), we can compute the **barycentric coordinates** of $\mathbf{x}\in\mathbb{R}^d$ as

$$u_j(\mathbf{x})=\frac{\det\left[\widehat{\mathbf{x}}_0\dots\widehat{\mathbf{x}}_{j-1}\,\widehat{\mathbf{x}}\,\widehat{\mathbf{x}}_{j+1}\dots\widehat{\mathbf{x}}_d\right]}{\det\left[\widehat{\mathbf{x}}_0\dots\widehat{\mathbf{x}}_d\right]},\qquad j=0,\dots,d, \tag{2.25}$$

[21] Greek "βαρνσ" = "heavy" (Meyer & Steinthal, 1973), which is reflected in "barometer", but **not** in "barista". Unfortunately, the closing "sigma", usually given as `\varsigma` in LaTeX, is not correct in the font used here.

[22] This is not even a square matrix!

[23] Which should not come as too much of a surprise.

[24] Sometimes this is a nice theoretic tool, but one should **never** solve linear systems by using determinants. This is almost like dividing by zero, maybe an even worse sin.

which motivates us to consider $\det \widehat{\mathbf{X}}$ a little more closely.

Theorem 2.18 (Volume formula) *For $\mathbf{X} \in \mathbb{R}^{d+1 \times d}$ in general position we have that*

$$\mathrm{vol} [\![\mathbf{X}]\!] = \frac{\left| \det \widehat{\mathbf{X}} \right|}{d!} \tag{2.26}$$

Proof: We begin by showing that the **unit simplex** has volume $\frac{1}{d!}$ which we will do by induction on d and which is clear for $d = 1$ where the unit simplex is the unit interval $[0,1]$ and

$$\mathrm{vol}\,[0,1] = \int_0^1 dx = 1.$$

For $\mathbf{X}_{d+1} := [\mathbf{0}\ \mathbf{I}_{d+1}] \in \mathbb{R}^{d+1 \times d+2}$, $d \geq 1$, we then have[25]

$$\begin{aligned}
\int_{[\![\mathbf{X}_{d+1}]\!]} dx &= \int_0^1 \int_0^{1-x_{d+1}} \cdots \int_0^{1-x_2-\cdots-x_{d+1}} dx_{d+1}\, dx_1 \dots dx_d\, dx_{d+1} \\
&= \int_0^1 \int_{(1-x_{d+1})[\![\mathbf{X}_d]\!]} d(x_1,\dots,x_d)\, dx_{d+1} = \int_0^1 (1-x_{d+1})^d \int_{[\![\mathbf{X}_d]\!]} dx_{d+1} \\
&= \frac{1}{d!}\int_0^1 (1-x)^d dx = \frac{1}{d!}\int_0^1 x^d dx = \frac{1}{d!}\, \frac{x^{d+1}}{d!}\bigg|_{x=0}^1 = \frac{1}{(d+1)!}.
\end{aligned}$$

For arbitrary points $\mathbf{X} \in \mathbb{R}^{d \times d+1}$ one integrates similarly over

$$\mathbf{x}_0 + \sum_{j=1}^d \alpha_j (\mathbf{x}_j - \mathbf{x}_0), \qquad 0 \leq \alpha_j, \quad \sum_{j=1}^d \alpha_j \leq 1,$$

and gets by a change of variables that

$$\int_{[\![\mathbf{X}]\!]} = |\det [\mathbf{x}_1 - \mathbf{x}_0 \dots \mathbf{x}_d - \mathbf{x}_0]| \underbrace{\int_{[\![\mathbf{X}_d]\!]} dx}_{=\frac{1}{d!}} = \frac{|\det [\mathbf{x}_1 - \mathbf{x}_0 \dots \mathbf{x}_d - \mathbf{x}_0]|}{d!}$$

and since[26]

$$\begin{aligned}
\det [\mathbf{x}_1 - \mathbf{x}_0 \dots \mathbf{x}_d - \mathbf{x}_0] &= \det \begin{bmatrix} 1 & 0 & \dots & 0 \\ \mathbf{x}_0 & \mathbf{x}_1 - \mathbf{x}_0 & \dots & \mathbf{x}_d - \mathbf{x}_0 \end{bmatrix} \\
&= \det \begin{bmatrix} 1 & 1 & \dots & 1 \\ \mathbf{x}_0 & \mathbf{x}_1 & \dots & \mathbf{x}_d \end{bmatrix} = \det \widehat{\mathbf{X}},
\end{aligned}$$

(2.26) follows. □

---

[25] Since

$$(1 - x_{d+1})[\![\mathbf{X}_d]\!] = \left\{ \mathbf{x} \in \mathbb{R}^d : \mathbf{x} \geq 0, \sum_{j=1}^d x_j \leq 1 - x_{d+1} \right\}$$

is exactly one slice of the simple for fixed x_d.

[26] The determinant is not changed when rows or columns are subtracted from other rows or columns.

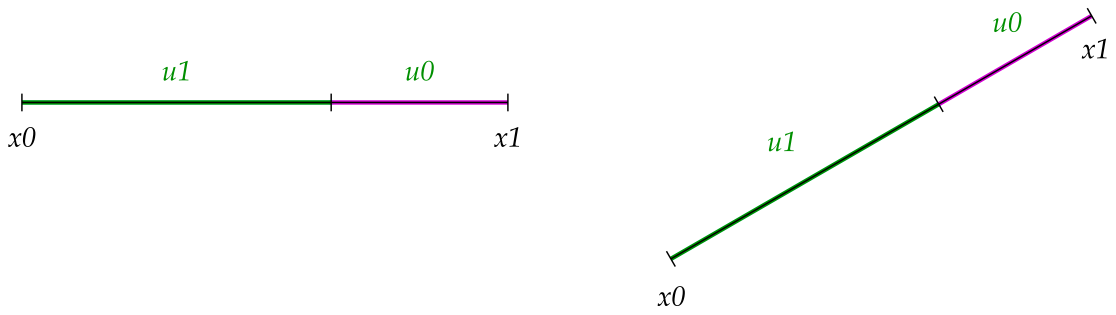

Figure 2.1: Barycentric coordinates with respect to a straight line in 2D. The position of the reference does not affect the relative volumes (lengths) of the subintervals and therefore the barycentric coordinates remain invariant.

Substituting (2.26) into (2.25), we now get the volume interpretation of barycentric coordinates:

$$u_j(\mathbf{x}) = \frac{\operatorname{vol}[\![\mathbf{x}_0 \dots \mathbf{x}_{j-1}\, \mathbf{x}\, \mathbf{x}_{j+1} \dots \mathbf{x}_d]\!]}{\operatorname{vol}[\![\mathbf{X}]\!]}, \qquad j = 0, \dots, d, \qquad \mathbf{x} \in [\![\mathbf{X}]\!]. \tag{2.27}$$

If $\mathbf{X}$ has less than $k + 1 \le d + 1$ columns, then (2.27) can still be extended, the only difference is that we consider the k–dimensional volume of the simplex $[\![\mathbf{X}]\!] = \mathbf{X}\mathbb{S}_k$, that is,

$$u_j(\mathbf{x}) = \frac{\operatorname{vol}_k[\![\mathbf{x}_0 \dots \mathbf{x}_{j-1}\, \mathbf{x}\, \mathbf{x}_{j+1} \dots \mathbf{x}_k]\!]}{\operatorname{vol}_k[\![\mathbf{X}]\!]}, \qquad j = 0, \dots, k, \qquad \mathbf{x} \in [\![\mathbf{X}]\!]. \tag{2.28}$$

This is shown in Fig. 2.1: the 1–dimensional volume of $[\![\mathbf{x}_0\, \mathbf{x}_1]\!]$ is the length of the interval and the barycentric coordinates are the relative lengths of the intervals *opposite* of the reference corner.

For the 2D case, the geometry of barycentric coordinates can be seen in Fig. 2.2. The point $\mathbf{x}$ splits the triangle into three subtriangles and the barycentric coordinates are the fraction of the total area that is covered by the respective triangle where one of the reference points is replaced by $\mathbf{x}$, which is the triangle *opposite* the reference point.

Now, the name is easily explained: the **barycenter** of the nondegenerate[27] k–dimensional simplex $[\![\mathbf{X}]\!]$, $\mathbf{X} \in \mathbb{R}^{d \times k+1}$ is the point $\mathbf{x}$ with barycentric coordinates $\mathbf{u}(x) = \left(\frac{1}{k+1}, \dots, \frac{1}{k+1}\right)$.

What makes barycentric coordinates elegant and useful is the fact that they are independent of position and scale of the coordinate system.

Proposition 2.19 (Invariance) *Barycentric coordinates[28] are invariant under affine maps with a nonsingular matrix* $\mathbf{A}$.

[27] From now on we automatically request the reference simplex to be nondegenerate when speaking of barycentric coordinates. This statement should actually be made in the text and not in a footnote, but this is a good way of checking if the reader also looks at footnotes.

[28] See footnote 27!

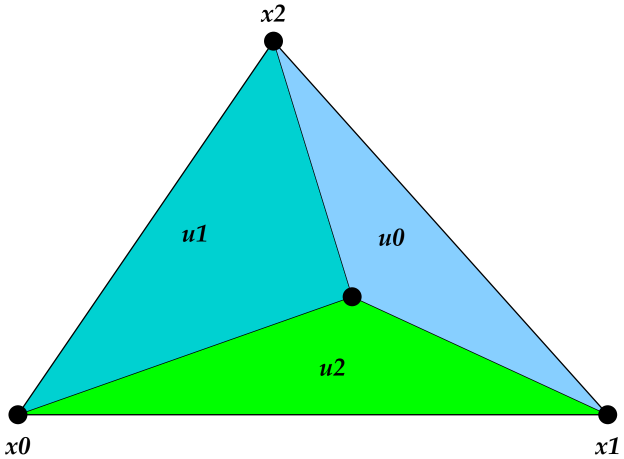

Figure 2.2: Barycentric coordinates with respect to a triangle. $\mathbf{x}$ is the point inside the triangle and the barycentric coordinates are the ratios between the area of the triangle opposite to the reference point and the area of the full triangle and the

Proof: Let $\mathbf{X} \in \mathbb{R}^{d \times k+1}$ be a nondegenerate reference simplex and

$$\mathbf{X}' := E_{\mathbf{A},\mathbf{y}}(\mathbf{X}) = \mathbf{A}\mathbf{X} + \mathbf{y}\mathbf{1}^T = [\mathbf{A}\mathbf{x}_0 + \mathbf{y} \dots \mathbf{A}\mathbf{x}_k + \mathbf{y}]$$

its image under the affine map. Now, if $\mathbf{x} = \mathbf{X}\mathbf{u}(\mathbf{x})$, then

$$\begin{aligned} E_{\mathbf{A},\mathbf{y}}(\mathbf{x}) &= \mathbf{A}\mathbf{x} + \mathbf{y} = \mathbf{A}\mathbf{X}\mathbf{u}(\mathbf{x}) + \mathbf{y} = \mathbf{A}\mathbf{X}\mathbf{u}(\mathbf{x}) + \mathbf{y}\ \underbrace{\mathbf{1}^T\mathbf{u}(\mathbf{x})}_{=1} \\ &= \left(\mathbf{A}\mathbf{X} + \mathbf{y}\mathbf{1}^T\right)\mathbf{u}(\mathbf{x}) = \mathbf{X}'\mathbf{u}(\mathbf{x}) \end{aligned}$$

and since barycentric coordinates are unique, this proves the claim, see Exercise 2.7. □

Exercise 2.7 Show that $E_{\mathbf{A},\mathbf{y}}(\mathbf{X})$ is nondegenerate if $\mathbf{X}$ is nondegenerate and $\mathbf{A}$ is nonsingular. ◇

Definition 2.20 (Affine space) *For a given nondegenerate reference system $\mathbf{X} \in \mathbb{R}^{d \times n+1}$, $n \le d$, we denote by*

$$\mathbb{E}_n := [\![\mathbf{X}]\!]_* = \mathbf{X}\,\mathbb{A}_n$$

*the **affine space** or **euclidean space** of dimension k and identify each point with its barycentric coordinates.*

It first seems artificial and somewhat strange to define $\mathbb{E}_n$ this way but it gives a clearer and more intuitive geometric flavor than working in the vector space $\mathbb{R}^n$. First note that affine combinations

$$\mathbf{u} = \sum_{j=1}^{k} \alpha_j\, \mathbf{u}_j, \qquad \sum_{j=1}^{k} \alpha_j = 1,$$

	$+\mathbb{E}_n$	$-\mathbb{E}_n$	$\pm\mathbb{E}'_n$	$\times\mathbb{R}$	aff.comb.
$\mathbb{E}_n$	-	$\mathbb{E}'_n$	$\mathbb{E}_n$	-	$\mathbb{E}_n$
$\mathbb{E}'_n$	$\mathbb{E}_n$	$\mathbb{E}_n$	$\mathbb{E}'_n$	$\mathbb{E}'_n$	$\mathbb{E}'_n$

Table 1: Admissible operations between $\mathbb{E}_n$ and $\mathbb{E}'_n$ and their results.

satisfy

$$\mathbf{1}^T\mathbf{u} = \sum_{j=1}^{k} \alpha_j \underbrace{\mathbf{1}^T\mathbf{u}_j}_{=1} = \sum_{j=1}^{k} \alpha_j = 1,$$

hence any affine combination of barycentric coordinates is barycentric coordinates again. On the other hand, $\lambda\mathbf{u}$, $\lambda \in \mathbb{R}$, does not make sense as then we loose the necessary property that barycentric coordinates sum to one. If we set

$$\mathbf{v} = \mathbf{u} - \mathbf{u}', \qquad \mathbf{u}, \mathbf{u}' \in \mathbb{A}_n,$$

then

$$\mathbf{1}^T\mathbf{v} = \mathbf{1}^T(\mathbf{u} - \mathbf{u}') = \mathbf{1}^T\mathbf{u} - \mathbf{1}^T\mathbf{u}' = 1 - 1 = 0,$$

so that $\mathbf{v}$ also does not satisfy the requirements for barycentric coordinates. Nevertheless, this type of objects is interesting as the difference between two points is the natural concept of a direction.

Definition 2.21 (Directions) $\mathbf{v} \in \mathbb{R}^{n+1}$ *is called a* ***direction*** *if* $\mathbf{1}^T\mathbf{v} = 0$. *The set of all directions is the vector space* $\mathbb{E}'_n$.

This separation between points and directions is what is behind the "arrow notion" of vectors that can be quite frequently found in the literature; in fact, to some extent directions only make sense as displacement between points, so the "arrow" is only relevant when fixed to a point and therefore pointing to a new one. Table 1 shows which operations are admissible between points and directions and what the results are. Moreover, this can be used to define linear and affine spaces in a concise way that is much simpler as the one often found in analysis, cf. (Sauer, 2015; Spivak, 1965).

Definition 2.22 *A* k *dimensional* ***affine subspace*** *given by* $(\mathbf{x}, \mathbf{Y})$, $\mathbf{x} \in \mathbb{E}_n$, $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_k] \in (\mathbb{E}'_n)^k$ *is*

$$\mathbb{E}_n \supseteq \mathbf{x} + \mathbf{Y}\mathbb{R}^k = \left\{ \mathbf{x} + \sum_{j=1}^{k} \alpha_j \, \mathbf{y}_j : \alpha \in \mathbb{R}^k \right\} \tag{2.29}$$

To be nondegenerate, the directions $\mathbf{y}_j$ *must be* ***linearly invariant***.

Exercise 2.8 Show that Definitions 2.20 and 2.22 are consistent, i.e., that for any k–dimensional nondegenerate affine subspaces $(\mathbf{x}, \mathbf{Y})$ of $\mathbb{R}^d$ there exists a nondegenerate reference system $\mathbf{X} \in \mathbb{R}^{d \times k-1}$ such that

$$\mathbf{x} + \mathbf{Y}\,\mathbb{R}^k = \mathbf{X}\,\mathbf{A}_k$$

and vice versa. ◊

Even if the next result is very easy to prove, it shows us the "different roles" of $\mathbb{R}^n$.

Theorem 2.23 (Cartesian vs. euclidean space) *The cartesian space $\mathbb{R}^d$ can be identified with $\mathbb{E}_d$ and $\mathbb{E}'_d$ in $\mathbb{R}^d$ by means of the reference system $\mathbf{X} = [\mathbf{0}\,\mathbf{e}_1 \dots \mathbf{e}_d] \in \mathbb{R}^{d \times d+1}$.*

Proof: Since

$$\begin{bmatrix} \mathbf{1}^T \\ \mathbf{X} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \mathbf{0} & \mathbf{e}_1 & \dots & \mathbf{e}_d \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

is invertible, any point $\mathbf{x} \in \mathbb{R}^d$ has unique barycentric coordinates

$$\mathbf{u}(\mathbf{x}) = \begin{bmatrix} \mathbf{1}^T \\ \mathbf{X} \end{bmatrix}^{-1} \widehat{\mathbf{x}} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} 1 - \mathbf{1}^T\mathbf{x} \\ \mathbf{x} \end{bmatrix}.$$

Hence, $\mathbf{x} \simeq \mathbf{u}(\mathbf{x}) = \begin{bmatrix} 1 - \mathbf{1}^T\mathbf{x} \\ \mathbf{x} \end{bmatrix}$. Taking a **direction** $\mathbf{y} = \mathbf{x} - \mathbf{x}'$ in $\mathbb{R}^d$, then

$$\mathbf{y} \simeq \mathbf{v}(\mathbf{y}) = \begin{bmatrix} 1 - \mathbf{1}^T\mathbf{x} \\ \mathbf{x} \end{bmatrix} - \begin{bmatrix} 1 - \mathbf{1}^T\mathbf{x}' \\ \mathbf{x}' \end{bmatrix} = \begin{bmatrix} -\mathbf{1}^T(\mathbf{x} - \mathbf{x}') \\ \mathbf{x} - \mathbf{x}' \end{bmatrix} = \begin{bmatrix} -\mathbf{1}^T\mathbf{y} \\ \mathbf{y} \end{bmatrix}.$$

In particular,

$$\mathbf{1}^T\mathbf{u}(\mathbf{x}) = 1 - \mathbf{1}^T\mathbf{x} + \mathbf{1}^T\mathbf{x} = 1 \qquad \text{and} \qquad \mathbf{1}^T\mathbf{v}(\mathbf{y}) = -\mathbf{1}^T\mathbf{y} + \mathbf{1}^T\mathbf{y} = 0.$$

□

Exercise 2.9 Prove that

$$\begin{bmatrix} 1 & \mathbf{1}^T \\ 0 & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -\mathbf{1}^T \\ 0 & \mathbf{I} \end{bmatrix}.$$

◊

Remark 2.24 *Theorem 2.23 shows how to embed $\mathbb{R}^d$ into $\mathbb{R}^{d+1}$ to obtain the barycentric representations for $\mathbb{E}_d$ and $\mathbb{E}'_d$.*

The barycentric approach can be used to do analysis with functions $f : \mathbb{E}_n \to \mathbb{R}$, especially to define derivatives. Indeed, let $\mathbf{x} \in \mathbb{E}_n$ and $\mathbf{y} \in \mathbb{E}'_n$ and consider the *difference*

$$\Delta_\mathbf{y} f(\mathbf{x}) := f(\mathbf{x} + \mathbf{y}) - f(\mathbf{x}).$$

Definition 2.25 (Directional derivative) *The* ***directional derivative*** $D_{\mathbf{y}} f(\mathbf{x})$ *at* $\mathbf{x} \in \mathbb{E}_n$ *for* $\mathbf{y} \in \mathbb{E}'_n$ *is defined as*

$$D_{\mathbf{y}} f(\mathbf{x}) := \lim_{h \to 0} \frac{\Delta_{h\mathbf{y}} f}{h} = \lim_{h \to 0} \frac{f(\mathbf{x} + h\mathbf{y}) - f(\mathbf{x})}{h}, \tag{2.30}$$

provided the limit exists in which case we call f ***directionally differentiable*** *at* $\mathbf{x}$ *in the direction* $\mathbf{y}$*.* f *is called* ***continously differentiable*** *if* $(\mathbf{x}, \mathbf{y}) \mapsto D_{\mathbf{y}} f(\mathbf{x})$ *is a continuous function on* $\mathbb{E}_n \times \mathbb{E}'_n$*.*

To get an idea about how to handle the geometric objects, let us reprove a classical observation from Analysis (Sauer, 2015).

Proposition 2.26 *If* f *is continuously differentiable then* $D_{\mathbf{y}} f(\mathbf{x})$ *is a* ***linear map*** *in* $\mathbf{y}$*, that is*

$$D_{\alpha\mathbf{y}+\beta\mathbf{y}'} f(\mathbf{x}) = \alpha D_{\mathbf{y}} f(\mathbf{x}) + \beta D_{\mathbf{y}'} f(\mathbf{x}). \tag{2.31}$$

Proof: Homogeneity of the directional derivative is easily proved:

$$\begin{aligned} D_{\alpha\mathbf{y}} f(\mathbf{x}) &= \lim_{h \to 0} \frac{f(\mathbf{x} + h\alpha\mathbf{y}) - f(\mathbf{x})}{h} = \alpha \lim_{h \to 0} \frac{f(\mathbf{x} + h\alpha\mathbf{y}) - f(\mathbf{x})}{h\alpha} \\ &= \alpha \lim_{h' \to 0} \frac{f(\mathbf{x} + h'\mathbf{y}) - f(\mathbf{x})}{h'} = \alpha D_{\mathbf{y}} f(\mathbf{x}). \end{aligned}$$

For linearity, we consider

$$\begin{aligned} D_{\mathbf{y}+\mathbf{y}'} f(\mathbf{x}) &= \lim_{h \to 0} \frac{f\left(\mathbf{x} + h(\mathbf{y} + \mathbf{y}')\right) - f(\mathbf{x})}{h} \\ &= \lim_{h \to 0} \frac{1}{h} \left(f\left(\mathbf{x} + h(\mathbf{y} + \mathbf{y}')\right) - f(\mathbf{x} + h\mathbf{y}) + f(\mathbf{x} + h\mathbf{y}) - f(\mathbf{x}) \right) \\ &= \lim_{h \to 0} \frac{f\left((\mathbf{x} + h\mathbf{y}) + h\mathbf{y}'\right) - f(\mathbf{x} + h\mathbf{y})}{h} + \lim_{h \to 0} \frac{f(\mathbf{x} + h\mathbf{y}) - f(\mathbf{x})}{h} \\ &= D_{\mathbf{y}'} f(\mathbf{x}) + D_{\mathbf{y}} f(\mathbf{x}), \end{aligned}$$

where for the first term we use the differentiability. □

Exercise 2.10 Show that for any differentiable f one has

$$f(\mathbf{x} + \mathbf{y}) - f(\mathbf{x}) = \int_0^1 D_{\mathbf{y}} f(\mathbf{x} + t\mathbf{y})\, dt$$

and use that to complete the proof of Proposition 2.26. ◊

We will get back to the issue of barycentric coordinates when considering piecewise curves and surfaces.

Welcher aber ... durch die Geometria sein Ding beweist und die gründliche Wahrheit anzeigt, dem soll alle Welt glauben. Denn da ist man gefangen.

A. Dürer

Differential geometry | 3

If we want to work with curves and surfaces, we have to understand some of the mathematical backgrounds for these objects. Since we will mostly deal *locally* with *smooth* objects, **differential geometry** will be the proper context.

3.1 Curves

We begin our "poor man's differential geometry" by collecting some basic facts about curves where we will mostly follow (Farin, 1988; Kreyszig, 1959).

Definition 3.1 (Curve) *A (parametric)* ***curve*** *in* $\mathbb{R}^d$ *is a function* $\mathbf{f} : [a, b] \to \mathbb{R}^d$ *from a* ***parameter invterval*** *to* $\mathbb{R}^d$*, hence*

$$\mathbf{f}(t) = \begin{bmatrix} f_1(t) \\ \vdots \\ f_d(t) \end{bmatrix}, \qquad t \in I := [a, b]. \tag{3.1}$$

A curve is called ***continuous*** *or* ***differentiable*** *of some order if alle coefficient functions* f_j *are continuous or differentiable. In particular, the* p*th* ***derivative*** $\mathbf{f}^{(p)}$ *of a sufficiently smooth*[29] *curve is defined as*[30]

$$\mathbf{f}^{(p)} = \left[f_j^{(p)} : j = 1, \dots, n \right] = \begin{bmatrix} f_1^{(p)} \\ \vdots \\ f_d^{(p)} \end{bmatrix}. \tag{3.2}$$

We will write $\dot{\mathbf{f}}$ *and* $\ddot{\mathbf{f}}$ *for the first and second derivative of* $\mathbf{f}$ *with respect to* t.

A curve $\mathbf{f}$ *is called* ***regular*** *if* $\mathbf{f}$ *is differentiable and* $\dot{\mathbf{f}} \neq 0$.

Exercise 3.1 Show that the definition of a derivative for curves is consistent:

$$\dot{\mathbf{f}}(t) = \begin{bmatrix} \dot{f}_1(t) \\ \vdots \\ \dot{f}_d(t) \end{bmatrix} = \lim_{h \to 0} \frac{1}{h} \left(\mathbf{f}(t + h) - \mathbf{f}(t) \right).$$

◊

[29] That means that all derivatives up to order up to p exist for all component functions.
[30] This is more an explanation of the notation.

Remark 3.2 *The terminology for curves is not unique: In some part of the literature the curve is the function* $[a,b] \to \mathbb{R}^d$*, for others the curve is the* set $\mathbf{f}([a,b]) \subset \mathbb{R}^d$ *and the function* $\mathbf{f}$ *is then called a* ***parametrization*** *of the curve. Moreover, differentiability of a curve does not mean that its image is a smooth object, too, see Pic. 3.1.*

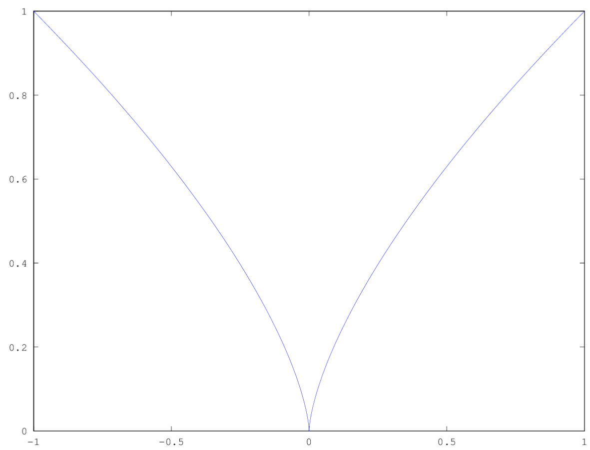

Figure 3.1: **Neil's parabola**

$$\mathbf{f}(t) = \begin{pmatrix} t^3 \\ t^2 \end{pmatrix}, \qquad t \in [-1,1],$$

is a smooth curve in the sense that $\mathbf{f}$ has derivatives of any order, but its image is not smooth as it has a visual cusp at $t = 0$. Note that the curve is not regular there..

3.1.1 Reparametrizations

Definition 3.3 (Reparametrization) *A* ***reparametrization*** *of a curve* $\mathbf{f} : [a,b] \to \mathbb{R}^d$ *is a function* $\varphi \in C^1(I')$, $I' := [a', b']$*, such that*

$$\dot{\varphi} > 0 \qquad \textit{and} \qquad \varphi(I') = I. \tag{3.3}$$

A C^k *reparamterization is a reparametrization with* $\varphi \in C^k(I')$.

Of course, a reparametrization does not change the curve as a set as clearly

$$(\mathbf{f} \circ \varphi)(I') = \mathbf{f}\left(\varphi(I')\right) = \mathbf{f}(I).$$

Definition 3.4 *For a differentiable curve* $\mathbf{f}$ *and* $a \le u \le v \le t$ *the* ***arc length*** *of the* ***curve segement*** *from* u *to* v *is defined as*

$$L_{[u,v]}\mathbf{f} := \int_u^v \|\dot{\mathbf{f}}(t)\| \, dt. \tag{3.4}$$

It can be easily shown, see (Sauer, 2015), that the arc length of the segment is equivalently described as

$$\lim_{t_{j+1}-t_j\to 0} \sum_{j=1}^{N} \|\mathbf{f}(t_j) - \mathbf{f}(t_{j-1})\|, \qquad u = t_0 < t_1 < \cdots < t_N = v$$

which is the length of the **piecewise linear function** that passes through the points $\mathbf{f}(t_j)$, hence, in the limit the **length** of the curve. Since

$$L_{[a,u]}\mathbf{f} = \int_a^u \|\dot{\mathbf{f}}(t)\|\,dt = \int_a^{u'} \|\dot{\mathbf{f}}(t)\|\,dt + \int_{u'}^{u} \|\dot{\mathbf{f}}(t)\|\,dt = L_{[a,u']}\mathbf{f} + L_{[u',u]}\mathbf{f},$$

the function

$$\ell(u) = \int_a^u \|\dot{\mathbf{f}}(t)\|\,dt$$

is **strictly monotonically increasing** whenever $\mathbf{f}$ is regular and $\ell := \ell(b)$ is the arc length of the full curve on $[a,b]$.

Exercise 3.2 Show that for any regular curve the function $u \mapsto \ell(u)$ is differentiable on I and $\dot{\ell} \neq 0$. $\diamond$

If $\mathbf{f}$ is a **regular curve** then $\varphi := \ell^{-1} : [0,\ell] \to [a,b]$ is differentiable with

$$\varphi'(s) := \frac{d\varphi}{ds}(s) = \frac{1}{\dot{\ell}(u)} = \frac{1}{\|\dot{\mathbf{f}}(u)\|} = \frac{1}{\left\|\dot{\mathbf{f}}(u)\right\|}, \qquad s = \ell(u),$$

and therefore

$$\left\|(\mathbf{f}\circ\varphi)'(s)\right\| = \|\dot{\mathbf{f}}(\varphi(\ell(u)))\,\varphi'(s)\| = \frac{\|\dot{\mathbf{f}}(u)\|}{\|\dot{\mathbf{f}}(u)\|} = 1 \tag{3.5}$$

and

$$L_{[0,s]}(\mathbf{f}\circ\varphi) = \int_0^s \underbrace{\|(\mathbf{f}\circ\varphi)'(\sigma)\|}_{=1}\,d\sigma = \int_0^s d\sigma = s, \qquad s \in [0,\ell], \tag{3.6}$$

because of which this special parametrization is called the **arc length parametrization** of $\mathbf{f}$. This is the most natural parametrization of a curve.

Definition 3.5 *The **tangent** of a regular curve* $\mathbf{f}$ *at* $u \in I$ *is the vector* $\dot{\mathbf{f}}(u)$*, the **tangent curve** is* $\mathbf{t_f} : t \to \dot{\mathbf{f}}$*.*

If $\mathbf{f}$ is parametrized with respect to the arc length, then $\mathbf{t_f}$ satisfies $\|\mathbf{t_f}(t)\| = 1$, $t \in I$. From now on, we denote by $\mathbf{f}(s)$ the arc length parametrization[31] of $\mathbf{f}$ and by $\mathbf{f}', \mathbf{f}''$, and so on, the derivatives of this function with respect to s.
From

$$1 = \|\mathbf{f}'\|^2 = (\mathbf{f}')^T\mathbf{f}' \qquad \Rightarrow \qquad 0 = \frac{d}{ds}\|\mathbf{f}'\|^2 = (\mathbf{f}'')^T\mathbf{f}' + (\mathbf{f}')^T\mathbf{f}'' = 2(\mathbf{f}'')^T\mathbf{f}',$$

[31] Which is, to be precise $\mathbf{f}(\varphi(s))$

it follows immediately that

$$\mathbf{f}'' \perp \mathbf{f}', \tag{3.7}$$

the second derivative with respect to arc length is perpendicular to the unit tangent. To that end,

$$\mathbf{n} := \frac{\mathbf{f}''}{\|\mathbf{f}''\|}$$

is called the **principal normal**[32] of $\mathbf{f}$ and is a unit vector perpendicular to the normal. The value

$$\kappa(s) := \|\mathbf{f}''(s)\|, \qquad s \in [0, \ell], \tag{3.8}$$

is called the **curvature** of $\mathbf{f}$ at s and

$$\rho(s) = \frac{1}{\kappa(s)}$$

the **radius of curvature**. Note that the curvature is an **intrinsic** value of the curve that does *not* depend on the (initial) parametrisation of $\mathbf{f}$ since we switched to the arc length parametrization before computing this value. Since

$$\mathbf{f}''(s) = \ddot{\mathbf{f}}(t)\,\varphi'(s) + \dot{\mathbf{f}}(t)\,\varphi''(s), \qquad t = \varphi(s) = \ell^{-1}(s),$$

the curvature is not really easy to compute for a general parametrization. There are interesting concepts of higher order like the **torsion** of a space curve which can be found for example in (Kreyszig, 1959), but would lead to far here.

3.1.2 Distances

The next question concerns the **distance** between two curves $\mathbf{f}, \mathbf{g}$, where we assume that they are defined over the same parameter interval I. We are only interested in "worst case" distances where we have the following options:

Parametric distance: Here one considers

$$d\,(\mathbf{f}, \mathbf{g}) := \max_{u \in I} \left\| \mathbf{f}(u) - \mathbf{g}(u) \right\|, \tag{3.9}$$

which of course requires the curves to be parametrized identically and depends strongly on the respective parametrizations.

Parameter free distance: Since intrinsically the parametrization does not matter, we could build it into the definition:

$$d\,(\mathbf{f}, \mathbf{g}) := \min_{\dot{\varphi} > 0} \max_{u \in I} \left\| \mathbf{f}_\varphi(u) - \mathbf{g}(u) \right\|, \tag{3.10}$$

where $\varphi : I \to I$ should be a regular paramterization. This is very geometric, but notrivial to compute as determining φ leads to a nonlinear **variational problem** which is not so easy to solve even if there are methods, cf. (Gelfand & Fomin, 1963; Kirk, 1970; Stengel, 1986).

[32] And again direction matters!

Arc length distance: somewhere in the middle one could use reweighted arc length parametrization, scaled linearly according to the ration between $\ell_\mathbf{f}$ and $\ell_\mathbf{g}$

$$d(\mathbf{f}, \mathbf{g}) := \max_{0 \le s \le \ell_\mathbf{f}} \left\| \mathbf{f}(s) - \mathbf{g}\left(\frac{\ell_\mathbf{g}}{\ell_\mathbf{f}} s\right) \right\|. \tag{3.11}$$

Especially when the curves are intially parameterized with respect to the arc length, this is a good method.

Hausdorff distance: for any point $\mathbf{f}(u)$ on the curve $\mathbf{f}(I)$ one chooses the closest point on $\mathbf{g}$ and maximizes this value:

$$\max_{u \in I} \min_{u' \in I} \left\| \mathbf{f}(u) - \mathbf{g}(u') \right\|.$$

Since min und max may not be interchanged so easily, we symmetrize the whole thing

$$d(\mathbf{f}, \mathbf{g}) := \max \left\{ \max_{u \in I} \min_{u' \in I} \left\| \mathbf{f}(u) - \mathbf{g}(u') \right\|, \max_{u' \in I} \min_{u \in I} \left\| \mathbf{f}(u) - \mathbf{g}(u') \right\| \right\}. \tag{3.12}$$

The drawback is again that this expression is hard to compute.

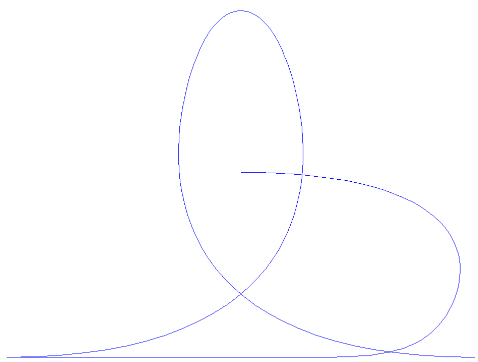

Figure 3.2: Two curves for which all the distance concepts lead to different results.

The choice of the distance concept depends very much on the application. Parametric distance is natural if the paramterizations are meaningful, Hausdorff distance is the most reasonable thing in terms of geometry of point sets.

3.1.3 Local frames and the difference between plane and space

Next, we return to our differential geometry and consider the difference between plane and space curves, i.e., $d = 2$ and $d = 3$, a little bit more carefully, see again (Kreyszig, 1959; Struik, 1961).

For $\mathbf{f} : I \to \mathbb{R}^d$, $d = 2, 3$, we consider a **Taylor expansion** of $\mathbf{f}$ around $u \in I$ and get for $\delta \in \mathbb{R}$,

$$\mathbf{f}(u+\delta) = \mathbf{f}(u) + \delta\,\dot{\mathbf{f}}(u) + \frac{\delta^2}{2}\ddot{\mathbf{f}}(u) + \frac{\delta^3}{3}\dddot{\mathbf{f}}(u) + \cdots = \sum_{j=0}^{\infty} \frac{\delta^j}{j!}\,\mathbf{f}^{(j)}(u),$$

where the derivatives of order higher than 3 will be ingored. Next, we *request* that the first d derivative $\mathbf{f}^{(j)}$, $j = 1, \dots, d$, are **linearly independent**. Then the form a local **coordinate system** that can be orthogonalized by means of the **Gram–Schmidt method**:

$$\begin{aligned}
\mathbf{t} &:= \frac{\dot{\mathbf{f}}(u)}{\left\|\dot{\mathbf{f}}(u)\right\|_2}, \\
\widetilde{\mathbf{n}} &:= \ddot{\mathbf{f}}(u) - \mathbf{t}\,\mathbf{t}^T\ddot{\mathbf{f}}(u) = \ddot{\mathbf{f}}(u) - \frac{\dot{\mathbf{f}}^T(u)\ddot{\mathbf{f}}(u)}{\left\|\dot{\mathbf{f}}(u)\right\|_2^2}\dot{\mathbf{f}}(u) \\
&= \frac{\ddot{\mathbf{f}}(u)\,\dot{\mathbf{f}}^T(u)\dot{\mathbf{f}}(u) - \dot{\mathbf{f}}(u)\,\ddot{\mathbf{f}}^T(u)\dot{\mathbf{f}}(u)}{\left\|\dot{\mathbf{f}}(u)\right\|_2^2} = \frac{\left(\mathbf{F}(u) - \mathbf{F}^T(u)\right)\dot{\mathbf{f}}(u)}{\left\|\dot{\mathbf{f}}(u)\right\|_2^2}, \qquad \mathbf{F} := \ddot{\mathbf{f}}\dot{\mathbf{f}}^T, \\
\mathbf{n} &:= \frac{\widetilde{\mathbf{n}}}{\left\|\widetilde{\mathbf{n}}\right\|_2} = \frac{\dot{\mathbf{f}}^T(u)\dot{\mathbf{f}}(u)\,\ddot{\mathbf{f}}(u) - \dot{\mathbf{f}}^T(u)\ddot{\mathbf{f}}(u)\,\dot{\mathbf{f}}(u)}{\left\|\dot{\mathbf{f}}^T(u)\dot{\mathbf{f}}(u)\,\ddot{\mathbf{f}}(u) - \dot{\mathbf{f}}^T(u)\ddot{\mathbf{f}}(u)\,\dot{\mathbf{f}}(u)\right\|_2} = \frac{\left(\mathbf{F}(u) - \mathbf{F}^T(u)\right)\dot{\mathbf{f}}(u)}{\left\|\left(\mathbf{F}(u) - \mathbf{F}^T(u)\right)\dot{\mathbf{f}}(u)\right\|_2}, \\
\mathbf{b} &:= \frac{\dot{\mathbf{f}}(u) \times \ddot{\mathbf{f}}(u)}{\left\|\dot{\mathbf{f}}(u) \times \ddot{\mathbf{f}}(u)\right\|_2}.
\end{aligned}$$

In $\mathbb{R}^3$ we again use the **vector product** from (2.12),

$$\mathbf{x} \times \mathbf{y} = \det \begin{bmatrix} \mathbf{e}_1 & x_1 & y_1 \\ \mathbf{e}_2 & x_2 & y_2 \\ \mathbf{e}_3 & x_3 & y_3 \end{bmatrix} = \begin{bmatrix} x_2y_3 - x_3y_2 \\ x_3y_1 - x_1y_3 \\ x_1y_2 - x_2y_1 \end{bmatrix}$$

definiert ist. Die **Tangente t** we already know as $\mathbf{f}'(s)$, $s = s(u)$ and because of the also known argument

$$1 = \left\|\mathbf{f}'(s)\right\|_2^2 = \left(\mathbf{f}'(s)\right)^T \mathbf{f}'(s) \qquad \Rightarrow \qquad 0 = \frac{d}{ds}\left(\left(\mathbf{f}'(s)\right)^T \mathbf{f}'(s)\right) = 2\left(\mathbf{f}'(s)\right)^T \mathbf{f}''(s)$$

the **normal n** must be a multiple of $\mathbf{f}''$ as

$$\frac{d^2}{ds^2}\mathbf{f}(u(s)) = \frac{d}{ds}\left(\dot{\mathbf{f}}(u(s))\;u'(s)\right) = \ddot{\mathbf{f}}(u(s))\;\left(u'(s)\right)^2 + \dot{\mathbf{f}}(u(s))\;u''(s) \tag{3.13}$$

lies in the span of $\dot{\mathbf{f}}$ and $\ddot{\mathbf{f}}$ and is perpendicular to $\mathbf{t}$, hence also to $\dot{\mathbf{f}}$. The **binormal b** only exists for $d = 3$ and is defined by being perpendicular to $\dot{\mathbf{f}}$ and $\ddot{\mathbf{f}}$.

Exercise 3.3 Show that in $\mathbb{R}^3$ the normal can be computed as

$$\mathbf{n} = \mathbf{b} \times \mathbf{t} = -\frac{\dot{\mathbf{f}} \times \left(\dot{\mathbf{f}} \times \ddot{\mathbf{f}}\right)}{\left\|\dot{\mathbf{f}}\right\|_2 \left\|\dot{\mathbf{f}} \times \ddot{\mathbf{f}}\right\|_2}.$$

Hint: use properties of the vector product. $\diamond$

Definition 3.6

1. *The vectors* $\mathbf{t}, \mathbf{n}, \mathbf{b}$ *form the* ***Frenet frame***[33] *at* u *and give natural and intrinsic local coordinate system at* u.
2. *The* ***osculating plane***[34] *is the affine plane through* $\mathbf{f}(u)$ *spanned by* $\mathbf{t}$ *und* $\mathbf{n}$. *In* $\mathbb{R}^2$ *it is meaningless.*

It is time to illustrate these concepts by means of a (very) simple example.

Example 3.7 *For* $I = [0, 1]$ *we consider the line segment*

$$\mathbf{f}(u) = \mathbf{a}\, u + \mathbf{b}, \qquad \mathbf{0} \neq \mathbf{a}, \mathbf{b} \in \mathbb{R}^d, \qquad d = 2, 3. \tag{3.14}$$

The tangents are $\mathbf{a}/\|\mathbf{a}\|_2$ *and of course*[35] *the curvature is zero as we can see by taking derivatives*

$$\dot{\mathbf{f}}(u) = \mathbf{a}, \qquad \ddot{\mathbf{f}}(u) = \mathbf{0}.$$

Now we use the regular[36] *reparametrization* $\varphi(u) = u^2$, *so that* $\mathbf{f}_\varphi(u) = \mathbf{a}u^2 + \mathbf{b}$, *hence*

$$\dot{\mathbf{f}}_\varphi = 2\mathbf{a}u, \qquad \ddot{\mathbf{f}}_\varphi = 2\mathbf{a} \neq \mathbf{0}.$$

The curvature is still zero since $\dot{\mathbf{f}}_\varphi$ *and* $\ddot{\mathbf{f}}_\varphi$ *are* linearly dependent *and the normal as "direction of curvature" is part of* $\ddot{\mathbf{f}}_\varphi$ *which is perpendicular to* $\dot{\mathbf{f}}_\varphi$.

Exercise 3.4 What is the arc length paramterization of $\mathbf{f}$ from (3.14)? ◇

3.1.4 Connecting curves

Most objects used in Geometric Modeling are of a *piecewise* nature, i.e., consist of various **curve pieces** or **surface patches**. Let us start simple with curves and let us consider two curves $\mathbf{f} : [t_0, t^*]$ and $\mathbf{g} : [t^*, t_1]$ which are combined into the **composite curve**

$$\mathbf{c}(t) = \begin{cases} \mathbf{f}(t), & t \in [t_0, t^*], \\ \mathbf{g}(t), & t \in [t^*, t_1], \end{cases} \qquad t \in [t_0, t_1] \tag{3.15}$$

and its behavior at t^*. If the values differ at t^*, it is not reasonable to speak of a composite curve, hence we always request that

$$\mathbf{f}(t^*) = \mathbf{g}(t^*). \tag{3.16}$$

Differentiability is getting more interesting. We could first require that, in addition to (3.16), we also have

$$\dot{\mathbf{f}}(t^*) = \dot{\mathbf{g}}(t^*) \tag{3.17}$$

[33] In German **Frenet–Dreibein**.

[34] In German **Schmiegeebene**, the name is due to (one of the) Johann Bernoulli(s), or, as written in (Struik, 1961), "John Bernoulli".

[35] *Intuitively* this is clear, but nevertheless this is no replacement for a proof.

[36] The zero of $\dot{\varphi}$ at $u = 0$ is irrelevant, and otherwise one could use $\varphi(u) = (1 - \varepsilon)u^2 + \varepsilon u$, $0 < \varepsilon < 1$, which just complicates the computations

which is, however, not intrinsic since

$$\dot{\mathbf{f}}_\varphi(t^*) = \frac{d}{dt}\left(\mathbf{f} \circ \varphi\right)(t^*) = \dot{\mathbf{f}}\left(\varphi(t^*)\right)\ \dot{\varphi}(t^*)$$

which violates (3.17) whenever $\dot{\varphi}(t^*) \neq 1$. Hence, differentiability is *not* intrinsic. This motivates the introduction of **geometric differentiabiliy**, abbreviated as G^1 where we request that

$$\mathbf{f}'(t^*) = \frac{\dot{\mathbf{f}}(t^*)}{\left\|\dot{\mathbf{f}}(t^*)\right\|_2} = \frac{\dot{\mathbf{g}}(t^*)}{\left\|\dot{\mathbf{g}}(t^*)\right\|_2} = \mathbf{g}'(t^*), \tag{3.18}$$

i.e., $\dot{\mathbf{f}}(t^*)$ and $\dot{\mathbf{f}}(t^*)$ are **collinear**.

Definition 3.8 (G^1) *The curves* $\mathbf{f}$ *and* $\mathbf{g}$ *join* G^1 ***smooth*** *at* t^* *if either of following three equivalent conditions is satisfied:*

1. *The unit tangents at* t^* *coincide, i.e., the tangents pint in the same direction.*
2. *The curve* $\mathbf{f}$ *can be reparameterized regularly such that* $\mathbf{f} \circ \varphi$ *and* $\mathbf{g}$ *join* C^1 *smooth at* t^*.
3. *If both curves are parametrized with respect to their arc length, they join differentiably.*

Even more interesting is the case of second order differentiability. Parametrically, this is easy, one just demands

$$\ddot{\mathbf{f}}(t^*) = \ddot{\mathbf{g}}(t^*) \tag{3.19}$$

in addition to (3.16) and (3.17). Geometrically, this would be

$$\mathbf{f}'(t^*) = \mathbf{g}'(t^*) \qquad \text{and} \qquad \mathbf{f}''(t^*) = \mathbf{g}''(t^*). \tag{3.20}$$

Rewriting the first of these conditions as

$$\mathbf{g}'(t^*) = \dot{\mathbf{f}}(t^*)\ t', \qquad t' = t'\left(s\left(t^*\right)\right) = \left\|\dot{\mathbf{f}}(t^*)\right\|_2^{-1}, \tag{3.21}$$

and the second as

$$\mathbf{g}''(t^*) = \ddot{\mathbf{f}}(t^*)\ (t')^2 + \dot{\mathbf{f}}(t^*)\ t'', \tag{3.22}$$

we see that $\mathbf{g}''$ has to lie in the osculating plane of $\mathbf{f}$ at t^* which is trivial for $d = 2$ but a real constraint for space curves. Note that (3.22) is a nonlinear equation in t' and t'' which can not really be solved. Therefore, the following definition of second order geometric differentiability is normally used.

Definition 3.9 (G^2) *Two parametric curves* $\mathbf{f}$ *and* $\mathbf{g}$ *join* G^2 ***smooth*** *at* t^* *if they join differentiably there*[37] *and are* C^2 *with respect to the arc length parametrization.*

[37] With respect to the global paramtrization on $[u_0, u_1]$

With this definition we get

$$t'_f = \left\| \dot{\mathbf{f}}(t^*) \right\|_2^{-1} = \left\| \dot{\mathbf{g}}(t^*) \right\|_2^{-1} = t'_g =: t',$$

and (3.22) yields that

$$\begin{aligned} \mathbf{0} &= f''(t^*) - g''(t^*) = \ddot{\mathbf{f}}(t^*)\ (t')^2 + \dot{\mathbf{f}}(t^*)\ t''_f - \ddot{\mathbf{g}}(t^*)\ (t')^2 + \dot{\mathbf{g}}(t^*)\ t''_g \\ &= (t')^2 \left(\ddot{\mathbf{f}}(t^*) - \ddot{\mathbf{g}}(t^*) + \dot{\mathbf{f}}(t^*)\ \frac{t''_f - t''_g}{(t')^2} \right). \end{aligned}$$

This observation can be formalized in the following statement.

Theorem 3.10 *If* t^* *is no* ***critical point****, a* C^1 *connection is a* G^2 *connection iff there exist* $\lambda \in \mathbb{R}$ *such that* $\ddot{\mathbf{f}}(t^*) - \ddot{\mathbf{g}}(t^*) = \lambda\, \dot{\mathbf{f}}(t^*)$.

3.2 Surfaces

The differential geometry for surfaces is more challenging or interesting[38], due to which we will not consider it in detail but just survey the main concepts without being mathematically precise. A surface is a two dimensional object which only becomes relevant if the **ambient space** $\mathbb{R}^d$ has at least dimension $d = 3$.

Definition 3.11 *A* ***parametric surface*** *is a mapping* $\mathbf{f} : \Omega \to \mathbb{R}^d$, *where* $\Omega \subset \mathbb{R}^2$ *and* $d \geq 3$.

The **parameter domain** Ω can be quite general in an application context. Besides the usual convicts like triangles[39] and rectangles[40], **trimmed surfaces** play an important role in CAD systems. Here one has a **closed curve**

$$\mathbf{f} : [a, b] \to \Omega' \subset \mathbb{R}^2, \qquad \mathbf{f}(a) = \mathbf{f}(b), \tag{3.23}$$

and $\Omega \subset \Omega'$ is the region enclosed by the curve. These curves can be quite general and also complicated but they should not be self intersecting, of course. For simplicity, we also assume that all components of $\mathbf{f}$ are at least C^2, see, for example[41] (Heuser, 1983; Sauer, 2015; Spivak, 1965).

Definition 3.12 (Derivative) *The (total)* ***derivative*** *or* ***Jacobian*** *of* $\mathbf{f}$ *is the matrix valued function*

$$\nabla \mathbf{f} := \left[\frac{\partial f_j}{\partial x_k} : \begin{array}{c} j = 1, \dots, d \\ k = 1, 2 \end{array} \right] = \begin{bmatrix} \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial v} \\ \vdots & \vdots \\ \frac{\partial f_d}{\partial u} & \frac{\partial f_d}{\partial v} \end{bmatrix}, \qquad x = (x_1, x_2) = (u, v). \tag{3.24}$$

[38] Add your favorite euphemism for "difficult" here.

[39] With barycnetric coordinates, of course.

[40] Which connect quite nicely to the theory of manifolds in Analysis, cf. (Sauer, 2015; Spivak, 1965).

[41] The permanent references to my lecture notes are not because they are claimed to be the best reference but because they are the most *accessible* one.

*The surface is said to be **regular** at x if* $\operatorname{rank} \nabla \mathbf{f} = 2$*, that is, if the columns of* $\nabla \mathbf{f}$ *are **linearly independent**.*

A surface **reparametrization** is a C^1 function $\phi : \Omega' \to \Omega$ and it is called **regular** if

$$\det \nabla \phi(x) > 0, \qquad x \in \Omega'. \tag{3.25}$$

Definition 3.13 *The **tangent space** of* $\mathbf{f}$ *at* x *is the two dimensional **affine subspace** generated by the reference system* $[\mathbf{f}(x), \nabla \mathbf{f}(x)]$*, and the **tangent plane** the respective point set in* $\mathbb{R}^d$*:*

$$T_{\mathbf{f}}(x) := [\mathbf{f}(x), \nabla \mathbf{f}(x)]\, \mathbf{A}_2 = \mathbf{f}(x) + \nabla \mathbf{f}(x)\, \mathbb{R}^2. \tag{3.26}$$

If ϕ is a regular reparametrization with $\phi(x') = x$, then, by the chain rule,

$$\begin{aligned} T_{\mathbf{f}\circ\phi}(x') &= \mathbf{f}(\underbrace{\phi(x')}_{=x}) + \nabla\left(\mathbf{f} \circ \phi\right)(x')\, \mathbb{R}^2 = \mathbf{f}(x) + \nabla \mathbf{f}(\phi(x'))\, \underbrace{\nabla \phi(x')\, \mathbb{R}^2}_{=\mathbb{R}^2} \\ &= \mathbf{f}(x) + \nabla \mathbf{f}(x)\, \mathbb{R}^2 = T_{\mathbf{f}}(x), \end{aligned}$$

which means that the tangent plane is independent of regular reparametrizations, hence an **intrinsic** property. If $d = 3$, then the **surface normal** can be computed as

$$\mathbf{n}(x) = \frac{\dfrac{\partial \mathbf{f}}{\partial u} \times \dfrac{\partial \mathbf{f}}{\partial u}}{\left\| \dfrac{\partial \mathbf{f}}{\partial u} \right\| \left\| \dfrac{\partial \mathbf{f}}{\partial u} \right\|}$$

and the tangent plane is

$$T_{\mathbf{f}}(x) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \mathbf{x}^T \mathbf{n}(x) = \mathbf{f}^T(x)\, \mathbf{n}(x) \right\}.$$

Note that the *direction* of the surface normal depends on the parametrization of the surface.

Curvature properties are a little bit more complicated as the **second derivative** of $\mathbf{f}$ is

$$\nabla^2 \mathbf{f} := \left[\frac{\partial^2 f_p}{\partial x_j \partial x_k} : \begin{array}{c} p = 1, \dots, d \\ j, k = 1, 2 \end{array} \right] \in \mathbb{R}^{d \times 2 \times 2},$$

which is a $d \times 2 \times 2$ **tensor**. These can be handled in general, see (Kreyszig, 1959; Struik, 1961), but needs a lot of terminology and theory. To keep things simple and get the idea of curvatures nevertheless, we restrict ourselves to $d = 3$, rotate our coordinate system such that $\mathbf{n}(x) = \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ and assume that locally

$$\mathbf{f}(x) = \begin{bmatrix} u \\ v \\ f(u, v) \end{bmatrix} = \begin{bmatrix} x \\ f(x) \end{bmatrix}$$

is a scalar valued function. Since we assumed that the tangent plane is horizontal, f has a minimum or maximum there and therefore $\nabla f(x) = 0$. The **Hessian**

$$\nabla^2 f(x) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial u^2} & \dfrac{\partial^2 f}{\partial u \partial v} \\ \dfrac{\partial^2 f}{\partial u \partial v} & \dfrac{\partial^2 f}{\partial v^2} \end{bmatrix}$$

of f is a symmetric 2×2 matrix with two real eigenvalues[42] $\lambda_1 \leq \lambda_2$ and (normalized) eigenvectors $\mathbf{x}_1, \mathbf{x}_2$ where

$$\lambda_1 = \mathbf{x}_1^T \nabla^2 f(x) \mathbf{x}_1 \leq \mathbf{y}^T \nabla^2 f(x) \mathbf{y} \leq \mathbf{x}_2^T \nabla^2 f(x) \mathbf{x}_2 = \lambda_2, \qquad \mathbf{y} \in \mathbb{R}^2, \ \|\mathbf{y}\| = 1. \tag{3.27}$$

Consequently, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the two directions in $\mathbb{R}^2$ where the above **bilinear form** is maximally or minimally curved. This is depicted in Fig. 3.3. Since

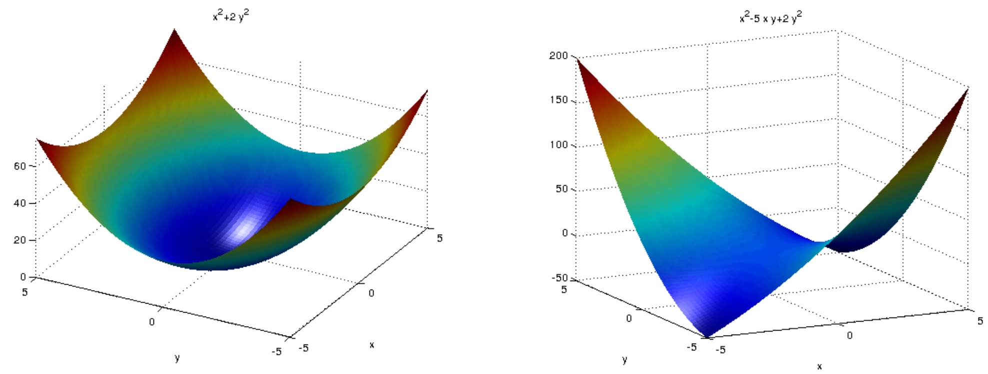

Figure 3.3: Two bilinear forms, one with two positive eigenvalues $0 < \lambda_1 < \lambda_2$ *(left)* and one with a negative and a positive eigenvalue $\lambda_1 < 0 < \lambda_2$ *(right)*. The main lines of curvature are quite visible.

$$\lambda_1 \lambda_2 = \det \nabla^2 f \qquad \text{and} \qquad \lambda_1 + \lambda_2 = \operatorname{trace} \nabla^2 f,$$

these two **invariants**[43] describe the curvature behavior of the quadratic form

$$\mathbf{y} \mapsto \mathbf{y}^T \nabla^2 f(x) \mathbf{y}$$

independently of the coordinate system we use for $\mathbf{y}$.

Definition 3.14 (Curvatures) *The **Gaussian curvature** κ_g and the **mean curvature** κ_m of f at x are defined as*

$$\kappa_g := \lambda_1 \lambda_2 = \det \nabla^2 f(x), \qquad \kappa_m = \frac{1}{2}(\lambda_1 + \lambda_2) = \frac{1}{2} \operatorname{trace} \nabla^2 f(x). \tag{3.28}$$

[42]This is "standard" Linear Algebra, cf. (Brieskorn, 1985; Fischer, 1984).

[43]Eigenvalues are not changed by a **similarity transform** of the form $A \mapsto T^{-1} A T$.

With a proper notion of determinant and trace, this can be carried over to arbitrary parametrized surfaces and defines a local system of principal curvature directions[44] and principal curvatures.

Let us briefly check that the notion is indeed intrinsic and consider

$$\frac{\partial}{\partial x_j'}(f \circ \phi)(x') = \frac{\partial f}{\partial u}(\phi(x'))\,\frac{\partial \phi_1}{\partial x_j'}(x') + \frac{\partial f}{\partial v}(\phi(x'))\,\frac{\partial \phi_2}{\partial x_j'}(x')$$

as well as

$$\begin{aligned}
&\frac{\partial^2}{\partial x_j'\partial x_k'}(f \circ \phi)(x') \\
&= \left(\frac{\partial^2 f}{\partial u^2}(x)\,\frac{\partial \phi_1}{\partial x_k'}(x') + \frac{\partial^2 f}{\partial u \partial v}(x)\,\frac{\partial \phi_2}{\partial x_k'}(x')\right)\frac{\partial \phi_1}{\partial x_j'}(x') + \frac{\partial f}{\partial u}(x)\,\frac{\partial^2 \phi_1}{\partial x_j'\partial x_k'}(x') \\
&\quad + \left(\frac{\partial^2 f}{\partial u \partial v}(x)\,\frac{\partial \phi_1}{\partial x_k'}(x') + \frac{\partial^2 f}{\partial v^2}(x)\,\frac{\partial \phi_2}{\partial x_k'}(x')\right)\frac{\partial \phi_2}{\partial x_j'}(x') + \frac{\partial f}{\partial v}(x)\,\frac{\partial^2 \phi_2}{\partial x_j'\partial x_k'}(x') \\
&= \left(\nabla^T \phi(x')\,\nabla^2 f(x)\,\nabla\phi(x')\right)_{jk} + \underbrace{\frac{\partial f}{\partial u}(x)}_{=0}\left(\nabla^2 \phi_1(x')\right)_{jk} + \underbrace{\frac{\partial f}{\partial v}(x)}_{=0}\left(\nabla^2 \phi_2(x')\right)_{jk} \\
&= \left(\nabla^T \phi(x')\,\nabla^2 f(x)\,\nabla\phi(x')\right)_{jk},
\end{aligned}$$

since we assumed the tangent plane at $f(x')$ to be horizontal. This, however, indicates[45] that the eigenvalues are the same: If $\mathbf{y}$ is a normalized eigenvector to the eigenvalue λ of $\nabla^2 f(x)$, then

$$\begin{aligned}
&\left(\nabla\phi(x')^{-1}\mathbf{y}\right)^T \nabla^2(f\circ\phi)(x')\left(\nabla\phi(x')^{-1}\mathbf{y}\right) \\
&\quad = \mathbf{y}^T\nabla^{-T}\phi(x')\,\nabla^T\phi(x')\,\nabla^2 f(x)\,\nabla\phi(x')\,\nabla\phi(x')^{-1}\mathbf{y} = \mathbf{y}^T\,\nabla^2 f(x)\,\mathbf{y} = \lambda,
\end{aligned}$$

hence the two eigenvalues $\lambda_1' \le \lambda_2'$ of the reparametrized surface satisfy $\lambda_1' \le \lambda_1 \le \lambda_2 \le \lambda_2'$ and since

$$\nabla^f(x) = \nabla^{-T}\phi(x')\,\nabla^2(f\circ\phi)(x')\,\nabla\phi(x'),$$

the same argument also leads to $\lambda_1 \le \lambda_1' \le \lambda_2' \le \lambda_2$ with the final consequence that $\lambda_1 = \lambda_1'$ and $\lambda_2 = \lambda_2'$. Hence, the curvatures are invariant under reparametrization which makes them an intrinsic quantity.

We can also ask the question when two surfaces join smoothly along either a point or a common curve. Continuity is obvious and once the two surfaces join continously along a curve, their restriction to that curve is indeed the curve itself. For simplicity, suppose that

$$\mathbf{f} : [s_0, s_1] \times [t_0, t^*] \to \mathbb{R}^3, \qquad \mathbf{g} : [s_0, s_1] \times [t^*, t_1] \to \mathbb{R}^3.$$

If they join continuously along the curve

$$\mathbf{h} : u \mapsto \mathbf{f}(u, t^*) = \mathbf{g}(u, t^*)$$

[44]The associated eigenvalues.

[45]This is *not* a full proof and not intended to be one.

then the partial derivatives are

$$\frac{\partial \mathbf{f}}{\partial u}(u, t^*) = \frac{\partial \mathbf{g}}{\partial u}(u, t^*) = \dot{\mathbf{h}}(u), \qquad u \in [s_0, s_1],$$

and the nature of the joint depends only on the **cross boundary derivative**. Geometric differentiability is still very simple to describe: the tangent planes have to coincide in each connection point. Higher order differentiability, however, is more intricate and we are not going to consider it here.

Suppose a contradiction were to be found in the axioms of set theory. Do you seriously believe that a bridge would fall down?

F. Ramsey

Geometric objects I: Curves & triangular surfaces 4

In this section we will introduce the basic 1D objects to be manipulated in a CAD system, namely curves.

4.1 Basic curves

We begin with the simples types of curves, namely line segments and circular arcs. They are so simple that this section mainly consists of definitions.

Definition 4.1 *The* ***line segment*** $\ell(\mathbf{x}_0, \mathbf{x}_1)$ *associated to* $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{R}^d$ *is the curve*

$$\ell(\mathbf{x}_0, \mathbf{x}_1)(t) = (1-t)\mathbf{x}_0 + t\mathbf{x}_1, \qquad t \in [0,1].$$

The ***polyline segment*** *or* ***polygon*** *is the curve*

$$\ell(\mathbf{x}_0, \dots, \mathbf{x}_n) = (1-t')\mathbf{x}_{\lfloor t \rfloor} + t'\mathbf{x}_{\lfloor t \rfloor + 1}, \qquad t' := t - \lfloor t \rfloor, \qquad t \in [0,n].$$

Exercise 4.1 Determine the arc length parametrization of a polygon $\ell(\mathbf{x}_0, \dots, \mathbf{x}_n)$. ◊

Definition 4.2 *A* ***circular arc*** $c(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2)$ *is the part of the circle that passes through the three points in the given order. A* ***planar circle*** $c(\mathbf{x}_c, r) \subset \mathbb{R}^2$ *is the well known set*

$$c(\mathbf{x}_c, r) = \{\mathbf{y} \in \mathbb{R}^2 : \|\mathbf{y} - \mathbf{x}_c\| = r\}. \tag{4.1}$$

The above definition of a circle has the drawback that it only works in $\mathbb{R}^2$, in higher dimensions it gives a **sphere** instead. The circular arc, on the other hand, works in arbitrary $\mathbb{R}^d$. If the three points are in general position, they define an affine plane $[\![\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2]\!]$ where the circle and circular arc is well–defined. If they are *not* in general position, then one of the two following cases happens:

1. Two or all three of the points coincide and the problem is simply underdetermined and unsolvable[46].

[46] There is a lot of circles through two points or one point.

2. The three points are **collinear**, i.e., they lie on a straight line. This would correspond to a circle with radius ∞ which would be somewhat reasonable[47] in $\mathbb{R}^2$, but in $\mathbb{R}^3$ there are infinitely many possible midpoints for this "circle".

Remark 4.3 (GIS) *Geographical* ***I****nformation* ***S****ystems (**GIS**) often use only polylines and circular arcs as geometric primitives.*

4.2 Bernstein, Bézier, de Casteljeau

We begin with a classic type of curves, namely the so–called **Bézier curves** which give a representation for polynomial curves. Since **barycentric coordinates** make things quite simple, we generalized things a bit and consider **triangular Bézier surfaces** immediately[48]. But we will always have curves in mind, of course.

Remark 4.4 *Bézier curves are named after* Paul Bézier*, a french engineer who worked for Renault in the CNC department(Bézier, 1972) and, being involved in the development of the UNISURF CAD system in the 1960s, was one of the pioneers of CA(G)D, (Bézier, 1986). The evaluation algorithm was named after the mathematician* Paul Faget de Casteljeau *who developed a similar system at Citroen simultaneously. Since, in contrast to Bézier, de Casteljau was not allowed to publish his results, there was a lifelong conflict about priorities of invention.*

To complete confusion, the basis polynomials that we will consider soon, are called ***Bernstein–Bézier polynomials*** *since* Sergej Bernstein *used them*[49] *in his constructive proof (Bernstein, 1912) of the* ***Weierstrass approximation theorem****. Bernstein polynomials in several variables were already known in the 1950s from (Dinghas, 1951; Lorentz, 1953).*

4.2.1 The de Casteljau algorithm

Now we want to model a **surface** $\mathbf{f} : \mathbb{S}_n \to \mathbb{R}^d$, defined on the n–dimensional **unit simplex** $\mathbb{S}_n$. "Model" means that we want to describe the surface by means of **finite information**, i.e., by finitely many values that are stored in a computer.

Example 4.5 *Let* $\mathbf{a}_0, \dots, \mathbf{a}_n$ *be points in* $\mathbb{R}^d$. *Then*

$$\mathbf{p}(t) = \sum_{j=0}^{n} \mathbf{a}_j t^j, \qquad t \in [a, b]$$

is a ***polynomial curce*** *defined by the* ***control points*** $\mathbf{a}_0, \dots, \mathbf{a}_n \in \mathbb{R}^d$. *That we work in* $\mathbb{R}^d$ *is not really relevant here as we could consider every component of* $\mathbf{p}$ *separately. However, the relationship between the coefficients is not really an intuitive one.*

[47] Though it makes much more sense to use the "line" primitive instead.

[48] This makes sense since triangular surfaces are not common in CAD systems and the "usual" way to generate surfaces is a different one.

[49] And their probabilistic interpretation.

To define a representation with more geometric meaning, we first need some more terminology.

Definition 4.6 *A tuple* $\alpha = (\alpha_0, \ldots, \alpha_d) \in \mathbb{N}_0^{d+1}$ *of nonnegative numbers is called a* ***multiindex****. The* ***lentgh*** $|\alpha|$ *of such s a multiindex is the number*

$$|\alpha| = \sum_{j=0}^{d} \alpha_j.$$

The set of all ***homogeneous multiindices*** *of length* n *wikk be written as*

$$\Gamma_n = \left\{ \alpha \in \mathbb{N}_0^{d+1} : |\alpha| = n \right\}. \tag{4.2}$$

By $\epsilon_j \in \Gamma_1$, $j = 0, \ldots, d$, *we denote the unit multiindices with the property that* $\epsilon_{j,k} = \delta_{jk}$, $j, k = 0, \ldots, d$.

Algorithm 4.7 (de Casteljau)
Given: control points $\mathbf{c}_\alpha$, $\alpha \in \Gamma_n$, *and* $u \in \mathbb{S}_s$.

1. *Initialize* $\mathbf{c}_\alpha^0(u) = \mathbf{c}_\alpha$, $\alpha \in \Gamma_n$.
2. *For* $k = 1, \ldots, n$

$$\mathbf{c}_\alpha^k(u) = \sum_{j=0}^{d} u_j \mathbf{c}_{\alpha+e_j}^{k-1}(u), \qquad \alpha \in \Gamma_{n-k}. \tag{4.3}$$

3. *Result:* $\mathbf{p}(u) = \mathbf{c}_0^n(u)$.

Remark 4.8 *The de Casteljeau algorithm is based on* ***barycentric coordinates****. That means that we can use* any *reference system in* $\mathbb{R}^s$ *but all that really counts is how a point subdivides this reference system.*

Definition 4.9 *The graph* $\mathbf{p}(\mathbb{S}_s)$ *computed by Algorithm 4.7 is called the* ***Bézier surface*** *associated to the* ***control polyhedron*** $\{\mathbf{c}_\alpha \; : \; \alpha \in \Gamma_n\}$.

Remark 4.10

1. *Each step of the de Casteljau algorithm subdivides any of the* s*–dimensional simplices* $\left[\mathbf{c}_{\alpha+e_j}^{k-1} : j = 0, \ldots, d\right]$, $\alpha \in \Gamma_{n-k}$ *in exactly the same way as* u *sudivides the unit simplex. Therefore, the algorithm can easily be carried over to a method to define functions on arbitrary* d*–simplices* Δ *by simply transferring the subdivision induced by* $x \in \Delta$ *to the respective subsimplices of the control polyhedron.*
2. *Some of the simplices* $\left[\mathbf{c}_{\alpha+e_j}^{k-1} : j = 0, \ldots, d\right]$ *may even be degenerated, for the determination of the "barycenter" via* u *this is totally irrelevant.*
3. *Each step of the algorithm can be seen as a* ***linear interpolation*** *of the vertices of the subsimplices.*
4. *The algorithm is a generalization of construction methods for conic sections that date back to Steiner*[50].

[50] Which is not too surprising as de Casteljau's background is in classical geometry.

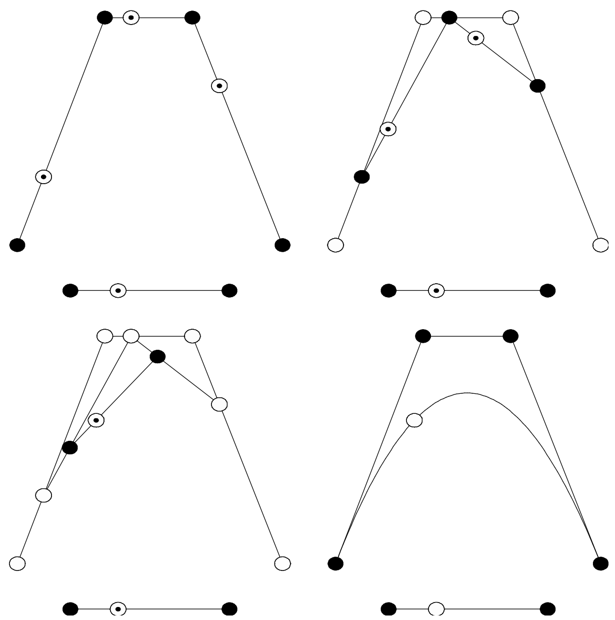

Figure 4.1: The de Casteljau algorithm for a curve. For each point in the curve it is a **ruler and compass** construction.

4.2.2 Bézier surfaces

Next, we will find out that the de Casteljeau algorithm produces polynomial surfaces of (total) degree n that can be given explicitly. These surfaces are called **Bézier surface** though Bézier himself only used that approach to define curves[51], see (Bézier, 1972).

Definition 4.11

1. *For a multiindex $\alpha \in \mathbb{N}_0^{s+1}$ the **multinomial coefficient***

$$\binom{|\alpha|}{\alpha} = \frac{|\alpha|!}{\alpha_0! \cdots \alpha_d!},$$

   is extended on $\mathbb{Z}^{s+1}$ by setting it equal to zero whenever it contains a negative component.

2. *The **Bernstein–Bézier basis polynomial**[52] with index $\alpha \in \Gamma_n$ is defined as*

$$B_\alpha(x) = \binom{|\alpha|}{\alpha} u^\alpha = \frac{|\alpha|!}{\alpha_0! \cdots \alpha_d!} u_0^{\alpha_0} \cdots u_d^{\alpha_d}, \qquad \alpha \in \mathbb{N}_0^{s+1}. \tag{4.4}$$

3. *Convention: B_α is even defined for $\alpha \in \mathbb{Z}^{s+1}$ with $B_\alpha \equiv 0$ whenever $\alpha \notin \mathbb{N}_0^{d+1}$.*

[51] His notion of a surface was that of a curve moving along another curve, a concept that leads to a **tensor product surface**, see (Farin, 1988, Kap. 1).

[52] Carl de Boor once proposed the abbreviation "B–polynomial".

Exercise 4.2 Show that B_α assumes its maximum at $u = \alpha/|\alpha|$. ◊

Lemma 4.12 (Recurrence relation) *For $\alpha \in \mathbb{N}_0^{s+1}$, $\alpha \neq 0$, we have*

$$B_\alpha(u) = \sum_{j=0}^{d} u_j \, B_{\alpha-\epsilon_j}(u). \tag{4.5}$$

Proof: Follows from

$$\begin{aligned}\binom{|\alpha|}{\alpha} &= \underbrace{\sum_{\substack{j=0\\ \alpha_j>0}}^{s} \frac{\alpha_j}{|\alpha|}}_{=1} \binom{|\alpha|}{\alpha} = \sum_{\substack{j=0\\ \alpha_j>0}}^{s} \frac{(|\alpha|-1)!}{\alpha_0!\cdots\alpha_{j-1}!\,(\alpha_j-1)!\alpha_{j+1}\cdots\alpha_d!}\\ &= \sum_{j=0}^{s} \binom{|\alpha|-1}{\alpha-\epsilon_j}\end{aligned}$$

since then

$$B_\alpha(u) = \binom{|\alpha|}{\alpha} u^\alpha = \sum_{j=0}^{s} \binom{|\alpha|-1}{\alpha-\epsilon_j} u_j u^{\alpha-\epsilon_j} = \sum_{j=0}^{s} u_j \, B_{\alpha-\epsilon_j}(u).$$

Note that we make use of the above convention here. □

Lemma 4.13 *The Bernstein–Bézier basis polynomials form a* ***nonnegative partition of unity****, that is*

$$B_\alpha \geq 0 \qquad \textit{and} \qquad \sum_{\alpha\in\Gamma_n} B_\alpha \equiv 1. \tag{4.6}$$

Proof: Nonnegativity is obvious and the partition of unity follows from the **multinomial formula**

$$1 = 1^n = (u_0 + \cdots + u_s)^n = \sum_{\alpha\in\Gamma_n} \binom{n}{\alpha} u^\alpha = \sum_{\alpha\in\Gamma_n} B_\alpha(u).$$

□

Theorem 4.14 (Representation of Bézier surfaces) *For $k = 0, \dots, n$ and $\alpha \in \Gamma_{n-k}$ we have*

$$\mathbf{c}_\alpha^k(u) = \sum_{\beta\in\Gamma_k} \mathbf{c}_{\alpha+\beta} B_\beta(u), \tag{4.7}$$

and especially for $k = n$

$$\mathbf{p}(x) = \mathbf{c}_0^n(u) = \sum_{\alpha\in\Gamma_n} \mathbf{c}_\alpha \, B_\alpha(u) \tag{4.8}$$

Definition 4.15 *We write*

$$\mathbb{R}^d(\Gamma_n) = \left\{(\mathbf{c}_\alpha : \alpha \in \Gamma_n) : \mathbf{c}_\alpha \in \mathbb{R}^d,\ \alpha \in \Gamma_n\right\}$$

and write, for a control polyhedron $\mathbf{c} \in \mathbb{R}^d(\Gamma_n)$*,*

$$B_n\mathbf{c} := \sum_{\alpha\in\Gamma_n} \mathbf{c}_\alpha B_\alpha(x)$$

to describe the associated Bézier surface.

Proof of Theorem 4.14: Induction on k where $k = 0$ is clear because $B_0 \equiv 1$. For the step $k \to k+1$ we observe that for $\alpha \in \Gamma_{n-k-1}$ the identity (4.3) yields

$$\begin{aligned}
\mathbf{c}_\alpha^{k+1}(u) &= \sum_{j=0}^{d} u_j \mathbf{c}_{\alpha+\epsilon_j}^{k}(u) = \sum_{j=0}^{d} u_j \sum_{\beta\in\Gamma_k} \mathbf{c}_{\alpha+\epsilon_j+\beta}\, B_\beta(u) \\
&= \sum_{j=0}^{d} \sum_{\beta\in\Gamma_{k+1}} u_j \mathbf{c}_{\alpha+\beta} B_{\beta-\epsilon_j}(u) = \sum_{\beta\in\Gamma_{k+1}} \mathbf{c}_{\alpha+\beta} \sum_{j=0}^{d} u_j\, B_{\beta-\epsilon_j}(u) \\
&= \sum_{\beta\in\Gamma_{k+1}} \mathbf{c}_{\alpha+\beta} B_\beta(u).
\end{aligned}$$

□

Next, we collect some properties of the Bézier surface $B_n\mathbf{c}$ that follow more or less directly from the de Casteljeau algorithm:

Convex hull property: the surface lies inside the **convex hull** of the control polyhedron,

$$B_n\mathbf{c}(\Delta) \subset [\![\mathbf{c}_\alpha : \alpha \in \Gamma_n]\!]. \tag{4.9}$$

Endpoint interpolation: the surface interpolates the control polygon at the vertices of the simplex

$$B_n\mathbf{c}(e_j) = \mathbf{c}_{n\epsilon_j}, \qquad j = 0, \dots, s. \tag{4.10}$$

Indeed, (4.9) follows since, by (4.3), we have, for $u \in \mathbb{S}_s$, $k = 1, \dots, n$ and $\alpha \in \Gamma_{n-k}$

$$\mathbf{c}_\alpha^k(x) \in [\![\mathbf{c}_\beta^{k-1}(x) : \beta \in \Gamma_{n-k+1}]\!],$$

hence,

$$B_n\mathbf{c}(u) \in [\![\mathbf{c}_\alpha^{n-1}(u) : \alpha \in \Gamma_1]\!] \subseteq \cdots \subseteq [\![\mathbf{c}_\alpha : \alpha \in \Gamma_n]\!].$$

(4.10) is due to

$$B_n\mathbf{c}(e_j) = \mathbf{c}_0^n(e_j) = \mathbf{c}_{\epsilon_j}^{n-1}(e_j) = \cdots = \mathbf{c}_{ne_j}.$$

Lemma 4.16 *The Bernstein–Bézier basis polynomials form a basis of the vector space* Π_n^0 *of all homogeneous polynomials of degree* n *in* u*.*

Proof: Since barycentric coordinates are homogeneous polynomials of **total degree** 1, it is clear that $B_\alpha \in \Pi_n^0$, $\alpha \in \Gamma_n$.Since als $\#\Gamma_n = \binom{n+d}{d} = \dim \Pi_n^0$, it suffices to prove the **linear independence** of the basis polynomials which we will do by induction on $n \in \mathbb{N}_0$. The case $n = 0$ is obvious since $B_0 = 1$ is not the zero function. For the induction step we assume that there exists $\mathbf{c} \in \mathbb{R}(\Gamma_{n+1})$ such that

$$0 = B_{n+1}\mathbf{c}(u) = \mathbf{c}_0^{n+1}(u), \qquad u \in \mathbb{S}_s,$$

and show that $\mathbf{c} = 0$. To that end, we consider the lower dimensional faces

$$\Delta_k = \{u : u_{k+1} = \cdots = u_d = 0\} \qquad k = 0, \ldots, s,$$

where $\Delta_s = \mathbb{S}_s$ and prove inductively[53] for $k = 0, \ldots, d$, that

$$\mathbf{c}_\alpha = 0, \qquad \alpha \in \{\beta \in \Gamma_{n+1} : \beta_{k+1} = \cdots = \beta_d = 0\}. \tag{4.11}$$

$k = 0$ is the **endpoint interpolation** (4.10). To advance from $k-1$ to k, we choose $u \in \Delta_k$ and obtain

$$B_{n+1}\mathbf{c}(u) = \sum_{\alpha_{k+1}=\cdots=\alpha_d=0} \mathbf{c}_\alpha B_\alpha(u). \tag{4.12}$$

Since $\Delta_{k-1} \subset \Delta_k$ the induction hypothesis yields that $\mathbf{c}_\alpha = 0$, $\alpha_k = \cdots = \alpha_d = 0$, that is, (4.12) becomes

$$\begin{aligned}
B_{n+1}\mathbf{c} &= \sum_{\substack{\alpha\in\Gamma_m,\,\alpha_k>0\\ \alpha_{k+1}=\cdots=\alpha_d=0}} \mathbf{c}_\alpha B_\alpha \\
&= (n+1)u_k \sum_{\substack{\alpha\in\Gamma_m,\,\alpha_k>0\\ \alpha_{k+1}=\cdots=\alpha_d=0}} \frac{\mathbf{c}_\alpha}{\alpha_k} B_{\alpha-e_k} \\
&= \sum_{\substack{\alpha\in\Gamma_n\\ \alpha_{k+1}=\cdots=\alpha_d=0}} \frac{\mathbf{c}_{\alpha+e_k}}{\alpha_k+1} B_\alpha.
\end{aligned}$$

To this expression we finally apply induction on n to complete the proof. □

Exercise 4.3 Prove the **degree elevation** formula. To that end, define for $\mathbf{c} \in \mathbb{R}^d(\Gamma_n)$ a new control polyhedron $\widehat{\mathbf{c}} \in \mathbb{R}^d(\Gamma_{n+1})$ by

$$\widehat{\mathbf{c}}_\beta = \sum_{j=0}^{d} \frac{\beta_j}{n+1} \mathbf{c}_{\beta-\epsilon_j}, \qquad \beta \in \Gamma_{n+1},$$

and show that $B_n\mathbf{c} = B_{n+1}\widehat{\mathbf{c}}$. ◊

4.2.3 Derivatives of Bézier surfaces

Now we consider derivatives of Bézier surfaces and their geometric meaning. We recall some concepts from the previous barycentric chapter.

[53] Yeah, a *double induction*!

Definition 4.17 *A vector* $y \in \mathbb{R}^{d+1}$ *is called* ***direction*** *if* $y_0 + \cdots + y_d = 0$. *The* ***directional derivative*** D_y *along* y, *is defined for* $f \in C^1(\mathbb{S}_s)$ *as*

$$D_y f = \sum_{j=0}^{d} y_j \frac{\partial f}{\partial u_j}. \tag{4.13}$$

The side condition $\sum y_j = 0$ reflects the fact that $y \in \mathbb{E}_s'$. Particular directional derivatives are the **axial deriviatives** $D_{e_j - e_k}$, $j, k = 0, \dots, s$, which for a **basis** for all directional derivatives since any $y \in \mathbb{E}_s'$ can be written as

$$y = \sum_{j=0}^{s} y_j e_j = \sum_{j \neq k} y_j (e_j - e_k) + \underbrace{\left(y_k + \sum_{j \neq k} y_j \right)}_{=0} e_k.$$

The directional derivatives of the basis polynomials can now be easily computed.

Lemma 4.18 *For a barycentric direction* $y \in \mathbb{R}^{s+1}$ *and* $\alpha \in \mathbb{N}_0^s$ *we have that*

$$D_y B_\alpha = |\alpha| \sum_{j=0}^{d} y_j \, B_{\alpha - e_j}. \tag{4.14}$$

In particular, for $j, k = 0, \dots, s$,

$$D_{e_j - e_k} B_\alpha = |\alpha| \left(B_{\alpha - e_j} - B_{\alpha - e_k} \right). \tag{4.15}$$

Proof: Since $B_\alpha = \binom{|\alpha|}{\alpha} u^\alpha$, we get, whenever $\alpha_j > 0$,

$$\frac{\partial}{\partial u_j} B_\alpha = \frac{|\alpha|!}{\alpha_0! \cdots \alpha_d!} \alpha_j u^{\alpha - e_j} = |\alpha| \binom{|\alpha - e_j|}{\alpha - e_j} u^{\alpha - e_j} = |\alpha| B_{\alpha - e_j},$$

and if $\alpha_j = 0$ it follows that $\frac{\partial}{\partial u_j} B_\alpha = 0$. $\square$

Definition 4.19 *For* $j = 0, \dots, s$, *the* ***shift operators*** $E_j : \mathbb{R}^N(\Gamma_n) \to \mathbb{R}^N(\Gamma_{n-1})$ *are defined as*

$$(E_j \mathbf{c})_\alpha = \mathbf{c}_{\alpha + e_j}, \qquad \alpha \in \Gamma_{n-1}.$$

With Lemma 4.18 and the notion of shift operators we can prove the following result.

Theorem 4.20 *If* $y_1, \dots, y_m$ *are* ***axial directions***, *i.e.,*

$$y_j = e_{\ell_j} - e_{k_j}, \qquad j = 1, \dots, m,$$

then

$$D_{y_1} \cdots D_{y_m} B_n \mathbf{c} = \frac{n!}{(n-m)!} \sum_{\alpha \in \Gamma_{n-m}} (E_{\ell_1} - E_{k_1}) \cdots (E_{\ell_m} - E_{k_m}) \, \mathbf{c}_\alpha \, B_\alpha \tag{4.16}$$

Proof: It is sufficient to consider the case $m = 1$, the rest is a simple induction[54]. By Lemma 4.18 we get

$$\begin{aligned} D_{e_j - e_k} B_n \mathbf{c} &= \sum_{\alpha \in \Gamma_n} \mathbf{c}_\alpha D_{e_j - e_k} B_\alpha = \sum_{\alpha \in \Gamma_n} \mathbf{c}_\alpha |\alpha| \left(B_{\alpha - e_j} - B_{\alpha - e_k} \right) \\ &= n \sum_{\alpha \in \Gamma_{n-1}} \left(\mathbf{c}_{\alpha + e_j} - \mathbf{c}_{\alpha + e_k} \right) B_\alpha = n \sum_{\alpha \in \Gamma_{n-1}} (E_j - E_k)\, \mathbf{c}_\alpha\, B_\alpha. \end{aligned}$$

□

Theorem 4.21 *Let $y \in \mathbb{R}^{s+1}$ be a barycentric direction. Then, for $k \geq 1$,*

$$D_y^k B_n \mathbf{c} = \frac{n!}{(n-k)!} \sum_{\alpha \in \Gamma_{n-k}} \sum_{\beta \in \Gamma_k} \mathbf{c}_{\alpha+\beta}\, B_\alpha(\cdot)\, B_\beta(y) \tag{4.17}$$

Remark 4.22 *The notation $B_\beta(y)$ is slightly abusive and only stands for*

$$B_\beta(y) = \binom{|\beta|}{\beta} y^\beta.$$

Proof of Theorem 4.21: Induction on k, where for $k = 1$ (4.14) yields

$$\begin{aligned} D_y B_n \mathbf{c} &= n \sum_{\alpha \in \Gamma_n} \mathbf{c}_\alpha \sum_{j=0}^{d} y_j B_{\alpha - e_j} = n \sum_{\alpha \in \Gamma_{n-1}} \sum_{j=0}^{d} y_j \mathbf{c}_{\alpha + e_j}\, B_\alpha \\ &= n \sum_{\alpha \in \Gamma_{n-1}} \sum_{\beta \in \Gamma_1} \mathbf{c}_{\alpha+\beta}\, B_\alpha(\cdot) B_\beta(y). \end{aligned}$$

For $k > 1$ the induction hypothesis and (4.5) give

$$\begin{aligned} D_y^k B_n \mathbf{c} &= D_y\, D_y^{k-1} B_n \mathbf{c} = D_y \sum_{\alpha \in \Gamma_{n-k+1}} \sum_{\beta \in \Gamma_{k-1}} \mathbf{c}_{\alpha+\beta}\, B_\alpha(\cdot) B_\beta(y) \\ &= \sum_{\alpha \in \Gamma_{n-k+1}} \sum_{\beta \in \Gamma_{k-1}} \mathbf{c}_{\alpha+\beta}\, B_\beta(y) \sum_{j=0}^{n} y_j B_{\alpha - e_j}(\cdot) \\ &= \sum_{\alpha \in \Gamma_{n-k}} \sum_{\beta \in \Gamma_{k-1}} \sum_{j=0}^{n} y_j \mathbf{c}_{\alpha+\beta+e_j}\, B_\beta(y) B_\alpha(\cdot) \\ &= \sum_{\alpha \in \Gamma_{n-k}} \sum_{\beta \in \Gamma_k} \mathbf{c}_{\alpha+\beta}\, B_\alpha(\cdot) \sum_{j=0}^{n} y_j B_{\beta - e_j}(y) = \sum_{\alpha \in \Gamma_{n-k}} \sum_{\beta \in \Gamma_k} \mathbf{c}_{\alpha+\beta}\, B_\alpha(\cdot) B_\beta(y). \end{aligned}$$

□

Corollary 4.23 *For any barycentric direction $y \in \mathbb{E}_s'$ and any $k \geq 1$ we have*

$$D_y^k B_n \mathbf{c}(u) = \frac{n!}{(n-k)!} \sum_{\beta \in \Gamma_k} \mathbf{c}_\beta^{n-k}(u)\, B_\beta(y), \qquad u \in \mathbb{S}_s. \tag{4.18}$$

In particular,

---

[54] And a nice exercise.

1. *the intermediate results of the* ***de Casteljeau algorithm*** *give the respective derivatives for free:*

$$D_y B_n \mathbf{c}(u) = n \sum_{j=0}^{d} y_j \mathbf{c}_{\epsilon_j}^{n-1}(u), \qquad u \in \mathbb{S}_s. \tag{4.19}$$

2. *The derivatives at the vertices* e_j *of* $\mathbb{S}_s$, $j = 0, \dots, s$, *are completeley determined by the control points around the edges:*

$$D_y^k B_n \mathbf{c}\,(e_j) = \frac{n!}{(n-k)!} \sum_{\beta \in \Gamma_k} \mathbf{c}_{(n-k)\epsilon_j + \beta}\, B_\beta(y), \qquad j = 0, \dots, s. \tag{4.20}$$

3. *The* ***tangent plane*** *at a vertex* e_j, $j = 0, \dots, s$, *is generated by*

$$(E_k - E_j)\, \mathbf{c}_{(n-1)\epsilon_j} = \mathbf{c}_{(n-1)\epsilon_j + \epsilon_k} - \mathbf{c}_{n\epsilon_j}, \qquad k = 0, \dots, s,\ k \neq j.$$

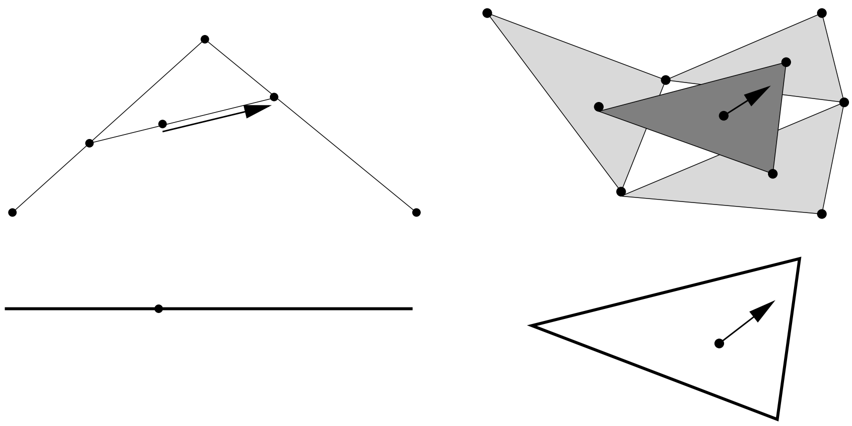

Figure 4.2: How to determine a directional derivative from the intermediate results of the de Casteljau algorithm.

Remark 4.24 (Control points and derivatives)

1. *The directional derivatives in Theorem 4.20 play the role of a* ***partial derivative*** *and are defined by the respective differences of control points.*

2. *In his book (Bézier, 1972), Bézier introduces Bézier curves precisely in this way: value and derivatives at the end points of an interval should be defined by the differences of control points.*

Exercise 4.4 Prove that there exists exactly one quadratic (i.e., $n = 2$) polynomial that assumes prescribed function values and tangent planes at the vertices of a simplex. ◊

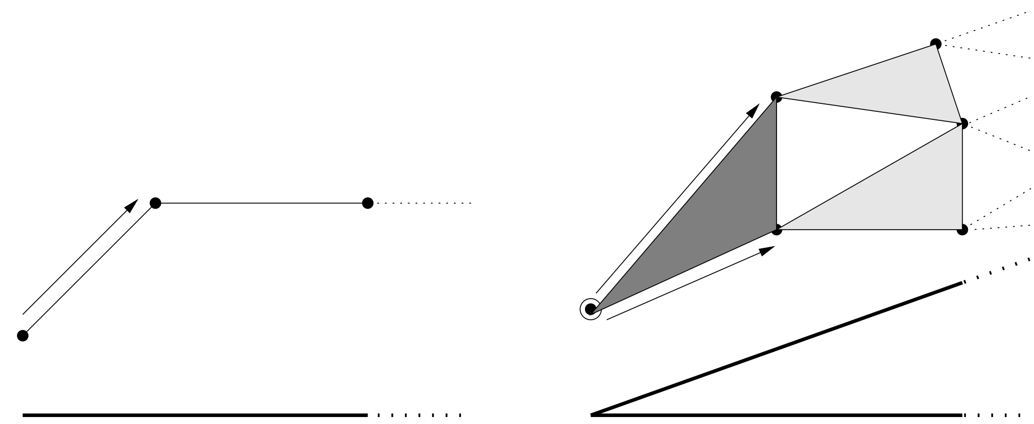

Figure 4.3: Derivatives and control points in one and two variables. The tangent plane in the vertex is given by the difference of the control points.

4.2.4 Blossoming and Subdivision

Any evaluation point u splits the standard simplex $\mathbb{S}_s$ into $s+1$ subsimplices as we know from the computation of barycentric coordinates. The restriction of a Bézier surface to such a subsimplex is again a Bézier surface whose coefficients have to be computed. For this purpose, there exists a very nice and elegant theory that emerges from reconsidering and slightly generalizing the **de Casteljeau algorithm** from Algorithm 4.7.

Algorithm 4.25 (de Castejau modified)
Given: $\mathbf{c} \in \mathbb{R}^d(\Gamma_n)$ *und* $u_1, \dots, u_n \in \mathbb{S}_s$.

1. *Initalize* $\mathbf{c}_\alpha^0() = \mathbf{c}_\alpha$, $\alpha \in \Gamma_n$.
2. *For* $k = 1, \dots, n$

$$\mathbf{c}_\alpha^k(u_1, \dots, u_k) = \sum_{j=0}^{s} u_{jk}\, \mathbf{c}_{\alpha+\epsilon_j}^{k-1}(u_1, \dots, u_{k-1}), \qquad \alpha \in \Gamma_{n-k}. \tag{4.21}$$

Result: $\mathbf{P_c}(u_1, \dots, u_n) = \mathbf{c}_0^n(u_1, \dots, u_n)$.

The difference to Algorithm 4.7 is that in each step of the iteration we can use *different* barycentric coordinates. Though it looks like a very naive generalization, this process yields a surprisingly meaningful result.

Proposition 4.26 *The function* $\mathbf{P_c}$ *is a* ***symmetric multiaffine form*** *with* ***diagonal*** $B_n\mathbf{c}$*, that is,*

1. *(symmetric) for any* ***permutation*** σ *of* $\{1, \dots, n\}$*, we have*

$$\mathbf{P_c}\left(u_{\sigma(1)}, \dots, u_{\sigma(n)}\right) = \mathbf{P_c}(u_1, \dots, u_n) \tag{4.22}$$

2. *(multiaffine) if* $u_j = \sum_{k=0}^m u'_k v_k$, $u'_k \in \mathbb{S}_s$, $v \in \mathbb{S}_m$, *then*

$$\mathbf{P_c}\left(u_1, \dots, u_n\right) = \sum_{k=0}^m v_k \mathbf{P_c}\left(u_1, \dots, u_{j-1}, u'_k, u_{j+1} \dots, u_n\right). \qquad (4.23)$$

3. *(diagonal)*

$$\mathbf{P_c}\left(u, \dots, u\right) = B_n \mathbf{c}(u), \qquad u \in \mathbb{S}_s. \qquad (4.24)$$

Proof: Since the permutations of $\{1, \dots, d\}$ are generated by the permutations that switch two **subsequent elements**, it suffices to show that

$$\mathbf{P_c}\left(u_1, \dots, u_n\right) = \mathbf{P_c}\left(u_1, \dots, u_{j-1}, u_{j+1}, u_j, u_{j+2}, \dots, u_n\right) \qquad (4.25)$$

By (4.21) this reduces to showing that

$$\mathbf{c}_\alpha^{j+1}\left(u_1, \dots, u_{j-1}, u_j, u_{j+1}\right) = \mathbf{c}_\alpha^{j+1}\left(u_1, \dots, u_{j-1}, u_{j+1}, u_j\right), \qquad \alpha \in \Gamma_{n-j-1}. \quad (4.26)$$

To that end, we simply apply (4.21) twice

$$\begin{aligned}
&\mathbf{c}_\alpha^{j+1}\left(u_1, \dots, u_{j-1}, u_j, u_{j+1}\right) \\
&= \sum_{k=0}^s u_{j+1,k} \mathbf{c}_{\alpha+\epsilon_k}^j \left(u_1, \dots, u_{j-1}, u_j\right) = \sum_{k=0}^s u_k \sum_{\ell=0}^s u_{j,\ell}\, \mathbf{c}_{\alpha+\epsilon_k+\epsilon_\ell}^{j-1}\left(u_1, \dots, u_{j-1}\right) \\
&= \sum_{k,\ell=0}^s u_{j+1,k} u_{j,\ell}\, \mathbf{c}_{\alpha+\epsilon_k+\epsilon_\ell}^{j-1}\left(u_1, \dots, u_{j-1}\right) = \mathbf{c}_\alpha^{j+1}\left(u_1, \dots, u_{j-1}, u_{j+1}, u_j\right),
\end{aligned}$$

since the expression within the sum is symmetric with respect to k and ℓ which yields (4.26) and (4.25).

(4.23) is verified in the same way by computing

$$\begin{aligned}
&\mathbf{c}_\alpha^j\left(u_1, \dots, u_j\right) = \mathbf{c}_\alpha^j\left(u_1, \dots, \sum_{k=0}^m u'_k v_k\right) \\
&= \sum_{\ell=0}^s \left(\sum_{k=0}^m u'_k v_k\right)_\ell \mathbf{c}_{\alpha+\epsilon_\ell}^{j-1}\left(u_1, \dots, u_{j-1}\right) = \sum_{k=0}^m v_k \underbrace{\sum_{\ell=0}^s u'_{k,\ell}\, \mathbf{c}_{\alpha+\epsilon_\ell}^{j-1}\left(u_1, \dots, u_{j-1}\right)}_{=\mathbf{c}_\alpha^j\left(u_1,\dots,u_{j-1},u'_k\right)} \\
&= \sum_{k=0}^m v_k\, \mathbf{c}_\alpha^j\left(u_1, \dots, u_{j-1}, u'_k\right).
\end{aligned}$$

Finally, (4.24) follows from the fact that in this case Algorithm 4.25 reduces to the de Casteljeau algorithm 4.7. $\square$

Remark 4.27 *Multiaffine forms are not such a totally new and unknown concept, as we know quite promiment examples of a* ***multilinear form****: the* ***derivative*** *of a function in several variable is a symmetric one, the* ***determinant*** *an alternating one.*

Theorem 4.28 (Blossoming Principle) *For any* d*–vector valued polynomial* $\mathbf{p}$ *with components in* Π_n^0 *there is a unique symmetric multiaffine form* $\mathbf{P} : (\mathbb{E}_s)^n \to \mathbb{R}^d$ *and vice versa such that*

$$\mathbf{p}(u) = \mathbf{P}(u, \dots, u), \qquad x \in \mathbb{S}_s. \tag{4.27}$$

Definition 4.29 *The multiaffine form* $\mathbf{P}$ *from Theorem 4.28 is called the* ***polar form*** *or* ***blossom*** *of* $\mathbf{p}$*. This concept was rediscovered by Ramshaw (Ramshaw, 1987) in 1987 and brought to attention in the CAGD community*

Proof of Theorem 4.28: We write $\mathbf{p}$ in its **Bézier representation** as

$$\mathbf{p} = B_n \mathbf{c}(\mathbf{p}) = \sum_{\alpha \in \Gamma_n} \mathbf{c}_\alpha(\mathbf{p}) \; B_\alpha, \qquad \mathbf{c}_\alpha(\mathbf{p}) \in \mathbb{R}^d, \; \alpha \in \Gamma_n,$$

and recall from Proposition 4.26 that the associated $\mathbf{P}_{\mathbf{c}(\mathbf{p})}$ is a symmetric multiaffine form that satisfies (4.27).

Conversely, let $e_j \in \mathbb{R}^{s+1}$, $j = 0, \dots, s$, denote the barycentric coordinates of the vertices of the simplex, then we have, for any n–affine form $\mathbf{P}$ that

$$\mathbf{P}(u, \dots, u) = \sum_{\alpha \in \Gamma_n} \mathbf{P}\Big(\underbrace{e_0, \dots, e_0}_{\alpha_0}, \dots, \underbrace{e_d, \dots, e_d}_{\alpha_d}\Big) B_\alpha(u) =: \mathbf{P}(e^\alpha) \, B_\alpha(u) \tag{4.28}$$

which shows that $\mathbf{P}(u, \dots, u)$ is of the desired form. But of course, we have to prove (4.28) which we will do by induction on n. In the trivial case $n = 0$ the form without arguments $\mathbf{P}()$ is constant, just like polynomials of degree 0. For $n \to n+1$ let $\mathbf{P}$ be a symmetric $(n+1)$–affine form. Since, trivially,

$$u = \sum_{j=0}^{s} u_j \, e_j,$$

we have that

$$\mathbf{P}(\underbrace{u, \dots, u}_{n+1}) = \sum_{j=0}^{s} u_j \, \mathbf{P}(e_j, \underbrace{u, \dots, u}_{n}) =: \sum_{j=0}^{s} u_j \, \mathbf{P}_j(\underbrace{u, \dots, u}_{n}) \tag{4.29}$$

and the functions $\mathbf{P}_j = (\nu_j, \cdot)$ are symmetric n–affine forms to which we can apply the induction hypothesis yielding

$$\mathbf{P}_j(u, \dots, u) = \sum_{\alpha \in \Gamma_n} \mathbf{P}_j(e^\alpha), \qquad j = 0, \dots, d,$$

to obtain

$$\begin{aligned}
\mathbf{P}\Big(\underbrace{u, \dots, u}_{n+1}\Big) &= \sum_{j=0}^{s} u_j \sum_{\alpha \in \Gamma_n} \mathbf{P}_j(e^\alpha) \, B_\alpha(u) = \sum_{j=0}^{d} u_j \sum_{\alpha \in \Gamma_n} \mathbf{P}(e_j, e^\alpha) \, B_\alpha(u) \\
&= \sum_{j=0}^{d} u_j \sum_{\alpha \in \Gamma_n} \mathbf{P}(e^{\alpha + \epsilon_j}) \, B_\alpha(u) = \sum_{j=0}^{d} u_j \sum_{\alpha \in \Gamma_{n+1}} \mathbf{P}(e^\alpha) \, B_{\alpha - \epsilon_j}(u) \\
&= \sum_{\alpha \in \Gamma_{n+1}} \mathbf{P}(e^\alpha) \underbrace{\sum_{j=0}^{d} u_j B_{\alpha - \epsilon_j}(u)}_{= B_\alpha(u)} = \sum_{\alpha \in \Gamma_{n+1}} \mathbf{P}(e^\alpha) \, B_\alpha(u),
\end{aligned}$$

which proves (4.28). Uniqueness of this relationship follows from (4.28) and the uniqueness of the Bézier representation. □

The result we are aiming for is an almost direct consequence of the following result.

Proposition 4.30 *For* $\mathbf{c} \in \mathbb{R}^d(\Gamma_n)$ *let* $\mathbf{P}$ *be the* ***polar form*** *of* $\mathbf{p} = B_n\mathbf{c}$. *Then the intermediate points of the* ***de Casteljeau algorithm*** *are of the form*

$$\mathbf{c}_\beta^k(u) = \mathbf{P}\left(u^k, e^\beta\right) := \mathbf{P}(\underbrace{u, \dots, u}_{k}, e^\beta), \qquad \beta \in \Gamma_{n-k}. \tag{4.30}$$

Proof: Induction on k once more[55]. For $k = 0$ we have

$$\mathbf{c}_\beta^0(u) = \mathbf{c}_\beta = \mathbf{P}\left(e^\beta\right), \qquad \beta \in \Gamma_n,$$

which is (4.30). For $k-1 \to k$ the de Casteljeau algorithm implies

$$\begin{aligned} \mathbf{c}_\beta^k(u) &= \sum_{j=0}^{d} u_j\, \mathbf{c}_{\beta+\epsilon_j}^{k-1}(u) = \sum_{j=0}^{s} u_j\, \mathbf{P}\left(u^{k-1}, e^{\beta+\epsilon_j}\right) = \sum_{j=0}^{s} u_j\, \mathbf{P}\left(e_j, u^{k-1}, e^\beta\right) \\ &= \mathbf{P}\left(\sum_{j=0}^{s} u_j\, e_j, u^{k-1}, e^\beta\right) = \mathbf{P}\left(u, u^{k-1}, e^\beta\right) = \mathbf{P}\left(u^k, e^\beta\right) \end{aligned}$$

and advances the induction hypothesis. □

Definition 4.31 (Subsimplex)

1. *For* $u \in \mathbb{S}_s$ *and* $j = 0, \dots, s$, *we denote by*

$$\Delta_j = \Delta_j(u) := [\![u, e_0, \dots, e_{j-1}, e_{j+1}, \dots, e_d]\!] \tag{4.31}$$

   the j*th* ***subsimplex*** *of* $\mathbb{S}_s$.

2. *The* ***restriction*** *of* $B_n\mathbf{c}$ *on* Δ_j, *written as* $B_n\mathbf{c}|_{\Delta_j} : \mathbb{S}_s \to \mathbb{R}^d$ *is the Bézier surface based on the barycentric coordinates with respect to* Δ_j.

Exercise 4.5 Prove that

$$\mathbb{S}_s = \bigcup_{j=0}^{s} \Delta_j.$$

◊

We can now easily find the control polygon for the restriction among the intermediate results of the **de Casteljeau algorithm**.

Theorem 4.32 *For* $u \in \Delta$ *and* $j = 0, \dots, d$ *we have*

$$B_n\mathbf{c}|_{\Delta_j(u)} = \sum_{\alpha \in \Gamma_n} \mathbf{c}_{\alpha - \alpha_j \epsilon e_j}^{\alpha_j}(u)\, B_\alpha. \tag{4.32}$$

[55] It seems as if induction is the only this people in CAGD can handle. The truth, however, is much simpler: when things are given recursively, induction is usually the technique of choice.

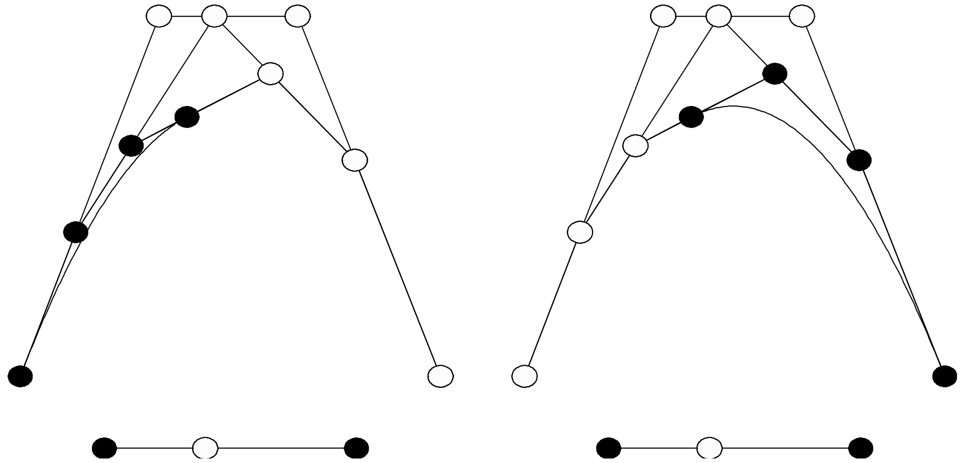

Figure 4.4: The intermediate points of the de Casteljeau algorithm for a curve, interpreted as control points of the subdivision of this curve.

Proof: [56] Let **P** denote the **polar form** of $B_n\mathbf{c}$. According to (4.28) and (4.30), the control points of $B_n\mathbf{c}|_{\Delta_j}$ have the form

$$\mathbf{c}_{j,\alpha} = \mathbf{P}\left(e_0^{\alpha_0}, \ldots, e_{j-1}^{\alpha_{j-1}}, u^{\alpha_j}, e_{j+1}^{\alpha_{j+1}}, \ldots, e_d^{\alpha_d}\right) = \mathbf{P}\left(u^{\alpha_j}, e^{\alpha-\alpha_j\epsilon_j}\right) = \mathbf{c}^{\alpha_j}_{\alpha-\alpha_j e_j}(u).$$

□

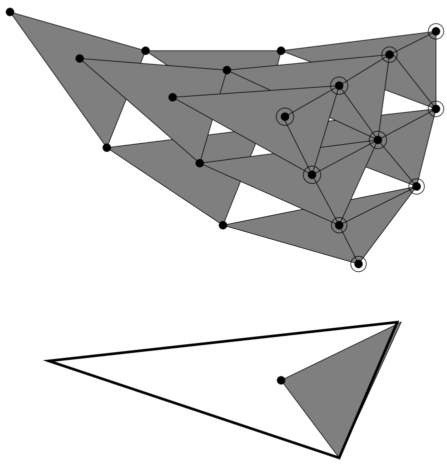

Figure 4.5: The analogy of Fig. 4.4 for surfaces.

The subdivision of control pollygons and control polyhedra, respectively, can be seen in Fig. 4.4 and Fig. 4.5.

[56]This proof is now extremely short and simple. But this is due to the fact that we used the proper concepts and did the main work before.

4.3 Spline curves

Although Bézier curves and surfaces are geometrically intuitive, the suffer from a classical problem of poynomials: They are *locally determined* which influences the curve *globally*.

Example 4.33 *Consider an arbitrary polynomial*[57]

$$p(x) = \sum_{j=0}^{n} p_j (x - x^*)^j, \qquad x \in I.$$

If we take any open *subintervall* $J := (x^* - \varepsilon, x^* + \varepsilon) \subset I$*, around* x^**, then we can determine all derivatives*[58] *as*

$$p^{(k)}(x^*) = k!\, p_k, \qquad k = 0, \dots, n,$$

which uniquely defines p everywhere*. In particular, if we modify* p *on* J*, we will see the effects of this modification everywhere.*

The same principle applies to Bézier curves and surfaces. Even if they are "quasilocal"[59], which means that due to

$$\max_{u \in \mathbb{S}_s} B_\alpha(u) = B_\alpha\left(\frac{\alpha}{|\alpha|}\right), \qquad \alpha \in \mathbb{N}_0^{s+1}, \tag{4.33}$$

the coefficient $\mathbf{c}_\alpha$ has its strongest influence at $\frac{\alpha}{|\alpha|}$, they are still global[60] in the sense that any modification of $\mathbf{c}_\alpha$ affects the curve everywhere.

Exercise 4.6 Prove (4.33). $\diamond$

To overcome the globality problem, we switch to *localized* curves. But in order to do so, we first have to make clear what we mean by "local".

Definition 4.34 *Let* $n, m > 0$ *be given with the intuition that* $n \gg m$*.*

1. *A* ***knot sequence***[61] *for* m, n *is a vector*[62] $T_{m,n} = (t_0, \dots, t_{n+m+1}) \in \mathbb{R}^{n+m+2}$ *with the properties*

$$t_0 \le t_1 \le \cdots \le t_n \le \cdots \le t_{n+m+1} \tag{4.34}$$

and

$$t_j < t_{j+m+1}, \qquad j = 0, \dots, n. \tag{4.35}$$

---

[57]This is only **scalar valued**, i.e., $d = 1$, which is totally sufficient to see the phenomenon.

[58]Recall from Analysis I that, in order to define a **derivative** at some point, we need to know the function in an **open neighborhood** of that point, cf. (Forster, 1976)(Sauer, 2014).

[59]If something does not have a certain property which one would like to have, it hase become a common bad habit in mathematics to call it "quasi–something" which means nothing, however.

[60]And not "quasiglobal", so this is a serious property.

[61]In principle, knot sequences could even be infinite, and indeed the "original" definition of Schoenbergs **cardinal spline** used $\mathbb{Z}$ as a knot sequence, and that for good reasons (Schoenberg, 1973). But here we will stay in the finite world.

[62]In the sloppy meaning of "tuple" as we will not add knot sequences. Let's call it a **finite sequence** if we really want to be precise. But do we?

2. *The number* $\mu > 0$ *for which*

$$t_{j-1} < t_j = \cdots = t_{j+\mu-1} < t_{j+\mu}$$

holds is called the ***multiplicity*** *of the* ***knot*** $t_j = \cdots = t_{j+k-1}$.

3. *The knots* $t_0, \dots, t_m$ *are called left* ***boundary knot****, the knots* $t_{n+1}, \dots, t_{n+m+1}$ *right* ***boundary knot****, respectively.*

4. *In many cases we will consider knot sequences whose boundary knots have multiplicity* $m + 1$*, that is,*

$$t_0 = \cdots = t_m, \qquad t_{n+1} = \cdots = t_{n+m+1}.$$

Remark 4.35 *The requirement* (4.35) *can be rephrased as follows: the multiplicity of any knot must not exceed* $m + 1$.

4.3.1 The de Boor algorithm

We stick to the algorithmic approach an construct a new type of curves by a *localized* variant of the de Casteljeau algorithm 4.7. To localize thigs, we make use of the knot sequence from Definition 4.34. It will be useful now to define the half open intervals

$$I_j^k := [t_j, t_{j+k+1}), \qquad j = 0, \dots, n,\ k = 0, \dots, m, \tag{4.36}$$

formed by knot sequence where j defines the location and k the spacing of the interval. Any point $x \in \mathbb{R}$ has barycentric coordinates

$$u_0(x|I_j^k) = \frac{x - t_j}{t_{j+k+1} - t_j}, \qquad u_1(x|I_j^k) = \frac{t_{j+k+1} - x}{t_{j+k+1} - t_j} \tag{4.37}$$

with respect to this interval.

Exercise 4.7 Verify the explicit expressions for the barycentric coordinates in (4.37). ◇

Now we can define the classical algorithm for spline evaluation.

Algorithm 4.36 (de Boor)
Given: ***Knot sequence*** $T_{m,n}$, control points $\mathbf{d}_0, \dots, \mathbf{d}_n \in \mathbb{R}^d$ *and*[63] $x \in [t_m, \dots, t_{n+1}]$.

1. *(Localization) Determine* $r \in \{m, \dots, n\}$ *such that* $x \in [t_r, t_{r+1})$.

2. *Initialize* $\mathbf{d}_j^0(x) = \mathbf{d}_j$, $j = r - m, \dots, r$.

3. *For* $k = 1, \dots, m$ *compute the* ***convex combination***[64]

$$\begin{aligned} \mathbf{d}_j^k(x) &= u_0\left(x|I_j^{m-k+1}\right)\mathbf{d}_{j-1}^{k-1}(x) + u_1\left(x|I_j^{m-k+1}\right)\mathbf{d}_j^{k-1}(x), \\ & \qquad j = r - m + k, \dots, r. \end{aligned} \tag{4.38}$$

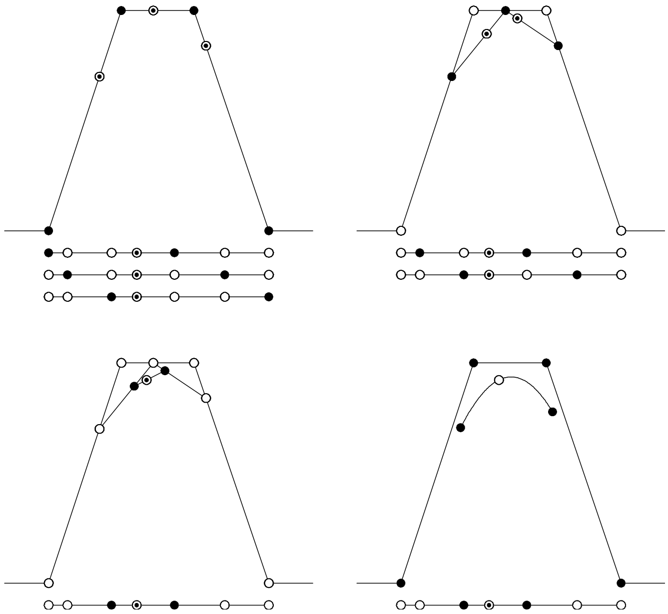

Figure 4.6: The de Boor algorithm for $m = 3$, a so called **cubic spline**. The last picture shows the curve segment on the interval $[t_r, t_{r+1}]$.

Result: *point* $\mathbf{d}_r^m(x)$.

Fig 4.6 shows what happens geometrically in the de Boor algorithm: in the k–th step we proceed all intervals that contain x and $m-k+2$ knots and use them to partition the respective edges of the control polygon. It is visible that this process drags the points much more to the center than the de Casteljeau algorithm does. By construction we already see that the resulting curve is local: on $[t_r, t_{r+1})$ the curve only depends on the control points $\mathbf{d}_{r-m}, \dots, \mathbf{d}_r$.

The index r to be determined in the first step of the algorithm is unique as long as x is not a knot.

Remark 4.37 *The index* r*, determined by* $x \in [t_i, t_{i+1})$ *does not really influence the way* how *the value is computed but mainly* which *control points contribute to this computation. Indeed, one might use the following modified version of the algorithm:*

1. *Initialize* $\mathbf{d}_j^0(x) = \mathbf{d}_j$*,* $j = 0, \dots, n$.
2. *For* $k = 1, \dots, m$ *compute*

$$\begin{aligned} \mathbf{d}_j^k(x) &= u_0\left(x|I_j^{m-k+1}\right)\mathbf{d}_{j-1}^{k-1}(x) + u_1\left(x|I_j^{m-k+1}\right)\mathbf{d}_j^{k-1}(x), \\ & \qquad j = k, \dots, n. \end{aligned} \tag{4.39}$$

3. *Pick the* r*th component from the vector* $\left(\mathbf{d}_j^m(x) : j = m, \dots, n\right)$.

[63]This is no misprint, x has to lie between the boundary knots which play a special role.

[64]It is a convex combination with nonnegative barycentric coordinates due to the choice of r!

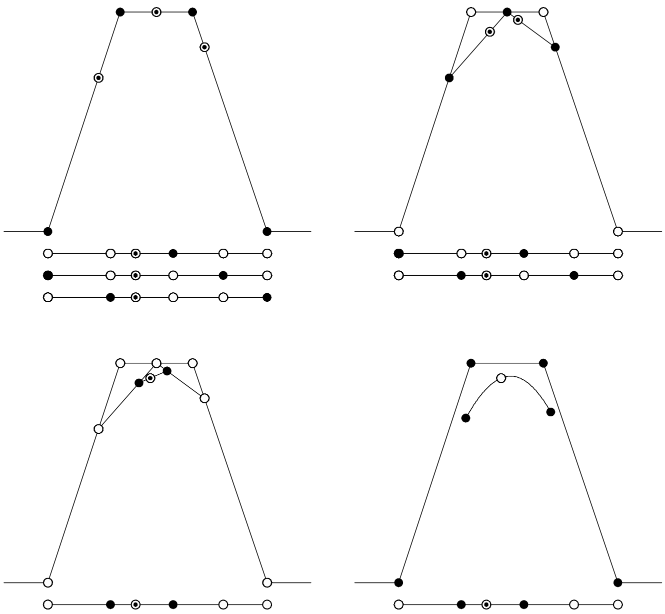

Figure 4.7: The de Boor algorithm for a cubic spline with a **double knot**.

From a computational point of view, this does not make sense as a point evaluation algorithm when $n \gg m$*, since the modification computes the function everywhere.*

Definition 4.38 *The **spline curve*** $N_{m,T}\mathbf{d}$ *is defined as the function* $x \mapsto \mathbf{d}_j^m(x)$*,* $x \in [t_m, \dots, t_{n+1}]$*. The **control polygon** of this spline curve is* $\mathbf{d} = (\mathbf{d}_j : j = 0, \dots, n)$*.*

Next, we collect some properties of the spline curve that follow directly from Algorithm 4.36.

Proposition 4.39 *For a knot sequence* $T_{m,n}$ *and a control polygon* $\mathbf{d}$ *we have*

1. *(**convex hull property**):*

$$N_{m,T}\mathbf{d}\left([t_r, t_{r+1})\right) \subseteq [\![\mathbf{d}_{r-k} : k = 0, \dots, m]\!], \qquad r = m, \dots, n. \tag{4.40}$$

2. *(**interpolation** at* m*–fold knots): if* $x = t_{j-m+1} = \cdots = t_j$*, then* $N_{m,T}\mathbf{d}(x) = \mathbf{d}_{j-m}$*.*

3. *(de Casteljau): for* $n = m$ *and* $t_0 = \cdots = t_m$*,* $t_{m+1} = \cdots = t_{2m+1}$*, we have* $N_{m,T}\mathbf{d} = B_m\mathbf{d}$*.*

4. *(**pieceweise polynomial**):*

$$N_{m,T}\mathbf{d}|_{(t_j,t_{j+1})} \in \Pi_m, \qquad j = m, \dots, n.$$

Proof: The property 1) follows from the already mentioned fact that all barycentric coordinates appearing in the process are nonnegative.

For 2) we remark that whenever $t_{i-m+1} = \cdots = t_i$ each of the intervals I_j^{m-k+1} in (4.39) has $t_i = x$ as left endpoint which yields $u_0\left(x|I_j^{m-k+1}\right) = 1$, $u_1\left(x|I_j^{m-k+1}\right) = 0$ and therefore $\mathbf{d}_{r-k}^k(x) = \mathbf{d}_{r-m}$.

Moreover, 3) is obvious and 4) a direct consequence of the explicit expression (4.37) for the barycentric coordinates which shows that in each step $\mathbf{d}_j^{k-1}(x)$ is multiplied by a polynomial of degree 1 in x. □

Corollary 4.40 *If* $T_{m,n}$ *has* $m+1$*–fold boundary knots, the spline curve admits* ***end point interpolation****, that is*

$$N_{m,T}\mathbf{d}\,(t_m) = \mathbf{d}_0, \qquad N_{m,T}\mathbf{d}\,(t_{n+1}) = \mathbf{d}_n. \tag{4.41}$$

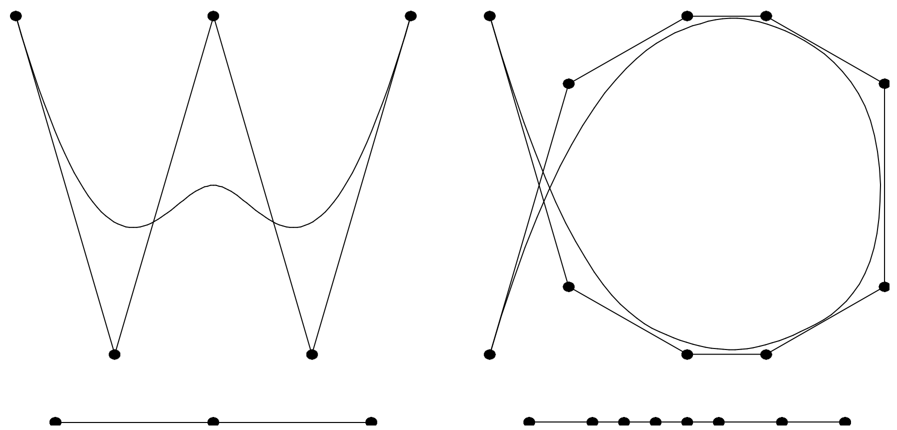

Figure 4.8: Two cubic ($m = 3$) spline curves with boundary knots of multiplicity 4.

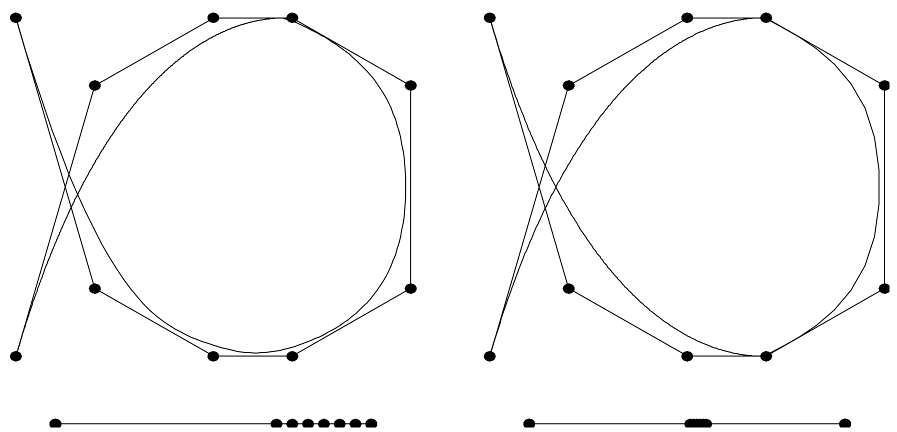

Figure 4.9: Variation of the knots of the examples in Fig. 4.8.

4.3.2 B–splines

Like we did before with Bernstein–Bézier basis polynomials, we will derive an explicit representation of the Splines curve with respect to appropriate basis functions. Again this will be done by "dualizing" the evaluation algorithm, this time the **de Boor algorithm**. The definition follows is a straightforward way from noting that the de Boor algorithm is *linear* in $\mathbf{d}$, that is $N_{m,T}(\mathbf{d}+\mathbf{d}') = N_{m,T}\mathbf{d} + N_{m,T}\mathbf{d}'$. Writing $\mathbf{d}$ formally as

$$\begin{aligned}\mathbf{d} &= (\mathbf{d}_0,\dots,\mathbf{d}_n) = \sum_{j=0}^{n} (0,\dots,0,\mathbf{d}_j,0,\dots,0) = \sum_{j=0}^{n} \mathbf{d}_j \underbrace{(0,\dots,0,1,0,\dots,0)}_{=:\delta_j} \\ &= \sum_{j=0}^{n} \mathbf{d}_j\, \delta_j\end{aligned}$$

with the scalar[65] sequences $\delta_0,\dots,\delta_n$, we see that

$$N_{m,T}\mathbf{d} = \sum_{j=0}^{n} \mathbf{d}_j\, N_{m,T}\delta_j =: \sum_{j=0}^{n} \mathbf{d}_j\, N_j^m(\cdot|T). \tag{4.42}$$

Definition 4.41 (B–spline) *The jth **B–spline** of degree m with respect to T is defined by means of the de Boor algorithm as*

$$N_j^m(\cdot|T) := N_{m,T}\,\delta_j. \tag{4.43}$$

The definition of the B–spline was cheap and simple, and though this is already sufficient to plot them like in Fig 4.10, the task will be to give meaning to definition by deriving properties of the functions N_j^m.

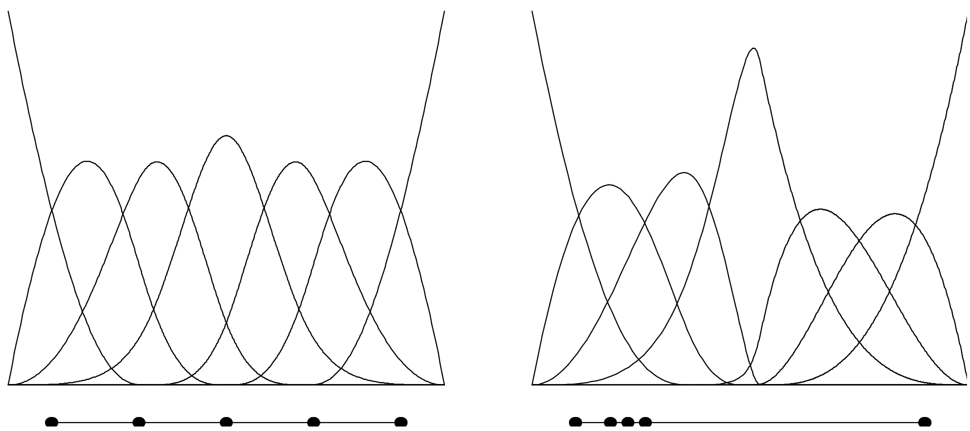

Figure 4.10: A small collection of **cubic** ($m = 3$) B–splines with different knot distributions..

[65]We might also say "$d = 1$".

Lemma 4.42 *The B–splines are nonnegative functions with* ***compact support****. More precisely,*

$$N_j^m(x|T) > 0, \quad x \in (t_j, t_{j+m+1}), \qquad \text{and} \qquad N_j^m(x|T) = 0, \quad x \notin [t_j, t_{j+m+1}] \tag{4.44}$$

In particular, we have the local relation

$$N_{m,T}\mathbf{d}(x) = \sum_{k=j-m}^{j} \mathbf{d}_k N_k^m(x|T), \qquad x \in [t_j, t_{j+1}), \qquad j = m, \dots, n. \tag{4.45}$$

Proof: Let us recall first that to determine the value of $N_{m,T}\mathbf{d}(x)$ at the position $x \in [t_r, t_{r+1})$ we used the control points $d_{r-m}, \dots, d_r$. Since the other coefficients do not matter there, we must have $N_j^m(x|T) = 0$, $j \notin \{r-m, \dots, r\}$ or, equivalently, $N_j^m(x|T) \neq 0$ implies $j \in \{r-m, \dots, r\}$ or $r \in \{j, \dots, j+m\}$. Therefore, the B–splines $N_j^m(x|T)$ vanishes outside $[t_j, \dots, t_{j+m+1}]$. If, on the other hand, $x \in (t_r, t_{r+1})$ and $r \in \{j, \dots, j+m\}$ then[66]

$$(t_r, t_{r+1}) \subseteq (t_j, t_{j+m-k+1}) \subset I_j^{m-k+1}, \qquad j = r-m+k, \dots, r, \; k = 1, \dots, m,$$

all all barycentric coordinates in (4.39) are strictly positive and therefore the computed coefficients for $d = \delta_j$ in the first step of the algorithm satisfy

$$d_j^1(x) = u_0\left(x|I_j^m\right) \underbrace{d_{j-1}^0(x)}_{=0} + u_1\left(x|I_j^m\right) \underbrace{d_j^0(x)}_{=1} = u_1\left(x|I_j^m\right) > 0$$

and

$$d_{j+1}^1(x) = u_0\left(x|I_{j+1}^m\right) d_j^0(x) + u_1\left(x|I_{j+1}^m\right) d_j^0(x) = u_0\left(x|I_{j+1}^m\right) > 0$$

as well as $d_k^1(x) = 0$, $k \notin \{j, j+1\}$. The same computations then give that

$$d_j^2(x), d_{j+1}^2(x), d_{j+2}^2(x) > 0, \qquad d_k^2(x) = 0, \quad k \notin \{j, j+1, j+2\},$$

and, by induction, that for $k = 1, \dots, m$

$$d_j^k(x), \dots, d_{j+k}^2(x) > 0, \qquad d_\ell^k(x) = 0, \quad \ell \notin \{j, \dots, j+k\}, \tag{4.46}$$

and since $r \in \{j, \dots, j+m\}$ it follows by setting $k = m$ in (4.46) that

$$N_j^m(x|T) = N_{m,T}\delta_j(x) = d_r^m(x) > 0.$$

. $\square$

Lemma 4.43 *The B–splines form a nonnegative* ***partition of unity*** *on the interval enclosed by the extremal boundary knots, that is,*

$$\sum_{j=0}^{n} N_j^m(x|T) \equiv 1, \qquad x \in [t_m, t_{n+1}]. \tag{4.47}$$

[66]Since $r \geq j$ and $r+1 \leq j+m+1$.

Proof: Whenever $d_0 = \cdots = d_n = 1$, thus $d_j^0(x) = 1$, $j = 0, \dots, n$, we get from (4.39) and by induction[67] on k, that[68]

$$\begin{aligned} d_j^k(x) &= u_0\left(x|I_j^{m-k+1}\right) \underbrace{d_{j-1}^{k-1}(x)}_{=1} + u_1\left(x|I_j^{m-k+1}\right) \underbrace{d_j^{k-1}(x)}_{=1} \\ &= u_0\left(x|I_j^{m-k+1}\right) + u_1\left(x|I_j^{m-k+1}\right) = 1, \end{aligned}$$

holds for $j = k, \dots, m$. □

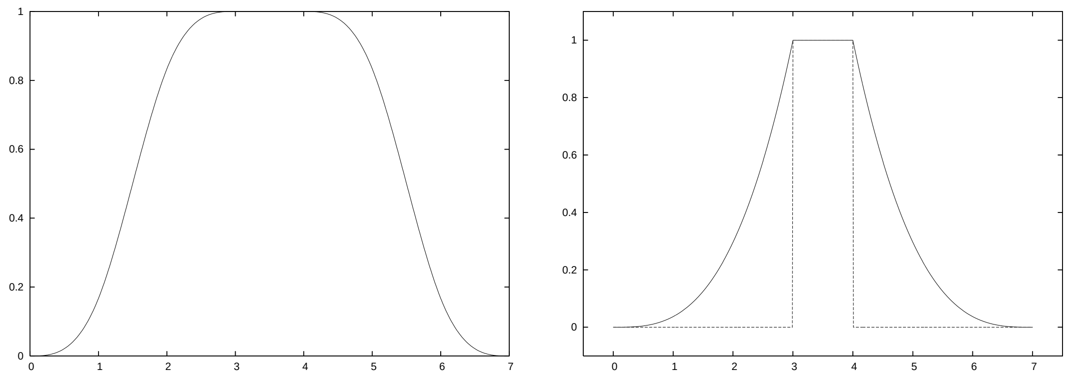

Figure 4.11: Plot of the function

$$\sum_{j=0}^{n} N_j^m(\cdot|T)$$

for the knot sequence $(0, 1, 2, 3, 4, 5, 6, 7)$ and $m = 3$ *(left)* which only equals 1 in the interval $[3, 4]$.

With triple boundary knots there is a nondifferentiable corner at 3 and 4 while the knot sequence 3 and 4, while the knot sequence $(3, 3, 3, 3, 4, 4, 4, 4)$ gives a sharp jump there *(right)*.

The following recurrence relation is due to *Carl de Boor* (Boor, 1972) and was the basis for the de Boor algorithm. That we introduce things in the opposite way here is due to the fact that we want to follow a totally algorithmic approach as introduced in (Sauer, 1996). In principle, this shows that the two approaches are fully equivalent: the algorithm follows from the recurrence and the recurrence from the algorithm.

Theorem 4.44 (Recurrence relation for B–splines)

[67] After so many inductions we should now be used to it and therefore we can get a little bit more informal with them. **Warning:** Don't do that at home, dear children.

[68] To some extend working with splines always needs a bit of the skill of taming the beastly indices.

1. *The B–splines of degree zero have the form*

$$N_j^0(x|T) = \chi_{[t_j,t_{j+1})}(x) = \begin{cases} 1 & x \in [t_j, t_{j+1}), \\ 0 & x \notin [t_j, t_{j+1}), \end{cases} \qquad j = 0, \dots, n+m. \tag{4.48}$$

2. *For $k \geq 1$ one has*

$$\begin{aligned} N_j^k(x|T) &= u_1\left(x|I_j^k\right) N_j^{k-1}(x|T) + u_0\left(x|I_{j+1}^k\right) N_{j+1}^{k-1}(x|T), \\ & j = 0, \dots, n+m-k. \end{aligned} \tag{4.49}$$

Remark 4.45

1. *In the expressions* $N_j^k(\cdot|T)$ *in* (4.49) *we have to interpret* $T = T_{m,n} = (t_0, \dots, t_{m+n+1})$ *as* $T = T_{k,m+n-k}$ *which means that we will have* $n + m - k$ *B–splines of degree* k *in this process.*

2. *Written explicitly,* (4.49) *reads as*

$$N_j^k(x|T) = \frac{x - t_j}{t_{j+k} - t_j} N_j^{k-1}(x|T) + \frac{t_{j+k+1} - x}{t_{j+k+1} - t_{j+1}} N_{j+1}^{k-1}(x|T), \tag{4.50}$$

   $j = 0, \dots, n+m-k.$

3. *The formula* (4.50) *is undefined for* $k+1$*–fold knots but in this case also the B–spline* N_j^{k-1} *or* N_{j+1}^{k-1} *makes no sense since its support would be at most one point. This motivates the convention that such functions are set to zero as well as the respective quotient* $0/0$ *in* (4.49).

4. *Note that* (4.49) *uses barycentric coordinates with respect to different reference intervals which do* ***not*** *sum to* 1.

Proof of Theorem 4.44: The identity (4.48) follows directly from the de Boor algorithm. To express $N_j^m(\cdot|T)$ by means of B–splines of degree $m-1$ we use the intermediate points $d_j^1(x)$, $j = 1, \dots, n$, from (4.39) and write[69]

$$N_{m,T} d(x) = \sum_{j=0}^n d_j N_j^m(x|T) = \sum_{j=1}^n d_j^1(x)\, N_j^{m-1}(x|T). \tag{4.51}$$

For the particular scalar control polygon $d = \delta_k$, the identity (4.39) yields

$$d_j^1(x) = u_0\left(x|I_j^m\right) d_{j-1} + u_1\left(x|I_j^m\right) d_j = \begin{cases} u_1\left(x|I_k^m\right) & j = k, \\ u_0\left(x|I_{k+1}^m\right) & j = k+1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.52}$$

Substituting (4.52) in (4.51) then results in

$$N_j^m(x|T) = u_1\left(x|I_j^m\right) N_j^{m-1}(x|T) + u_0\left(x|I_{j+1}^m\right) N_{j+1}^{m-1}(x|T), \tag{4.53}$$

which completes the proof. □

[69] To these intermediate points we apply a de Boor algorithm of degree $m-1$. That is all behind (4.51).

4.3.3 The spline space

The "B" in "B–spline" has its reasons, of course. As might be expected, the letter stands for "basis", so let us identify the space for which they are a basis.

Definition 4.46 *For $m \in \mathbb{N}$ and a* ***knot sequence*** *T the* ***spline space*** *$\mathbb{S}_m(T)$ is defined as the set of all*

1. ***piecewise polynomials*** *of degree m*
$$f|_{[t_j,t_{j+1})} \in \Pi_m, \quad j = m, \dots, n, \tag{4.54}$$

2. *that are differentiable of order $m - \mu$ at a knot t_j of* ***multiplicity*** *μ, i.e. for $t_{j-1} < t_j = \cdots = t_{j+\mu-1} < t_{j+\mu}$ we have*
$$f \in C^{m-\mu}(t_{j-1}, t_{j+\mu}). \tag{4.55}$$

Exercise 4.8 Show that the spline space $\mathbb{S}_m(T)$ is a **vector space**. $\diamond$

The next result, the famous **Curry–Schoenberg**[70] theorem, is the basis of spline theory and shows that B–splines are indeed a basis[71] of the spline space.

Theorem 4.47 (Curry–Schoenberg) *The B–splines $N_j^m(\cdot|T)$, $j = 0, \dots, n$, are a basis of $\mathbb{S}_m(T)$.*

For this theorem we have to prove quite a bit. Although (4.54) follows directly from the de Boor algorithm, as we already know from Proposition 4.39, we still have to show

1. the differentiability of the B–splines around knots,
2. the linear independence of the B–splines
3. that the dimension of the spline space equals $n + 1$.

This could be done via **blossoming**, see (Seidel, 1989; Sauer, 1996), but we will follow a more direct approach which, as a side effect, also gives us a formula for the derivative of a spline curve that we'll need anyway. To make our life easier, we make one more assumption, namely that

$$t_m < t_{m+1} \qquad \text{und} \qquad t_n < t_{n+1}, \tag{4.56}$$

which means that no **boundary knot** is an **inner knot**[72] which is in particular the case for $(m+1)$–fold boundary knots.

[70]Originally, H. B. Curry was a pure number theorist but during World War II he and I. Schoenberg, the "father of splines" worked together in military research where they developed a first theory of splines that remained unpublished, however, until 1966 (Curry & Schoenberg, 1966), "for no good reason" as Schoenberg once called it. So splines are a child of the war: $\pi\tau o\lambda\varepsilon\mu o\sigma$ $\pi\alpha\tau\eta\rho$ $\pi\alpha\nu\tau\omega\nu$ as Heraklit says.

[71]No, THE basis!

[72]Somehow the condition make sense when written this way.

Lemma 4.48 *For* $x \in \mathbb{R} \setminus T$ *we have*

$$\frac{d}{dx} N_j^m (x|T) = \frac{m}{t_{j+m} - t_j} N_j^{m-1} (x|T) - \frac{m}{t_{j+m+1} - t_{j+1}} N_{j+1}^{m-1} (x|T), \qquad j = 0, \dots, n. \tag{4.57}$$

Equation (4.57) does not make sense if t_j or t_{j+1} is a knot of multiplicity $m+1$. However, in that case the support interval of the respective B–spline would be the empty set and therefore the function is zero and we apply the convention that then the whole term in the sum (4.57) is zero.

Proof: Induction on m, where the case $m = 1$ can be easily checked by hand. Since restricted to any open and convex subset U of $\mathbb{R} \setminus T$ the B–spline N_j^m is a polynomial, we can differentiate as much as we want and get[73]

$$\begin{aligned}
&\left(\frac{d}{dx} N_j^m (\cdot|T) \right)(x) \\
&= \left(\frac{d}{dx} \left(\frac{\cdot - t_j}{t_{j+m} - t_j} N_j^{m-1} (\cdot|T) + \frac{t_{j+m+1} - \cdot}{t_{j+m+1} - t_{j+1}} N_{j+1}^{m-1} (\cdot|T) \right) \right)(x) \\
&= \frac{1}{t_{j+m} - t_j} N_k^{m-1} (\cdot|T) - \frac{1}{t_{j+m+1} - t_{j+1}} N_{j+1}^{m-1} (\cdot|T) \\
&\quad + \frac{x - t_j}{t_{j+m} - t_j} \frac{d}{dx} \left(N_j^{m-1} (\cdot|T) \right)(x) + \frac{t_{j+m+1} - x}{t_{j+m+1} - t_{j+1}} \frac{d}{dx} \left(N_{j+1}^{m-1} (\cdot|T) \right)(x).
\end{aligned} \tag{4.58}$$

The induction hypothesis and yet another application of the B–spline recurrence then yield

$$\begin{aligned}
&\frac{x - t_j}{t_{j+m} - t_j} \frac{d}{dx} \left(N_j^{m-1} (\cdot|T) \right)(x) \\
&= \frac{x - t_j}{t_{j+m} - t_j} \left(\frac{m-1}{t_{j+m-1} - t_j} N_j^{m-2} (x|T) - \frac{m-1}{t_{j+m} - t_{j+1}} N_{j+1}^{m-2} (x|T) \right) \\
&= \frac{m-1}{t_{j+m} - t_j} \underbrace{\left(\frac{x - t_j}{t_{j+m-1} - t_j} N_j^{m-2} (x|T) + \frac{t_{j+m} - x}{t_{j+m} - t_{j+1}} N_{j+1}^{m-2} (x|T) \right)}_{= N_j^{m-1}(x|T)} \\
&\quad - \frac{m-1}{t_{j+m} - t_j} \underbrace{\left(\frac{t_{j+m} - x}{t_{j+m} - t_{j+1}} + \frac{x - t_j}{t_{j+m} - t_{j+1}} \right)}_{= (t_{j+m} - t_j)/(t_{j+m} - t_{j+1})} N_{j+1}^{m-2} (x|T) \\
&= \frac{m-1}{t_{j+m} - t_j} N_j^{m-1} (x|T) - \frac{m-1}{t_{j+m} - t_{j+1}} N_{j+1}^{m-2} (x|T)
\end{aligned} \tag{4.59}$$

[73]Now comes the point where we have to work a little bit. In each approach to splines there is one technical and computational proof. So let's get over it!

as well as

$$
\begin{aligned}
&\frac{t_{j+m+1}-x}{t_{j+m+1}-t_{j+1}}\frac{d}{dx}\left(N_{j+1}^{m-1}(\cdot|T)\right)(x)\\
&= \frac{t_{j+m+1}-x}{t_{j+m+1}-t_{j+1}}\left(\frac{m-1}{t_{j+m}-t_{j+1}}N_{j+1}^{m-2}(x|T)-\frac{m-1}{t_{j+m+1}-t_{j+2}}N_{j+2}^{m-2}(x|T)\right)\\
&= \frac{m-1}{t_{j+m+1}-t_{j+1}}\underbrace{\left(\frac{x-t_{j+1}}{t_{j+m}-t_{j+1}}+\frac{t_{j+m+1}-x}{t_{j+m}-t_{j+1}}\right)}_{=(t_{j+m+1}-t_{j+1})/(t_{j+m}-t_{j+1})}N_{j+1}^{m-2}(x|T)\\
&\quad -\frac{m-1}{t_{j+m+1}-t_{j+1}}\underbrace{\left(\frac{x-t_{j+1}}{t_{j+m}-t_{j+1}}N_{j+1}^{m-2}(x|T)+\frac{t_{j+m+1}-x}{t_{j+m+1}-t_{j+2}}N_{j+2}^{m-2}(x|T)\right)}_{=N_{j+1}^{m-1}(x|T)}\\
&= \frac{m-1}{t_{j+m}-t_{j+1}}N_{j+1}^{m-2}(x|T)-\frac{m-1}{t_{j+m+1}-t_{j+1}}N_{j+1}^{m-1}(x|T).
\end{aligned}
\tag{4.60}
$$

Substitutin (4.59) and (4.59) into (4.58) we get

$$
\begin{aligned}
&\left(\frac{d}{dx}N_j^m(\cdot|T)\right)(x)\\
&= \frac{1}{t_{j+m}-t_j}N_j^{m-1}(x|T)-\frac{1}{t_{j+m+1}-t_{j+1}}N_{j+1}^{m-1}(x|T)+\frac{m-1}{t_{j+m}-t_j}N_j^{m-1}(x|T)\\
&\quad -\frac{m-1}{t_{j+m}-t_{j+1}}N_{j+1}^{m-2}(x|T)+\frac{m-1}{t_{j+m}-t_{j+1}}N_{j+1}^{m-2}(x|T)-\frac{m-1}{t_{j+m+1}-t_{j+1}}N_{j+1}^{m-1}(x|T)\\
&= \frac{m}{t_{j+m}-t_j}N_k^{m-1}(x|T)-\frac{m}{t_{j+m+1}-t_{j+1}}N_{j+1}^{m-1}(x|T),
\end{aligned}
$$

which verifies (4.57). □

Exercise 4.9 Verify (4.57) for $m = 1$. What does that mean geometrically? ◇

This also gives us the derivative of a spline curve in a straightforward way.

Corollary 4.49 *For* $x \in \mathbb{R} \setminus T$

$$
\frac{d}{dx}N_{m,T}\,\mathbf{d}(x) = m\sum_{j=0}^{n+1}\frac{\mathbf{d}_j-\mathbf{d}_{j-1}}{t_{j+m}-t_j}N_j^{m-1}(x|T),
\tag{4.61}
$$

where $\mathbf{d}_{n+1} = \mathbf{d}_{-1} = 0$[74].

Proof: From Lemma 4.48 we get

$$
\begin{aligned}
\frac{d}{dx}N_{m,T}\,\mathbf{d}(x) &= m\sum_{j=0}^{n}\mathbf{d}_j\left(\frac{N_j^{m-1}(x|T)}{t_{j+m}-t_j}-\frac{N_{j+1}^{m-1}(x|T)}{t_{j+1+m}-t_{j+1}}\right)\\
&= m\sum_{j=0}^{n+1}\frac{\mathbf{d}_j-\mathbf{d}_{j-1}}{t_{j+m}-t_j}N_j^{m-1}(x|T).
\end{aligned}
$$

□

[74] And once again $N_j^{m-1}(x|T)/(t_{j+m}-t_j) \equiv 0$.

Proposition 4.50 $N_j^m(\cdot|T) \in \mathbb{S}_m(T)$, $j = 0, \dots, n$.

Proof: Once more inductionn on m[75]. For $m = 0$ all knots have to be simple as the maximal multiplicity is bounded by $m + 1 = 1$ and the B–splines are piecewise constant functions that *trivially* belong to $C^{-1}(\mathbb{R})$.

To prove $m - 1 \to m$, $m \geq 1$, we only need to check differentiability. At an $m + 1$–fold knot the de Boor algorithm already gives us a discontinuity: the B–spline must have the values 0 and 1 there simultaneously. For a knot $t \in (t_j, t_{j+m+1})$ of multiplicity $\mu \leq m$, we can differentiate in an open interval U with $U \cap T = \{t\}$ and the derivative is well–defined according to (4.57) and $m - 1 - \mu$ times continously differentiable by the induction hypothesis. Hence, N_j^m is $m - \mu$ times continously differentiable in U. □

Lemma 4.51 *The B–splines* $N_j^m(\cdot|T)$, $j = 0, \dots, n$, *are* ***linearly independent****.*

Proof: Yet another induction on m where $m = 0$ is clear since the B–splines have disjoint support. For $m \to m + 1$ we assume that no knot has multiplicity $m + 1$ and there were coefficients $\mathbf{d} = (d_j : j = 0, \dots, n)$, such that

$$0 = N_{m,T}\,\mathbf{d} = \sum_{j=0}^{n} d_j\, N_j^m(\cdot|T)\,. \tag{4.62}$$

Taking derivative of both sides yields for $x \in \mathbb{R} \setminus T$ that

$$0 = \frac{d}{dx} N_{m,T}\,\mathbf{d} = m \sum_{j=0}^{n+1} \frac{d_j - d_{j-1}}{t_{j+m} - t_j} N_j^{m-1}(\cdot|T)\,.$$

According to Proposition 4.50 the right hand side of this identity is continuous[76] and the induction hypothesis gives $d_j = d_{j-1}$, hence

$$0 = d_0 = d_1 = d_2 = \cdots = d_n = d_{n+1} = 0.$$

□

Lemma 4.52 $\dim \mathbb{S}_m(T) = n + 1$.

Proof: On the interval $I_m = (t_m, t_{m+1})$ which is $\neq \emptyset$ due to (4.56) we define an arbitrary polynomial $p_m \in \Pi_m$. Let $t_{m+1} = t_{m+2} = \cdots = t_{m+\mu} < t_{m+\mu+1}$ be a knot of multiplicity μ. Expanding a polynomial p_{m+1} on $(t_{m+1}, t_{m+\mu+1})$ as

$$p_{m+1} = \sum_{j=0}^{m} \frac{a_j}{j!} (x - t_{m+1})^j\,,$$

the differentiability conditions of the spline space take the form

$$p_m^{(j)}(t_{m+1}) = p_{m+1}^{(j)}(t_{m+1}) = a_j, \qquad j = 0, \dots, m - \mu.$$

[75] How else should one make use of recurrence relations anyway?
[76] No $m + 1$–fold knots left.

Hence the space of all piecewiese polynomials of proper differentiability on the first two intervals has the dimension $m + 1 + \mu$ – the $m + 1$ degrees of freedom of p_m and the μ free parameters of p_{m+1}. If the next knot $t_{m+\mu+1}$ has multiplicity ν, we get ν more free parameters and so on. Inductively we can conclude that for the knot sequence

$$T_\ell = (t_m, \dots, t_{m+\ell+1}), \qquad t_m < t_{m+1}, \quad t_{m+\ell} < t_{m+\ell+1},$$

plus appropriate boundary knot we have $\dim \mathbb{S}_m(T_\ell) = m + \ell + 1$ and the case $\ell = n - m$ proves the lemma. □

Proof of Theorem 4.47: We only have to connect the pieces we collected so far. According to Proposition 4.50

$$\operatorname{span}\left\{ N_j^m(\cdot|T) \ : \ j = 0, \dots, n \right\} \subseteq \mathbb{S}_m(T),$$

but since due to the linear independence of the B–Splines and because of Lemma 4.52 the dimensions of the two vector spaces coincide, the spaces must be identical. □

4.3.4 Interpolation

The name "spline" is due to a physical device, a flexible ruler used for the interpolation of curves, initially in ship constructions[77]. Let us first clarify what interpolation means.

Definition 4.53 (Interpolation) *Given **sites**[78] $x_j \in I$ and **data** $\mathbf{y}_j \in \mathbb{R}^d$, $j = 0, \dots, n$, the **interpolation problem** consists of finding a function $\mathbf{f} : I \to \mathbb{R}$ such that*

$$\mathbf{f}(x_j) = \mathbf{y}_j, \qquad j = 0, \dots, n. \tag{4.63}$$

Remark 4.54

1. *The name* interpolation *has been invented by Wallis in 1655, according to (Bauschinger, 1900), see (Gasca & Sauer, 2000). Originally, it was used to estimate non existing values in tables, for example logarithms. Mostly, polynomials were used for that purpose and we will see in a moment why.*

2. *Of course, all the points have to be different, otherwise the respective data values would also have to coincide which makes the problem redundant.*

3. *Clearly, the interpolation problem* (4.63) *has many solutions and a major question is how to restrict the functions*[79] *such that the solution becomes unique.*

4. *By working on the components in $\mathbb{R}^d$ separately, it suffices to consider the case* $d = 1$.

[77] This can be seen from the German word "**Straklatte**" for this device.
[78] This terminology is due to Carl de Boor, other people use "points", "locations", even "knots".
[79] And perhaps the points.

5. *If the function* f *is chosen from a linear space*[80] *spanned by* $f_0, \dots, f_m$*, it can be represented as*

$$f = \sum_{k=0}^{m} a_k f_k(x_j), \qquad \mathbf{a} = [a_k : k = 0, \dots, m] \in \mathbb{R}^{m+1} \tag{4.64}$$

and the interpolation problem can be written as

$$f(x_j) = \sum_{k=0}^{m} a_k f_k(x_j), \quad j = 0, \dots, n,$$

which takes the matrix form

$$\underbrace{\begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix}}_{=:\mathbf{f}} = \underbrace{\begin{bmatrix} f_0(x_0) & \dots & f_m(x_0) \\ \vdots & \ddots & \vdots \\ f_0(x_n) & \dots & f_m(x_n) \end{bmatrix}}_{=:\mathbf{F}(X)} \mathbf{a}. \tag{4.65}$$

The matrix $\mathbf{F}(X)$ *is called the* ***collocation matrix*** *for the basis* $\{f_0, \dots, f_m\}$ *and the sites* $X = \{x_0, \dots, x_n\}$*.*

6. *The linear system* (4.65) *has a unique solution only if the collocation matrix* $\mathbf{F}(X)$ *is a square one, but clearly this is only a necessary condition and in no way sufficient. Anyway, it means that the dimension of the space and the number of interpolation conditions has to coincide.*

7. *The order of the sites is not relevant for the solvability of the interpolation problem or its solution, but it may affect*numerical *properties of algorithms to solve it.*

The simplest universal interpolation space in one variable are the polynomials.

Theorem 4.55 *The interpolation problem* $f(x_j) = y_j$*,* $j = 0, \dots, n$*, for distinct* x_j *always has a unique solution in* Π_n*.*

Proof: We can write down the solution explicitly as[81]

$$f = \sum_{j=0}^{n} y_j \prod_{k \neq j} \frac{\cdot - x_k}{x_j - x_k} \qquad \Rightarrow \qquad f(x_j) = y_j, \quad j = 0, \dots, n.$$

For uniqueness suppose that f and g are solutions, then $p = f - g$ is a polynomial of degree n with

$$p(x_j) = f(x_j) - g(x_j) = y_j - y_j = 0, \qquad j = 0, \dots, n,$$

[80] To be very precise: a real vector space.

[81] Please do not use this to really *compute* the interpolating polynomial numerically as this formula is very **ill–conditioned**.

hence

$$p(x) = \underbrace{(x - x_0) \cdots (x - x_n)}_{\in \Pi_{n+1}} q(x)$$

which is only in Π_n if $q = 0$. $\square$

Exercise 4.10 Is unique interpolation at arbitrary n distinct sites $x_1, \ldots, x_n$ possible with the space $\{x, \cdots, x^n\}$? Prove it or give a counterexample. $\diamond$

An immediate consequence of Theorem 4.55 is that interpolation with polynomials of degree at most n is only possible at $\le n+1$ sites: for more sites, interpolate at $x_0, \ldots, x_n$ by a unique f and then require $y_{n+1} \neq f(x_{n+1})$, this problem cannot be solved.

This, on the other hand, means that spline interpolation cannot be so simple any more. Since restricted to any notrivial knot interval (t_j, t_{j+1}), splines are in Π_m, no such interval may contain more than $m + 1$ interpolation sites. But this does not take into account the interaction between the intervals due to differentiability, so the full requirement is stronger and as follows.

Theorem 4.56 (Schoenberg–Whitney) *The splines space* $\mathbb{S}_m(T)$ *with basis* $N_j^m(\cdot|T)$, $j = 0, \ldots, n$, *allows for unique interpolation at sites*[82] $X = \{x_0 < x_1 < \cdots < x_n\}$ *if and only if*

$$t_j < x_j < t_{j+m+1}, \qquad j = 0, \ldots, n. \tag{4.66}$$

We will not give the full proof of this theorem[83] even if it quite interesting, but we can easily show that the condition (4.66) is necessary.

Proof of Theorem 4.56, "$\Rightarrow$": Suppose there exists some j violating (4.66) which means that either[84] $x_j \le t_j$ or $x_j \ge t_{j+m+1}$. Let us start with the first case. Since $N_k^m(\cdot|T)$ is supported[85] on $[t_k, t_{k+m+1}]$, it follows that

$$N_k^m(x_j|T) = 0, \qquad k = j, \ldots, n, \tag{4.67}$$

and therefore the first $j + 1$ *rows* of the collocation matrix are of the form

$$\begin{bmatrix} N_0(x_0|T) & \ldots & N_{j-1}(x_0|T) & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ N_0(x_j|T) & \ldots & N_{j-1}(x_j|T) & 0 & \ldots & 0 \end{bmatrix}$$

which means that they are linearly dependent. But this means that the collocation matrix cannot be invertible, hence the interpolation is unsolvable in general.

In the other case, $x_j \ge t_{j+m+1}$, we have

$$N_k^m(x_j|T) = 0, \qquad k = 0, \ldots, j, \tag{4.68}$$

[82] There is no restriction in ordering them in incrasing size.

[83] Which has even more nice consequences, the collocation matrices for splines are in fact **banded** and **totally nonnegative** which makes the numerically exceptionally well to handle.

[84] Both at the same time is clearly impossible.

[85] Recall once more: the **support** is the **closure** of the set of all points where a function is nonzero.

and since $x_j \le x_\ell$, $j \le \ell$, we also get

$$N_k^m(x_l|T) = 0, \qquad k = 0,\dots,j,\ l = j,\dots,n. \tag{4.69}$$

Consquently, the first $j+1$ *columns* of the collocation matrix look as follows:

$$\begin{bmatrix} N_0^m(x_0|T) & \dots & N_j^m(x_0|T) \\ \vdots & \ddots & \vdots \\ N_0^m(x_{j-1}|T) & \dots & N_j^m(x_{j-1}|T) \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix},$$

and are linearly dependent. □

The classical interpolation problem that "initiated" splines was not interpolation at arbitrary sites but interpolation at **simple knots**. Indeed, if we set $x_j = t_{m+j}$, $j = 0,\dots,n-m+1$, then

$$t_j < t_{m+j} = x_j < t_{j+m+1}, \qquad j = 0,\dots,n-m+1,$$

and the sites satisfy the necessary condition of Schoenberg–Whitney. However, these are only $n-m+2$ conditions so far and the spline space has dimension $n+1$, hence, we have to request $n+1-(n-m+2) = m-1$ further conditions. The most popular ones are the so–called **natural boundary conditions** and to distribute them symmetrically on both ends of the spline, it is conveniet that $m-1$ is even or m odd[86]

Theorem 4.57 (Natural spline) *Let* $m = 2r+1 \in \mathbb{N}$ *let* $T = T_{m,n}$ *be a knot sequence with* ***simple knots****. Then, for* $n \ge m+r$ *and any given values* $\mathbf{y}_j$, $j = 0,\dots,n-m+1$, *there exists a* unique *spline curve* $N_{m,T}\mathbf{d}$, *such that*

$$\begin{aligned} N_{m,T}\,\mathbf{d}\,(t_{m+j}) &= \mathbf{y}_j, \qquad j = 0,\dots,n-m+1, & (4.70)\\ N_{m,T}^{(k)}\,\mathbf{d}\,(t_\ell) &= 0, \qquad k = r+1,\dots,2r,\ \ell = m, n+1. & (4.71) \end{aligned}$$

Definition 4.58 *The spline satisfying* (4.70) *and* (4.71) *is called the* ***natural interpolating spline*** *or simple* ***natural spline*** *of degree* m *for the data* $\mathbf{y}_j$ *and the knots* T.

Proof: Due to the Schoenberg–Whitney theorem 4.56, the spline interpolation with[87]

$$x_j = t_{j+r+1} \in (t_j, t_{j+m+1}), \qquad j = 0,\dots,n,$$

[86]There, is a still popular misunderstanding that splines would only exist of odd order and that even order splines are useless. This is *not* true, of course, it is only the notion of the natural spline that would fail and, as we will see soon, the natural spline is only natural for $m = 3$ and not even then.

[87]Now we include some of the boundary knots as well. This is possible as we requested *all* knots to be simple.

is uniquely solvable. By $\mathbf{s}_{-r}, \dots, \mathbf{s}_r$ we now denote the solutions of the interpolation problem based on (4.70) and the additional conditions

$$\mathbf{s}_j(x_k) = \begin{cases} 1 & j < 0 \text{ und } k = r - j, \\ 1 & j > 0 \text{ und } k = n + j - r, \\ 0 & \text{sonst}, \end{cases} \qquad k = 0, \dots, r-1, n+1, \dots, n+r.$$

All these splines interpolate at $t_m, \dots, t_{n+1}$ and only differ at the "additional points" $t_{r+1}, \dots, t_{m-1}$ und $t_{n+2}, \dots, t_{n+r+1}$: $\mathbf{s}_0$ vanishes on all of them while $\mathbf{s}_{-1}, \dots, \mathbf{s}_{-r}$ take the value 1 at one of the additional points to the left, $\mathbf{s}_1, \dots, \mathbf{s}_r$ at one to the right, see Fig. 4.12.

These $m = 2r + 1$ splines are nonzero and linearly independent. If we consider the linear system for $a_{-r}, \dots, a_r$, given by

$$\sum_{j=-r}^{r} a_j \, \mathbf{s}_j^{(k)}(t_\ell) = 0, \qquad k = r+1, \dots, 2r, \; \ell = m, n+1,$$

then these are $2r$ *homogeneous equations* in the $2r+1$ unknowns $a_{-r}, \dots, a_r$, hence there always exists a nontrivial solution $a_{-r}^*, \dots, a_r^*$ with either $a_{-r}^* + \cdots + a_r^* = 0$ or, after normalization, $a_{-r}^* + \cdots + a_r^* = 1$. Setting

$$\mathbf{s} = \sum_{j=-r}^{r} a_j^* \, \mathbf{s}_j,$$

and taking into account that for $j = 0, \dots, n - m + 1$ we have

$$\mathbf{s}(t_{m+j}) = \sum_{k=-r}^{r} a_k^* \underbrace{\mathbf{s}_k(t_{j+m})}_{=\mathbf{y}_j} = \mathbf{y}_j \sum_{k=-r}^{r} a_k^* = \begin{cases} \mathbf{y}_j, & a_{-r}^* + \cdots + a_r^* = 1, \\ 0, & a_{-r}^* + \cdots + a_r^* = 1, \end{cases}$$

if follows that $\mathbf{s}$ is either a valid solution of (4.70) or a *nonzero* solution of the respective homogeneous problem[88], depending on whether $\sum a_j^*$ has the value 1 or 0. It still remains to prove that the second case is usually impossible which will also verify the uniqueness claim. □

4.3.5 Minimality and origin of the name

Now, we can finally describe the "valuable" minimal property of spline interpolants.

Definition 4.59 (Energy norm) *For* $k \in \mathbb{N}$ *and* $I \subset \mathbb{R}$ *we define*[89] *the* ***energy norm*** $|\cdot|_k$ *as the* ***seminorm***[90]

$$|f|_k = |f|_{k,I} = \left(\int_I \left| f^{(k)}(x) \right|^2 \, dx \right)^{1/2}, \qquad f \in C^{(k)}(I). \tag{4.72}$$

[88] The case $y_j = 0$.

[89] Strictly speaking, we must choose I such that we can do integration there. With the (fairly cheap) **Riemann integral** one usually encounters in Analysis courses, I would be a reasonable union of finite intervals, for integration over all of $\mathbb{R}$ it would be better to introduce a **Lebesgue integral**. This is all very interesting, but for our purposes here the subtle differences are not relevant.

[90] A seminorm has almost the same definition as a norm with the difference that $\|x\| = 0$ does not imply $x = 0$ any more.

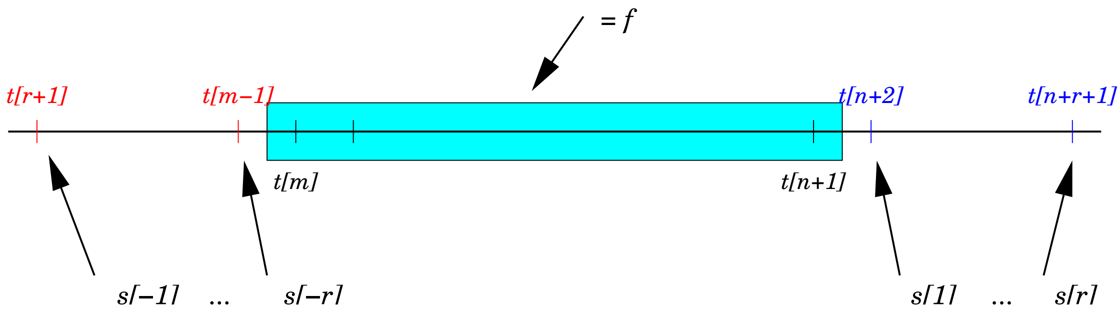

Figure 4.12: The extended interpolation problem from the proof of Theorem 4.57. On the marked region all splines take the prescibed values, at the additional points they behave in a 0/1 way except $\mathbf{s}_0$ which vanishes in all of them.

Exercise 4.11 Which functions satisfy $|f|_k = 0$? ◊

Exercise 4.12 Show that for a compact interval I, $\|f\| := \max_{x\in I} |f(x)|$ and $k \geq 1$ the following expressions are norms

1. $\|f\| + |f|_k$,
2. $\max\{\|f\|, |f_k|\}$
3. $\|f\| + \sum_{j=1}^{k} 2^{-k} |f|_k$.

◊

Remark 4.60 (Energy norms) *In the special case $k = 2$ the energy norm is the integral over the second derivative which can be considered an approximation for the curvature and therefore the integral is an approximation for the* ***bending energy*** *of the curve.*

Theorem 4.61 (Minimality) *Let $m = 2r + 1 \in \mathbb{N}$ and suppose that T consists of simple knots. For $f \in C^{r+1}(I)$ let $S_f = N_{m,T}\,\mathbf{d}$ be a spline that satisfies (4.70) and (4.71) for $\mathbf{y}_j = f(t_j)$. Then*

$$|S_f|_{r+1,I} \leq |f|_{r+1,I} \tag{4.73}$$

Proof: We start with

$$\begin{aligned} |f - S_f|^2_{r+1,I} &= \int_I \left(f^{(r+1)}(x) - (S_f)^{(r+1)}(x)\right)^2 dx \\ &= \int_I \left(f^{(r+1)}(x)\right)^2 - 2f^{(r+1)}(x)(S_f)^{(r+1)}(x) + \left((S_f)^{(r+1)}(x)\right)^2 dx \\ &= |f|^2_{r+1,I} - 2\int_I \left(f^{(r+1)}(x) - (S_f)^{(r+1)}(x)\right)(S_f)^{(r+1)}(x)\, dx - |S_f|^2_{r+1,I}. \end{aligned} \tag{4.74}$$

Figure 4.13: A real world spline, consisting of a bendable ruler and weights, called **ducks** that tie it to certain places where the curve has to interpolate. The spline was given to the Numeric Mathematics group of Giessen by H. Hollenhorst.

For $j = m, \dots, n$ we next use **partial integration** to show that

$$
\begin{aligned}
&\int_{t_j}^{t_{j+1}} \left(f^{(r+1)}(x) - S_f^{(r+1)}(x) \right) S_f^{(r+1)}(x)\, dx \\
&= \left(f^{(r)}(x) - S_f^{(r)}(x) \right) S_f^{(r+1)}(x) \Big|_{t_j}^{t_{j+1}} - \int_{t_j}^{t_{j+1}} \left(f^{(r)}(x) - S_f^{(r)}(x) \right) S_f^{(r+2)}(x)\, dx \\
&= \sum_{l=0}^{k} (-1)^{r-l} \left(f^{(r-l)}(x) - S_f^{(r-l)}(x) \right) S_f^{(r+l+1)}(x) \Big|_{t_j}^{t_{j+1}} \\
&\qquad + (-1)^{k+1} \int_{t_j}^{t_{j+1}} \left(f^{(r-k)}(x) - S_f^{(r-k)}(x) \right) \underbrace{S_f^{(r+k+2)}(x)}_{=0 \text{ für } k=r}\, dx, \qquad k = 1, \dots, r \\
&= \sum_{l=0}^{r} (-1)^{r-l} \left(f^{(r-l)}(x) - S_f^{(r-l)}(x) \right) S_f^{(r+l+1)}(x) \Big|_{t_j}^{t_{j+1}},
\end{aligned}
$$

and summation of these expression over $j = m, \dots, n$ yields

$$
\begin{aligned}
&\int_I \left(f^{(r+1)}(x) - S_f^{(r+1)}(x) \right) S_f^{(r+1)}(x)\, dx \\
&\quad = \sum_{\ell=0}^{r} (-1)^{r-l} \left(f^{(r-l)}(x) - S_f^{(r-l)}(x) \right) S_f^{(r+\ell+1)}(x) \Bigg|_{t_m}^{t_{n+1}} = 0.
\end{aligned}
$$

Note that the term for $\ell = r$ vanishes since S_f interpolates f at t_m, t_{n+1} while the other terms for $l = 0, \dots, r-1$ are zero due to (4.71). Substituting this into (4.74) we eventually obtain.

$$|S_f|^2_{r+1,I} = |f|^2_{r+1,I} - |f - S_f|^2_{r+1,I} \leq |f|^2_{r+1,I} \,.$$

Here equality holds if and only if $f - S_f \in \Pi_r$.

This also verfies the uniqueness of the natural spline interpolant for $n \geq m+r$: if f is any other solution of the minimization problem, then we must have $f - S_f \in \Pi_r$ and this polynomial has to vanish at at least $r+1$ points, namely the knots where S_f interpolates. This is only possible for the zero polynomial, hence $f = S_f$. □

Proof of Theorem 4.57, continued: Now suppose that $n \geq m + r$ and that the second case in the preceding step of the proof has occured, which means that there exists a nonzero spline

$$\mathbf{s} = \sum_{j=-r}^{r} a_j^* \, \mathbf{s}_j$$

which solves the homogeneous system. Therefore, we can apply Theorem 4.61 with $f = 0$ as well as[91] $S_f = \mathbf{s}$ and obtain that

$$0 = |f|_{r+1,I} \geq |S_f|^2_{r+1,I} = \int_I |\mathbf{s}^{(r+1)}(x)|^2 \, dx$$

which yields that $\mathbf{s}^{(r+1)} = 0$, hence $\mathbf{s} \in \Pi_r$. But $n \geq m + r$ means that this polynomial must vanish at the at least $r+2$ points t_{m+j}, $j = 0, \dots, n-m+1$, hence is zero which gives a contradiction. This has two consequences:

1. The value $a^*_{-r} + \cdots a^*_r$ must be $\neq 0$ and therefore can be normalized to 1. Hence, the natural spline exists.
2. The natural spline interpolant must be unique because the difference of two natural interpolants is another nonzero solution of the homogeneous problem.

□

Remark 4.62 *This proof almost looks like a cyclical argument: First we claim that there exists a natural spline, then we show its minimality and then we use the minimality to prove its existence. Showing properties without existence is a dangerous thing to do as for elements of the empty set* any *property holds true.*

Nevertheless the argument is correct which we can see by summarizing the steps:

1. *There either exists a natural spline interpolant or, if not, there exists a nonzero solution of the homogeneous problem.*
2. *Any spline interpolant is minimizing the energy norm.*
3. *If there would exist a nonzero homogeneous solution, then it has to be zero, an obvious contradiction.*

[91] Now with a *scalar* $\mathbf{s}$.

4. *Consequently, there must be a natural spline interpolant and it has to be unique.*

Remark 4.63 *In (Boor, 1990), Carl de Boor makes the following statement*

> Die Extremaleigenschaft des interpolierenden Splines wird häufig für die große praktische Nützlichkeit der Splines verantwortlich gemacht. Dies ist jedoch glatter "Volksbetrug" ....

There is nothing to add to this.

4.3.6 The Marsden identity and knot insertion

Any polynomial is a piecewise polynomial and therefore any polynomial is a spline, even regardless of the underlying knot sequence. Thus, for every **polynomial curve p** there must be coefficients $\mathbf{p}_0, \dots, \mathbf{p}_n$ such that

$$\mathbf{p}(x) = \sum_{j=0}^{n} \mathbf{p}_j \, N_j^m(x|T), \qquad x \in [t_m, t_{n+1}], \; \mathbf{p} \in \Pi_m. \tag{4.75}$$

Our goal is to give a formula for these coefficients. To that end, let us recall the intervals $I_k = [t_k, t_{k+1})$, $k = 0, \dots, n+m$, and let us now write the polynomial pieces of the B–splines *explicitly* as

$$p_{j,k}^m = N_j^m(\cdot|T)\Big|_{I_k} \in \Pi_m, \qquad j = 0, \dots, n, \; k = 0, \dots, n+m. \tag{4.76}$$

Here, we restrict ourselves on scalar valued splines for the proof which is no restriction since we can always act on the components of the curves separately. The polynomials from (4.76) are polynomials and therefore have a **polar form** or **blossom** which we denote by $P_{j,k}^m$, resprectively. In the case of (4.75), we just have one polynomial p and therefore one polar form P.

Theorem 4.64 (Spline Duality / Marsden identity) *Let $N_{m,T}\mathbf{d}$ be a spline curve for a knot sequence T and let $\mathbf{p}_k = N_{m,T}\mathbf{d}|_{I_k} \in \Pi_m^d$ be its polynomial pieces with polar forms $\mathbf{P}_k$, $k = m, \dots, n$. Then we have the duality relation*

$$\mathbf{d}_k = \mathbf{P}_j(t_{k+1}, \dots, t_{k+m}), \qquad k = j-m, \dots, j, \quad j = m, \dots, n. \tag{4.77}$$

In particular, if the spline curve is a polynomial curve,

$$\mathbf{d}_k = \mathbf{P}(t_{k+1}, \dots, t_{k+m}), \qquad k = 0, \dots, n. \tag{4.78}$$

The following technical lemma is just based on straightforward computations but nevertheless is the main ingredient for the proof.

Lemma 4.65 *The polar forms $P_{j,k}^m$ for $p_{j,k}^m$, $j = 0, \dots, n$, $k = 0, \dots, n+m$, are determined recursively as*

$$\begin{aligned} P_{j,k}^0() &= \delta_{jk}, & (4.79)\\ P_{j,k}^\ell(x_1, \dots, x_\ell) &= u_1\left(x_\ell | I_j^\ell\right) P_{j,k}^{\ell-1}(x_1, \dots, x_{\ell-1}) \\ &\quad + u_0\left(x_\ell | I_{j+1}^\ell\right) P_{j+1,k}^{\ell-1}(x_1, \dots, x_{\ell-1}), \qquad \ell = 1, \dots, m. & (4.80) \end{aligned}$$

Proof: We first remark that for $x_1 = \cdots = x_m = x$ the recurrence (4.79) und (4.80) is precisely the B–spline recursion (4.49), hence $P_{j,k}^m(x^m) = p_{j,k}^m(x)$. That $P_{j,k}^m$ is a **multiaffine form** follows from the fact that the barycentric coordinates are affine functions. What remains is symmetry. To prove that, we apply (4.80) twice and get

$$\begin{aligned}
P_{j,k}^\ell(x_1,\dots,x_\ell) &= u_1(x_\ell|I_j^\ell)\, P_{j,k}^{\ell-1}(x_1,\dots,x_{\ell-1}) + u_0(x_\ell|I_{j+1}^\ell)\, P_{j+1,k}^{\ell-1}(x_1,\dots,x_{l-1}) \\
&= u_1(x_\ell|I_j^\ell)\, u_1(x_{\ell-1}|I_j^{\ell-1})\, P_{j,k}^{\ell-2}(x_1,\dots,x_{\ell-2}) \\
&\quad + u_1(x_\ell|I_j^\ell)\, u_0(x_{\ell-1}|\Delta_{j+1}^{\ell-1})\, P_{j+1,k}^{\ell-2}(x_1,\dots,x_{\ell-2}) \\
&\quad + u_0(x_\ell|I_{j+1}^\ell)\, u_1(x_{\ell-1}|I_{j+1}^{\ell-1})\, P_{j+1,k}^{l-2}(x_1,\dots,x_{\ell-2}) \\
&\quad + u_0(x_\ell|I_{j+1}^\ell)\, u_0(x_{\ell-1}|I_{j+2}^{\ell-1})\, P_{j+2,k}^{\ell-2}(x_1,\dots,x_{\ell-2}).
\end{aligned}$$

Since

$$u_1(x_\ell|I_j^\ell)\, u_1(x_{\ell-1}|I_j^{\ell-1}) = \frac{(x_\ell - t_j)\,(x_{\ell-1} - t_j)}{(t_{j+\ell} - t_j)\,(t_{j+\ell-1} - t_j)},$$

and

$$u_0(x_\ell|I_{j+1}^\ell)\, u_0(x_{\ell-1}|I_{j+2}^{\ell-1}) = \frac{(t_{j+\ell+1} - x_\ell)\,(t_{j+\ell+1} - x_{\ell-1})}{(t_{j+\ell+1} - t_{j+1})\,(t_{j+\ell+1} - t_{j+2})},$$

as well as

$$\begin{aligned}
& u_1(x_\ell|I_j^\ell)\, u_0(x_{\ell-1}|I_{j+1}^{\ell-1}) + u_0(x_\ell|I_{j+1}^\ell) u_1(x_{\ell-1}|I_{j+1}^{\ell-1}) \\
&= \frac{(x_\ell - t_j)\,(t_{j+\ell} - x_{\ell-1})}{(t_{j+\ell} - t_j)\,(t_{j+\ell} - t_{j+1})} + \frac{(t_{j+\ell+1} - x_\ell)\,(x_{\ell-1} - t_{j+1})}{(t_{j+\ell+1} - t_{j+1})\,(t_{j+\ell} - t_{j+1})} \\
&= \frac{(t_{j+\ell+1} - t_{j+1})\,(x_\ell - t_j)\,(t_{j+\ell} - x_{\ell-1}) + (t_{j+\ell} - t_j)\,(t_{j+\ell+1} - x_\ell)\,(x_{\ell-1} - t_{j+1})}{(t_{j+\ell} - t_j)\,(t_{j+\ell} - t_{j+1})\,(t_{j+l+1} - t_{j+1})} \\
&= \frac{t_{j+\ell}\,(t_{j+\ell+1} - t_{j+1}) + t_{j+1}\,(t_{j+\ell} - t_j)}{(t_{j+\ell} - t_j)\,(t_{j+\ell} - t_{j+1})\,(t_{j+\ell+1} - t_{j+1})}\, x_\ell \\
&\qquad + \frac{t_j\,(t_{j+\ell+1} - t_{j+1}) + t_{j+\ell+1}\,(t_{j+\ell} - t_j)}{(t_{j+\ell} - t_j)\,(t_{j+\ell} - t_{j+1})\,(t_{j+\ell+1} - t_{j+1})}\, x_{\ell-1} \\
&\qquad - \frac{t_{j+\ell+1} - t_{j+1} + t_{j+\ell} - t_j}{(t_{j+\ell} - t_j)\,(t_{j+\ell} - t_{j+1})\,(t_{j+\ell+1} - t_{j+1})}\, x_\ell x_{\ell-1} \\
&\qquad - \frac{(t_{j+\ell+1} - t_{j+1})\, t_j t_{j+\ell} + (t_{j+\ell} - t_j)\, t_{j+\ell+1} t_{j+1}}{(t_{j+\ell} - t_j)\,(t_{j+\ell} - t_{j+1})\,(t_{j+\ell+1} - t_{j+1})} \\
&= \frac{(t_{j+\ell+1} t_{j+\ell} - t_{j+1} t_j)\,(x_\ell + x_{\ell-1}) - (t_{j+\ell+1} - t_{j+1} + t_{j+\ell} - t_j)\ \ x_\ell x_{\ell-1}}{(t_{j+\ell} - t_j)\,(t_{j+\ell} - t_{j+1})\,(t_{j+\ell+1} - t_{j+1})} \\
&\qquad - \frac{(t_{j+\ell+1} - t_{j+1})\, t_j t_{j+\ell} + (t_{j+\ell} - t_j)\, t_{j+\ell+1} t_{j+1}}{(t_{j+\ell} - t_j)\,(t_{j+\ell} - t_{j+1})\,(t_{j+\ell+1} - t_{j+1})}
\end{aligned}$$

are symmetric expression with respect to x_ℓ and $x_{\ell-1}$, so is $P_{j,k}^l(x_1,\dots,x_\ell)$. The rest of the interchanging argument is like in the proof of Proposition 4.26. $\square$

Equipped with Lemma 4.65, we can derive the following crucial formula.

Lemma 4.66 *For $\ell = 0, \dots, m$, $j = 0, \dots, n$ and $k = \ell, \dots, n$ we have*

$$P_{j,k}^{\ell}(t_{r+1}, \dots, t_{r+\ell}) = \delta_{jr}, \qquad r = k-\ell, \dots, k. \tag{4.81}$$

Remark 4.67 *Equation* (4.81) *is what one calls a **duality** in mathematics.*

Proof of Lemma 4.66: Induction on $\ell = 0, \dots, m$, where $\ell = 0$ is just (4.79). For $\ell > 0$ we first choose some $r > k - \ell$ which then also satisfies $r \ge k - \ell + 1 = k - (\ell - 1)$. By (4.80) and the induction hypothesis we then get

$$\begin{aligned} & P_{j,k}^{\ell}(t_{r+1}, \dots, t_{r+\ell}) \\ = \; & u_1(t_{r+\ell}|I_j^\ell)\, P_{j,k}^{\ell-1}(t_{r+1}, \dots, t_{r+\ell-1}) + u_0(t_{r+\ell}|I_{j+1}^\ell) P_{j+1,k}^{\ell-1}(t_{r+1}, \dots, t_{r+\ell-1}) \\ = \; & u_1(t_{r+\ell}|I_j^\ell)\, \delta_{jr} + u_0(t_{r+\ell}|I_{j+1}^\ell)\, \delta_{j+1,r} = \underbrace{u_1(t_{r+\ell}|I_r^\ell)}_{=1} \delta_{jr} + \underbrace{u_0(t_{r+\ell}|I_r^\ell)}_{=0} \delta_{j+1,r} \\ = \; & \delta_{jr}. \end{aligned}$$

For $r = k - \ell$ we make use of the symmetry of the polar form[92] to get

$$\begin{aligned} & P_{j,k}^{\ell}(t_{r+1}, \dots, t_{r+\ell}) = P_{j,k}^{\ell}(t_{r+2}, \dots, t_{r+\ell}, t_{r+1}) \\ = \; & u_1(t_{r+1}|I_j^\ell)\, P_{j,k}^{\ell-1}(t_{r+2}, \dots, t_{r+\ell}) + u_0(t_{r+1}|I_{j+1}^\ell)\, P_{j+1,k}^{\ell-1}(t_{r+2}, \dots, t_{r+\ell}) \\ = \; & u_1(t_{r+1}|I_j^\ell)\, \delta_{j,r+1} + u_0(t_{r+1}|I_{j+1}^\ell)\, \delta_{j+1,r+1} \\ = \; & \underbrace{u_1(t_{r+1}|I_{r+1}^\ell)}_{=0} \delta_{j,r+1} + \underbrace{u_0(t_{r+1}|I_{r+1}^\ell)}_{=1} \delta_{jr} = \delta_{jr}. \end{aligned}$$

□

Proof of Theorem 4.64: If I_k is a nontrivial interval then (4.45) yields

$$\mathbf{p}_k(x) = N_{m,T}\mathbf{d}(x) = \sum_{j=k-m}^{k} \mathbf{d}_j N_j^m(x|T) = \sum_{j=k-m}^{k} \mathbf{d}_j p_{j,k}^m(x), \qquad x \in I_k.$$

Taking the polar forms of both sides yileds

$$\mathbf{P}_k(x_1, \dots, x_m) = \sum_{j=k-m}^{k} \mathbf{d}_j P_{j,k}^m(x_1, \dots, x_m), \qquad x_1, \dots, x_m \in \mathbb{R}, \tag{4.82}$$

and substituting $x_\ell = t_{j+\ell}$, $\ell = 1, \dots, m$, in (4.82) for some $j \in \{k-m, \dots, k\}$, the identity (4.81) yields

$$\mathbf{P}_k(t_{j+1}, \dots, t_{j+m}) = \sum_{j=k-m}^{k} \mathbf{d}_j \underbrace{P_{j,k}^m(t_{j+1}, \dots, t_{j+m})}_{=\delta_{jk}} = \mathbf{d}_k. \tag{4.83}$$

(4.78) is then a direct consequence of (4.77). □

With the help of Theorem 4.64 we can also prove very easily a fundamental procedure in the manipulation of splines.

[92]Somewhere we have to apply Lemma 4.65 into which we invested so much effort.

Definition 4.68 *A knot sequence* $T' = T'_{m,n'} = (t'_0, \dots, t'_{m+n'+1})$ *is called a* ***refinement*** *of a knot sequence* T *if there exists a strictly monotonic mapping* $\tau : \{1, \dots, m + n\} \to \{1, \dots, m + n'\}$ *such that*

$$t_j = t'_{\tau(j)}, \qquad j = 0, \dots, n + m. \tag{4.84}$$

We write this as $T \subseteq T'$.

Remark 4.69 *Since* (4.84) *says that every knot in* T *can be found within* T' *with* at least *the same multiplicity, the notation* $T \subseteq T'$ *is justified.*

Since (4.84) implies that for any $k \in \{1, \dots, m+n'\}$ there exists a $j \in \{1, \dots, m+n\}$ such that

$$[t'_k, t'_{k+1}] \subseteq [t_j, t_{j+1}],$$

any piecewise polynomial on T is also a piecewise polynomial on T with *infinite* differentiability at the additional knots. Therefore we have proved the following simple but fundamental observation.

Theorem 4.70 (Nested spline spaces) *If* $T \subseteq T'$ *then also* $\mathbb{S}_m(T) \subseteq \mathbb{S}_n(T')$.

Consequently, any spline curve $\mathbf{s} = N_{m,T}\mathbf{d} \in \mathbb{S}_m(T)$ also belongs to $\mathbb{S}_m(T')$ and therefore can also be written as $N_{m,T'}\mathbf{d} \in \mathbb{S}_m(T')$, hence there exist coefficients $\mathbf{d}' \in \mathbb{R}^{d \times n'}$ such that

$$\sum_{j=0}^{n} \mathbf{d}_j \, N_j^m(\cdot|T) = N_{m,T}\mathbf{d} = N_{m,T}\mathbf{d}' = \sum_{j=0}^{n'} \mathbf{d}'_j \, N_j^m(\cdot|T'),$$

and the obvious question is: How can we compute $\mathbf{d}'$ from $\mathbf{d}$? This procedure is called **knot insertion**. We will consider the insertion of a single knot here, i.e., either we add a new point or we raise the multiplicity of an already existing knot[93] by one.

To that end, suppose that for some $j \le n + m$ we have

$$t_0 \le \cdots \le t_j \le t' \le t_{j+1} \le \cdots \le t_{n+m+1}$$

and set

$$T' = \{t'_k : k = 0, \dots, n + m + 2\}, \qquad t'_k = \begin{cases} t_k & k = 0, \dots, j, \\ t' & k = j + 1, \\ t_{k-1} & k = j + 2, \dots, n + m + 2. \end{cases} \tag{4.85}$$

Then we have the following algorithm to compute the new control points. The algorithm is usually attributed to Boehm[94], but there is also the so–called **Oslo algorithm** due to a group at Oslo University, see (Lyche, 1987). It obviously does the same in the case of simple knot insertion but is also capable of inserting several knots at the same time.

[93]Whose multiplicity should be $\le m$, of course, as multiplicity $> m + 1$ is still forbidden.

[94]His original name was *Böhm* but since "ö" is not so common in English, he changed it to be more readable.

Theorem 4.71 (Knot insertion) *If* T' *is given as in* (4.85), *then the new control points* $\mathbf{d}'$ *can be computed as*

$$\mathbf{d}'_k = \begin{cases} \mathbf{d}_k & k = 0, \dots, j-m, \\ u_0(t'|I_k^m)\, \mathbf{d}_{k-1} + u_1(t'|I_k^m)\, \mathbf{d}_k & k = j-m+1, \dots, j, \\ \mathbf{d}_{k-1} & k = j+1, \dots, n+1. \end{cases} \tag{4.86}$$

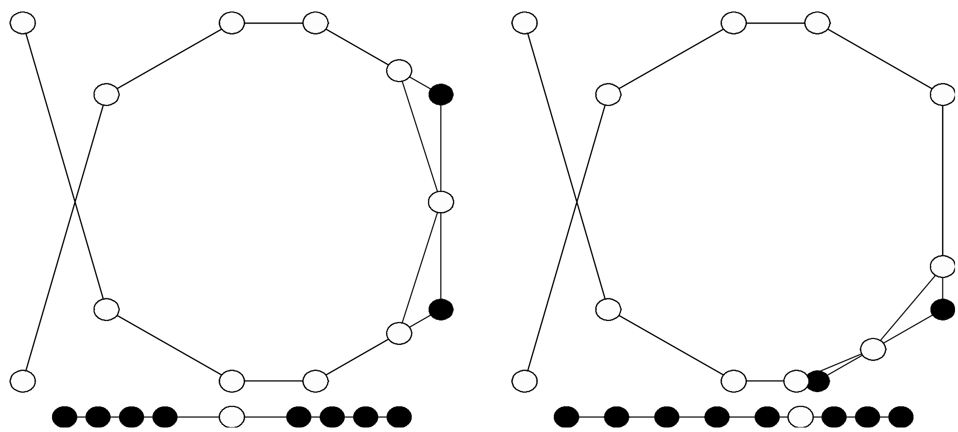

Figure 4.14: Two examples for knot insertion.

Proof: We again use (4.77), more precisely, the identity

$$\mathbf{d}^*_k = \mathbf{P}'_k\left(t^*_{k+1}, \dots, t^*_{k+m}\right), \qquad k = 0, \dots, n+1. \tag{4.87}$$

Since for $k = 0, \dots, j-m$ we have[95] $I'_k = I_k$, and therefore $\mathbf{P}'_k = \mathbf{P}_k$ as well as $\left(t'_{k+1}, \dots, t'_{k+m}\right) = (t_{k+1}, \dots, t_{k+m})$, it follows that

$$\mathbf{d}'_k = \mathbf{P}'_k\left(t'_{k+1}, \dots, t'_{k+m}\right) = \mathbf{P}_k\left(t_{k+1}, \dots, t_{k+m}\right) = \mathbf{d}_k, \qquad k = 0, \dots, j-m.$$

Whenever $k \ge j+1$, hence $I'_k = I_{k-1}$, $\mathbf{P}'_k = \mathbf{P}_{k-1}$ and $\left(t'_{k+1}, \dots, t'_{k+m}\right) = (t_k, \dots, t_{k+m-1})$, an analogous reasoning yields

$$\mathbf{d}'_k = \mathbf{P}'_k\left(t'_{k+1}, \dots, t'_{k+m}\right) = \mathbf{P}_{k-1}\left(t_k, \dots, t_{k+m-1}\right) = \mathbf{d}_{k-1}, \qquad k = j+1, \dots, n+1.$$

The interestign cases are of course the ones in which t' appears. There, we write t' as a barycentric combination of the knots t_k and t_{k+m}, that is

$$t' = u_0(t'|I_k^m)\, t_k + u_1(t'|I_k^m)\, t_{k+m}. \tag{4.88}$$

[95] With the obvious extension of notation.

Then

$$\begin{aligned}
\mathbf{d}'_k &= \mathbf{P}_k\left(t'_{k+1},\dots,t'_{k+m}\right) = \mathbf{P}_k\left(t_{k+1},\dots,t_j,t',t_{j+1},\dots,t_{k+m-1}\right)\\
&= u_0(t'|I_k^m)\,\mathbf{P}_k\left(t_{k+1},\dots,t_j,t_k,t_{j+1},\dots,t_{k+m-1}\right)\\
&\quad +u_1(t'|\Delta_k^m)\,\mathbf{P}_k\left(t_{k+1},\dots,t_j,t_{k+m},t_{j+1},\dots,t_{k+m-1}\right)\\
&= u_0(t'|I_k^m)\,\underbrace{\mathbf{P}_k\left(t_k,\dots,t_{k+m-1}\right)}_{=\mathbf{d}_{k-1}} + u_1(t'|I_k^m)\,\underbrace{\mathbf{P}_k\left(t_{k+1},\dots,t_{k+m}\right)}_{=\mathbf{d}_k}\\
&= u_0(t'|I_k^m)\,\mathbf{d}_{k-1} + u_1(t^*|I_k^m)\mathbf{d}_k,
\end{aligned}$$

which is (4.86). To make sure that we made no hidden mistakes in (4.88), we note that

$$\bigcap_{k=j-m+1}^{j} I_k^m = [t_{j-m+1},t_{j+1}] \cap \dots \cap [t_j,t_{j+m}] = [t_j,t_{j+1}] \ni t',$$

so that (4.88) is always well–defined, even when $t_j = t_{j+1}$ is knot of higher multiplicity. □

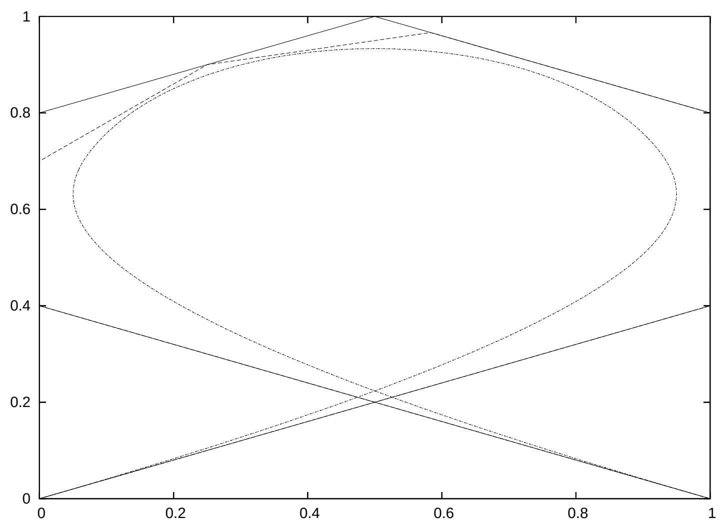

Figure 4.15: Insertion of knots and resulting spline curves, plotted twice. Surprisingly enough, the curves are the same.

Remark 4.72 (Knot insertion and multiple knots)

1. *The rule* (4.86) *can also be used to raise the* ***multiplicity*** *of knot, neither the formula nor the proof changes.*

2. *If a knot* t' *is inserted* m *times or raised to multiplicity* m*, then one of the control points has the value* $N_{m,T'}\mathbf{d}(t')$ *since spline curves interpolate a control point at knots of multiplicity* m*. Tracing the recurrence for this control point, we the obtain de Boor algorithm again.*

3. *Hence, the de Boor algorithm can be interpreted as knot insertion or could be factorized into knot insertion steps.*

Knot insertion can also be conveniently written in a different way by considering the matrices

$$\mathbf{d} = [\mathbf{d}_0 \ldots \mathbf{d}_n] \qquad \text{and} \qquad \mathbf{d}' = [\mathbf{d}'_0 \ldots \mathbf{d}'_n] \,.$$

Then $\mathbf{d}' = \mathbf{d}\mathbf{V}$ with the values

$$\alpha_k = u_0(t'|I_k^m), \qquad k = j-m+1, \ldots, j \tag{4.89}$$

and the resulting matrix

$$\mathbf{A} = \mathbf{A}_T\,(t') = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & \alpha_{j-m+1} & & & & \\ & & & 1-\alpha_{j-m+1} & \ddots & & & \\ & & & & \ddots & \alpha_j & & \\ & & & & & 1-\alpha_j & 1 & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix} \in \mathbb{R}^{n \times n+1}. \tag{4.90}$$

This matrix can be used to describe the relationship between the spline spaces $\mathbb{S}_m(T)$ and $\mathbb{S}_m\,(T')$ and plays an important role for singularity detection in spline curves, see (Hamm *et al.*, 2014).

Knot insertion can be applied to make splines comparable. Suppose that T and T' are two arbitrary knot sequences of degree[96] m and $\mathbf{d} \in \mathbb{R}^{d \times n}$ and $\mathbf{d} \in \mathbb{R}^{d \times n'}$ are control points for these knot sequences. If want to do operations on the two splines $N_{m,T}\mathbf{d}$ and $N_{m,T'}\mathbf{d}'$, for example addition or subtraction or general distance computations, it is more practical to use splines with identical knot sequences. To that end, let

$$T_\cup := T \cup T' = \min\{T^* : T \subseteq T^*, T' \subset T^*\}$$

the smallest knot sequence which contains both T and T'. Then

$$N_{m,T}\mathbf{d} = N_{m,T_\cup}\mathbf{d}\mathbf{A}_T(T_\cup \setminus T) \qquad \text{and} \qquad N_{m,T'}\mathbf{d}' = N_{m,T_\cup}\mathbf{d}'\mathbf{A}_{T'}(T_\cup \setminus T')$$

[96] Essentially this means only the restriction on the multiplicity of knots.

are spline curves based on the same knot sequence and therefore can be added subtracted or compared. In particular,

$$
\begin{aligned}
&\left\|(N_{m,T}\mathbf{d} - N_{m,T'}\mathbf{d}')(x)\right\| \\
&= \left\|(N_{m,T_\cup}\mathbf{d}\mathbf{A}_T(T_\cup \setminus T) - N_{m,T_\cup}\mathbf{d}'\mathbf{A}_{T'}(T_\cup \setminus T'))(x)\right\| \\
&= \left\|\sum_{j=0}^{n_\cup} (\mathbf{d}\mathbf{A}_T(T_\cup \setminus T) - \mathbf{d}'\mathbf{A}_T(T_\cup \setminus T'))_j \; N_j^m(x|T_\cup)\right\| \\
&\le \sum_{j=0}^{n_\cup} \left\|(\mathbf{d}\mathbf{A}_T(T_\cup \setminus T) - \mathbf{d}'\mathbf{A}_T(T_\cup \setminus T'))_j\right\| N_j^m(x|T_\cup) \\
&\le \max_j \left\|(\mathbf{d}\mathbf{A}_T(T_\cup \setminus T) - \mathbf{d}'\mathbf{A}_T(T_\cup \setminus T'))_j\right\| \underbrace{\sum_{j=0}^{n_\cup} N_j^m(x|T_\cup)}_{=1} \\
&= \max_j \left\|(\mathbf{d}\mathbf{A}_T(T_\cup \setminus T) - \mathbf{d}'\mathbf{A}_T(T_\cup \setminus T'))_j\right\|
\end{aligned}
$$

gives a first and simplest version of a distance estimate between spline curves.

> *Die hitzigsten Verteidiger eine Wissenschaft, die nicht den geringsten scheelen Seitenblick auf dieselbe vertragen können, sind gemeiniglich solche Personen, die es nicht sehr weit in derselben gebracht haben und sich dieses Mangels heimlich bewußt sind.*
>
> Lichtenberg

5 Geometric objects II: Surfaces in CAD

Curves are a nice thing, but normally the world around us is three dimensional which makes surfaces the things to go. Here we will mostly consider two dimensional, "classical" surfaces. Of particular interest will be methods which construct surfaces from curves since we now understand curves quite well already.

As curves only make sense in $\mathbb{R}^d$, $d \geq 2$, as geometric objects we need at least $d \geq 3$ when dealing with parametric surfaces.

5.1 Planes and derived objects

The simplest geometric object of the surface type is a plane. Since a two dimensional affine plane is a hyperplane at the same time there are different ways to define it.

Definition 5.1 (Planes and hyperplanes)

1. *A **plane** in $\mathbb{R}$ is a two dimensional **affine subspace** of the form*

$$\mathbf{x} + \mathbf{X}\mathbb{R}^2, \qquad \mathbf{x} \in \mathbb{R}^d, \ \mathbf{X} \in \mathbb{R}^{d\times 2}. \tag{5.1}$$

*The plane is called **nondegenerate** if the rank of $\mathbf{X}$ ist 2.*

2. *A **hyperplane** in $\mathbb{R}^d$ is an affine subspace of **codimension** 1, i.e., the solution set of a linear equation:*

$$\{\mathbf{x} : \mathbf{n}^T\mathbf{x} = c\}, \qquad \mathbf{n} \in \mathbb{R}^d \setminus \{0\}, \ c \in \mathbb{R}. \tag{5.2}$$

A few comments on (5.2). Since for any two solutions $\mathbf{x}, \mathbf{x}'$ of $\mathbf{n}^T\mathbf{x} = c$ and $\alpha \in \mathbb{R}$ we have

$$\mathbf{n}^T(\alpha\mathbf{x} + (1-\alpha)\mathbf{x}') = \alpha \underbrace{\mathbf{n}^T\mathbf{x}}_{=c} + (1-\alpha) \underbrace{\mathbf{n}^T\mathbf{x}'}_{=c} = c(\alpha + 1 - \alpha) = c,$$

the set defined in (5.2) is indeed an affine subspace of $\mathbb{R}^d$. The **normal n** for the plane is not unique since for any $\alpha \neq 0$

$$(\alpha \mathbf{n})^T \mathbf{x} = \alpha \mathbf{c} \qquad \Leftrightarrow \qquad \mathbf{n}^T \mathbf{x} = c,$$

so that we request **n** to be **normalized**[97], i.e., $\|\mathbf{n}\|_2 = 1$. This defines **n** up to its sign which we can fix such that $c \geq 0$ which only leaves ambiguities in the case of a **linear subspace** where $c = 0$.

Since dimension and codimension always add up to the dimension of the **ambient space** $\mathbb{R}^d$, the valuable identity[98] $1+2 = 3$ shows that in $\mathbb{R}^3$ nondegenerate planes and hyperplanes are the same.

However, hyperplanes are *infinite* objects and therefore not realistic in CAD systems. Due to that, most planar objects are restricted by means of curves.

Definition 5.2 (Jordan curve) *A continuous function* $\mathbf{f} : [a, b] \to \mathbb{R}^2$ *is called a **Jordan curve**[99] if it is*

1. ***closed**, i.e.,* $\mathbf{f}(a) = \mathbf{f}(b)$, *and*
2. ***injective** on* $[a, b)$, *i.e.,* $\mathbf{f}(t) \neq \mathbf{f}(t')$ *whenever* $t \neq t' \in [a, b)$.

For Jordan curves we have the following result which is as intuitive as nontrivial and has fist been proved by Camille Jordan in 1887/1893, see (Jordan, 1887), though the proof is considered incomplete.

Theorem 5.3 (Jordan curve theorem) *Any Jordan curve* $\mathbf{f} : I \to \mathbb{R}^2$, $I = [a, b]$, *decomposes* $\mathbb{R}^2$ *into two open regions* G_1, G_2 *with*

$$\partial G_j = \mathbf{f}(I), \qquad \mathbb{R}^2 = G_1 \cup G_2 \cup \mathbf{f}(I). \tag{5.3}$$

*One of these regions is bounded and called the **inner region**, the other one is unbounded.*

We are not going to prove this theorem here as the effort is too much for our purposes here, but nevertheless this is the mathematical background and justification of a method intuitively used in CAD: every continuous **closed curve** encloses a bounded domain which is called the **trimmed domain**. To trim a piece from a plane, one simply maps the trimmed region in $\mathbb{R}^2$ to the plane by means of (5.1).

5.2 Extrusion and ruled surfaces

There are some extremely simple methods to generate surfaces in $\mathbb{R}^d$, $d \geq 3$, from curves and they are the most widely used ones in CAD systems. The simplest way is to use **extrusion** which means to shift the curve along a vector.

[97]It shouldn't appear so uncommon to require normals to be normalized.

[98]Exercise: prove this identity. Seriously, try to *prove* it! What do you realize? You cannot, it's a definition!

[99]Alternatively, its image is called the curve.

Definition 5.4 (Extrusion) *For a curve* $\mathbf{f} : I \to \mathbb{R}^d$ *and a* ***translation vector*** $\mathbf{t} \in \mathbb{R}^d$*, the* ***extruded surface*** *ist defined as*

$$\mathbf{F} : I \times [0,1] \to \mathbb{R}^d, \qquad \mathbf{F}(x,y) = \mathbf{f}(x) + y\,\mathbf{t}. \tag{5.4}$$

If $\mathbf{f}$ *is a* ***planar curve****, i.e.,* $\mathbf{f}(I) \subset P$ *for some hyperplane* $P \subset \mathbb{R}^d$ *with normal* $\mathbf{n}$*, then the extrusion vector is normally chosen as* $\pm\mathbf{n}$*.*

Note that that the derivative of an extrusion takes a particular simple form,

$$J\mathbf{F}(x,y) = \nabla\mathbf{F}(x,y) = [\mathbf{f}'(x), \mathbf{t}]$$

and does not depend on y.

If we prescribe two curves, we can also connect them by the same method.

Definition 5.5 (Ruled surface) *For two curves* $\mathbf{f}_1, \mathbf{f}_2 : I \to \mathbb{R}^d$ *with identical parameter region* I *we define the* ruled surface *as*

$$\mathbf{F} : I \times [0,1] \to \mathbb{R}^d, \qquad \mathbf{F}(x,y) = (1-y)\,\mathbf{f}_1(x) + y\,\mathbf{f}_2(x). \tag{5.5}$$

Remark 5.6 (Ruled surfaces)

1. *A ruled surface connects* ***equiparametric*** *points on the two curves to each other by a straight line.*
2. *If we reparametrize one of the curves, the ruled surface changes. This can be used in algorithms, but sometimes it is also advisable to use an* ***arc length parametrization*** *for both curves. This, however, requires that both curves have the same length.*
3. *Extrusion is a ruled surface with* $\mathbf{f}_1 = \mathbf{f}_2 = \mathbf{f}$*.*

The derivative of a ruled surface can be computed as

$$J\mathbf{F}(x,y) = \left[(1-y)\,\mathbf{f}_1'(x) + y\,\mathbf{f}_2'(x), \mathbf{f}_2(x) - \mathbf{f}_1(x)\right].$$

Example 5.7 (Extrusion & Ruled surface)

1. *The extrusion of a circle in a plane by means of the plane normal is a* ***cylinder****, the extrusion of a line segment gives a rectangle.*
2. *The ruled surface for two line segments is a* ***bilinear function****, i.e., a surfce of the form*
$$F(x,y) = a + bx + cy + dxy.$$
It is a plane if the four endpoints of the line segments are ***coplanar****, otherwise it is a curved surface.*
3. *The ruled surface formed by a circle and a constant curve*[100] *is a cone, the ruled surface for two polygons with the same number of vertices is a prism.*

[100] In other words, $\mathbf{f}(I) = \mathbf{y}$ consists of a single point.

5.3 Tensor products

The geometric intuition of a *parametric curve* is the idea of transforming or "bending" an interval; in the same way, a *parametric surface* can be considered to be a deformation of a two dimensional parameter region Ω, hence as a mapping $\mathbf{F} : \Omega \to \mathbb{R}^d$, $\Omega \subset \mathbb{R}^2$. The main question in that context is: What is Ω? All of a sudden we have a huge choice of different domains, just to mention circles, triangles, squares, rectangles, polygons and so on.

A very simple, nice and intuitive approach to surfaces can be found in P. Bézier's introductionary chapter in the book (Farin, 1988):

> *This idea takes us back to a very old, and sometimes forgotten, definition of a surface: it is the locus of a curve which is at the same time moved and distorted.*

Mathematically, this concept of a surface uses a one paramatric family of curves $\mathbf{f}_y : I \to \mathbb{R}^d$, $y \in J$, where we use for all values of y *the same* parameterization interval I for the curves in x.

5.3.1 Bivariate splines

Recalling the preceding chapter[101] and can write the curve $\mathbf{f}_y(x)$ as a **spline curve**[102] of order m with respect to a **knot sequence** T where neither m nor T depends on y. Formally, this means that

$$\mathbf{f}_y(x) = \sum_{j=0}^{n} \mathbf{d}_{y,j} \, N_j^m(x \,|\, T) = \sum_{j=0}^{n} \mathbf{d}_j(y) \, N_j^m(x \,|\, T), \qquad (5.6)$$

where the difference between the two ways of writing the function is a purely formal one, we just write the letter y somewhere else.

The right hand side of (5.6) contains $\mathbf{d}_j$ as a function in y and we can again write each such function as a spline curve, this time of degree m' and with respect to a knot sequence T'. Since these are only *finitely many* functions, we can always assume that it is the same T', otherwise we would just use

$$T' = \bigcup_{j=0}^{n} T'_j$$

which is generated by **knot insertion** applied to the curves $\mathbf{d}_j$ which can be written as

$$\mathbf{d}_j(y) = \sum_{k=1}^{n'} \mathbf{d}_{jk} \, N_k^{m'}(y \,|\, T'), \qquad j = 0, \dots, n. \qquad (5.7)$$

---

[101] This sometimes happens in mathematics.

[102] After all, we do not know so many types of **free form curves**.

If we now substitute (5.7) into (5.6) and replace n, m, T by n_1, m_1, T_1 as well as n', m', T' by n_2, m_2, T_2 to make the expression more symmetric, we end up with

$$\mathbf{F}(x,y) = \sum_{j=1}^{n_1} \sum_{k=1}^{n_2} \mathbf{d}_{jk} \, N_j^{m_1}(x \,|\, T_1) \, N_k^{m_2}(y \,|\, T_2) \,. \tag{5.8}$$

Since this contains too many double indices, let us simplify the notation.

Definition 5.8 (Tensor product)

1. *By $\mu = (m_1, m_2) \in \mathbb{N}_0^2$ we denote the* ***multidegree*** *of the spline, by $\nu = (n_1, n_2) \in \mathbb{N}^2$ the number of control points in x- and y-direction, respectively. Instead of (x, y) we now write $x = (x_1, x_2)$ for consistency.*
2. *For two multiindices $\alpha, \beta \in \mathbb{N}_0^s$ we write $\alpha \le \beta$ if $\alpha_j \le \beta_j$, $j = 1, \dots, s$. This yields only a* ***partial ordering***[103]
3. *The* ***set product*** *of the knots is defined as*

$$T := T_1 \otimes T_2 = \{ t_\alpha = (t_{\alpha_1}, t_{\alpha_2}) \ : \ \alpha \le \nu + \mu + \mathbf{1} \}, \qquad \mathbf{1} = (1, \dots, 1), \tag{5.9}$$

and the ***tensor product*** *of the spline functions as*

$$N_\kappa^\mu(x \,|\, T) = N_{\kappa_1}^{\mu_1}(x_1 \,|\, T_1) \, N_{\kappa_2}^{\mu_2}(x_2 \,|\, T_2) \,. \tag{5.10}$$

4. *Finally, we write the control points as $\mathbf{d}_\kappa$, $\kappa \in \mathbb{N}_0^2$.*

These notational conventions allow us to write the tensor product **spline surface** as

$$N_\mu \mathbf{d}(x \,|\, T) = \sum_{\kappa \le \nu} \mathbf{d}_\kappa \, N_\kappa^\mu(x \,|\, T) \,, \tag{5.11}$$

which looks almost like the univariate case[104] and can be generalized to an arbitrary number of variables very easily.

Before we generalize tensor products a bit more, we have a quick look at the **de Boor algorithm** for tensor product spline surfaces: Essentially it is just an application of the idea in (5.6) and (5.7), so that we first compute for $j = 0, \dots, \nu_1$ the coefficients

$$d_j(y) = \sum_{k=1}^{n} d_{jk} \, N_k^{\mu_2}(y \,|\, T_2)$$

and then evaluate the spline curve with these coefficients and the knot sequence T_1 at x, see Fig. 5.1.

Exercise 5.1 Formulate and program the de Boor algorithm for bivariate tensor product spline surfaces. ◇

[103]Which means that there exist **incomparable** objects like $\alpha = (1,0)$, $\beta = (0,1)$, for which neither $\alpha \le \beta$ nor $\beta \le \alpha$ holds.

[104]Just with Greek letters.

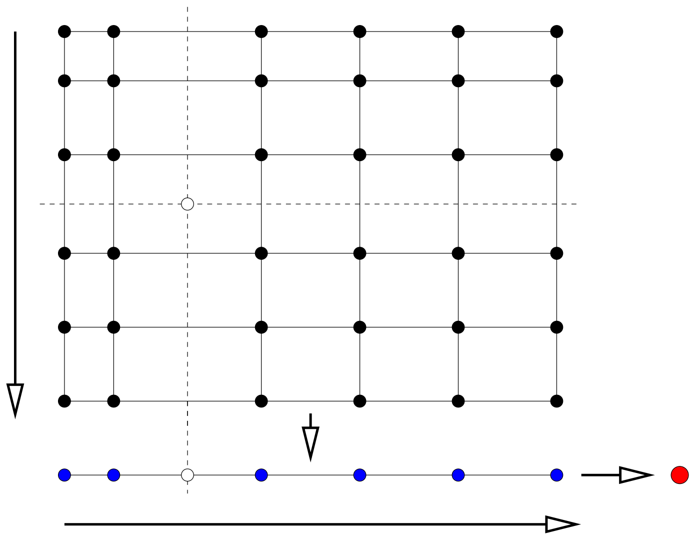

Figure 5.1: The **de Boor algorithm** for bivariate tensor product functions: To each "column" of control points we apply the univariate algorithm of 4.36 and thus get a "row" of controlpoints for $\mathbf{f}_y$. To these coefficients we apply once more the de Boor algorithm, this time with respect to x, and then get the result at the position (x, y).

5.3.2 Tensor product in arbitrarily many variables

The two dimensional concept of "curves along curves" is only the role model for a method that works in *any* number of variables.

Definition 5.9 (Tensor products) *For given $s \in \mathbb{N}$ and univariate splines we define the following objects:*

1. *to knot sequences*

$$T_j = \left\{ t_{j,1}, \dots, t_{j,\nu_j+\mu_j+1} \right\}, \qquad j = 1, \dots, s,$$

*of respective order μ_j the **set product** is given as*

$$T = \bigotimes_{j=1}^{s} T_j := \{ t_\alpha = (t_{1,\alpha_1}, \dots, t_{1,\alpha_s}) \ : \ \alpha \le \nu + \mu + \mathbf{1} \}.$$

2. *the **tensor product B–spline** is defined as*

$$N_\kappa^\mu (x \,|T) = \prod_{j=1}^{s} N_{\kappa_j}^{\mu_j} (x_j \,|\, T_j), \qquad \kappa \le \nu. \tag{5.12}$$

3. *for **control points** $\mathbf{d} = [\mathbf{d}_\kappa \ : \ \kappa \le \nu]$ the **spline curve** is obtained as*

$$N_\mu \mathbf{d}(\cdot \,|\, T) = \sum_{\kappa \le \nu} \mathbf{d}_\kappa \, N_\kappa^\mu (\cdot \,|\, T).$$

The theory developed so far carries over to the tensor product splines as well.

Lemma 5.10 *The B–splines N_κ^μ, $\kappa \le \nu$ form a nonnegative partition of unity.*

Proof: Nonnegativity follows directly from (5.12), for the partition of unity property we use induction on s where the case $s = 1$ should be known to us. With $\mu' = (\mu_1, \dots, \mu_{s-1})$ and respective objects ν', x' und T' we get

$$\begin{aligned} \sum_{\kappa \le \nu} N_\kappa (x \,|\, T) &= \sum_{\kappa' \le \nu'} \sum_{\kappa_s = 1}^{\nu_s} N_{\kappa'}^{\mu'} (x' \,|\, T') \, N_{\kappa_s}^{\mu_s} (x_s \,|\, T_s) \\ &= \sum_{\kappa' \le \nu'} N_{\kappa'}^{\mu'} (x' \,|\, T') \underbrace{\left(\sum_{\kappa_s = 1}^{\nu_s} N_{\kappa_s}^{\mu_s} (x_s \,|\, T_s) \right)}_{=1} \end{aligned}$$

which has value 1 by the induction hypothesis. □

Exercise 5.2 Show that N_κ^μ is supported on the **hypercube**

$$I_\kappa^\mu = \bigotimes_{j=1}^{s} \left[t_{j,\kappa_j}, t_{j,\kappa_j+\mu_j+1} \right].$$

◇

In analogy to what we did before, the multiplicity $\mu(t_\kappa)$ of a knot

$$t_\kappa = (t_{1,\kappa_1}, \dots, t_{s,\kappa_s})$$

is defined as the s–tuple[105]

$$\mu(t_\kappa) = \left(\mu(t_{1,\kappa_1}), \dots, \mu(t_{s,\kappa_s}) \right), \qquad 0 \le \kappa \le \nu + \mu + \epsilon, \tag{5.13}$$

and the space Π_μ of all polynomials of **multidegree**[106] μ is defined as

$$\Pi_\mu := \left\{ p = \sum_{\alpha \le \mu} p_\alpha (\cdot)^\alpha : p_\alpha \in \mathbb{R} \right\} \subset \Pi_{|\mu|}. \tag{5.14}$$

Next, we show that, as expected, the B–splines are also a basis of the tensor product spline space.

Definition 5.11 (Tensor product space) *Let $F_j \subset C(I_j)$, $j = 1, \dots, s$, be linear function spaces. The* ***tensor product space*** *is defined as*

$$F = \bigotimes_{j=1}^{s} F_j := \left\{ \sum_{k=1}^{n} \prod_{j=1}^{s} f_{jk}(x_j) : f_{j,k} \in F_j,\ n \in \mathbb{N} \right\}. \tag{5.15}$$

[105]Recall that $\epsilon \in \mathbb{N}_0^s$ stands for the multiindex $(1, \dots, 1)$.
[106]Note that this is a concept different from the **total degree** that has been used in the context of triangular Bézier surfaces!

Remark 5.12 *It is important to define the tensor product space as all possible sums of functions from* F_j *and not just as as products of functions. If* $\phi_{j,1}, \dots, \phi_{j,n_j}$ *are a basis of* F_j*, then a single product of functions* $f_j \in F_j$ *takes the form*

$$\begin{aligned} f(x) &= \prod_{j=1}^{s} f_j(x_j) = \prod_{j=1}^{s} \sum_{k=1}^{n_j} a_{jk}\phi_{j,k}(x_j) = \sum_{\kappa \le \nu} \underbrace{\left(\prod_{j=1}^{s} a_{j,\kappa_j} \right)}_{=:a_\kappa} \underbrace{\left(\prod_{j=1}^{s} \phi_{j,\kappa_j}(x_j) \right)}_{=:\phi_\kappa} \\ &= \sum_{\kappa \le \nu} a_\kappa \, \phi_\kappa. \end{aligned} \tag{5.16}$$

The ϕ_κ *are obtainable as*

$$\phi_\kappa = \phi_{1,\kappa_1} \otimes \cdots \otimes \phi_{s,\kappa_s}$$

but, for example in the case $n_j = n$*, the function*

$$f(x) = \sum_{k=0}^{n} \phi_{k\epsilon}(x)$$

cannot be written in the form (5.16) *since*

$$1 = a_{k\epsilon} = \prod_{j=1}^{s} a_{jk}$$

implies that all *coefficients* a_{jk} *are nonzero which implies that* all a_κ *would have to be nonzero as well. Hence, we cannot write any function in the space generated by the tensor products of the basis elements as tensor product.*

Lemma 5.13 *If* $\{\phi_{jk} : k = 1, \dots, n_j\}$ *is a basis of* F_j*,* $j = 1, \dots, s$*, then*

$$\phi_\kappa(x) := \prod_{j=1}^{s} \phi_{j,\kappa_j}(x_j), \qquad \kappa \le \nu = (n_1, \dots, n_s),$$

are a basis of $F_1 \otimes \cdots \otimes F_s$*.*

Proof: Using (5.16) for the function $f_k(x) = f_{1k}(x_1) \cdots f_{sk}(x_s)$, we obtain that

$$\sum_{j=1}^{n} f_k = \sum_{k=1}^{n} \sum_{\kappa \le \nu} a_{k,\kappa} \, \phi_\kappa = \sum_{\kappa \le \nu} \left(\sum_{k=1}^{n} a_{k,\kappa} \right) \phi_\kappa,$$

hence *any* element of F can be written in the form (5.16), therefore the functions ϕ_κ are a **generating system** for the tensor product space and they are a **basis** for the space provided they are linearly independent. To show this, set $\widehat{\kappa} = (\kappa_1, \dots, \kappa_{s-1})$ and $\widehat{\nu}$ and $\widehat{x}$ respectively and assume that

$$0 = \sum_{\kappa \le \nu} a_\kappa \, \phi_\kappa = \sum_{\kappa_s=1}^{n_s} \underbrace{\sum_{\widehat{\kappa} \le \widehat{\nu}} a_{(\widehat{\kappa},k)} \phi_{\widehat{\kappa}}(\widehat{x})}_{=:a_k(\widehat{x})} \; \phi_{s,k}(x_s) = \sum_{\kappa_s=1}^{n_s} a_k(\widehat{x}) \, \phi_{s,k}(x_s). \tag{5.17}$$

Since the functions $\phi_{s,k}$ are linearly independent, the "coefficients" $a_k(\widehat{x})$ have to be zero for all $\widehat{x}$ and since this is a tensor product function in $s - 1$ variables, the proof can be completed by a simple induction. □

Corollary 5.14 $\dim(F_1 \otimes \cdots \otimes F_s) = \dim F_1 \cdots \dim F_s$.

As simple as Corollary 5.14 is, it has a fundamental consequence that is known as the **curse of dimension**: the dimension of a tensor product space grows *exponentially* in the number of variables. For example, even if we have only 10 basis function in either variable, the dimension of the full space and therefore the number of coefficients to store, is 10^s which quite fast exceed available storage capacities.
As an immediate conseuqence of Lemma 5.13, we can give the basis of the spline space.

Theorem 5.15 *The tensor product spline space*

$$\mathbb{S}_\mu(T) = \bigotimes_{j=1}^{s} \mathbb{S}_{m_j}(T_j), \qquad T = \bigotimes_{j=1}^{s} T_j, \tag{5.18}$$

1. *consists of piecewise polynomials of multidegree* μ.
2. *is spanned by the B–splines* $N_\kappa^\mu(\cdot|T)$, $\kappa \le \nu$.

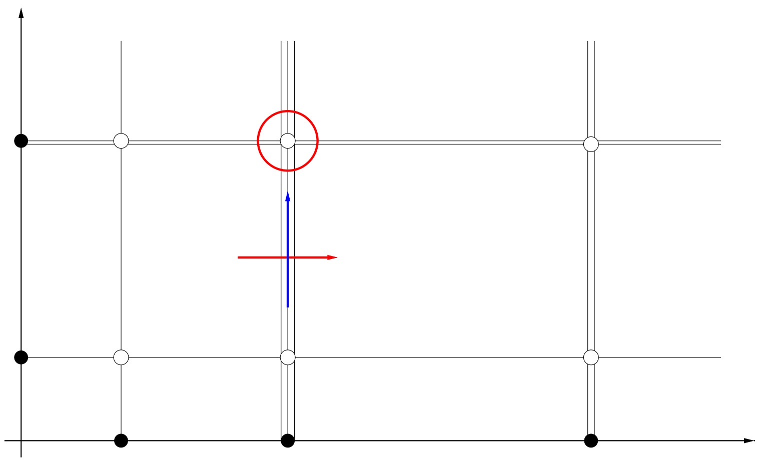

Figure 5.2: Continuity conditions of bivariate tensor product splines. They are different across knot lines and around knots.

Remark 5.16 (Differentiability) *One would expect in 1) a description of global smoothness as well, but this is more complicated, even in two variables as it is shown in Fig 5.2. The black dots are the knots in* x, $\{\dots, t_j, t_{j+1}, t_{j+1}, t_{j+1}, t_{j+2}, t_{j+2}, \dots\}$ *and* y, $\{\dots, t_j, t_{j+1}, t_{j+1}, \dots\}$, *respectively, the white dots are the tensor product knots of multiplicities* $(1,1)$, $(3,1)$, $(2,1)$, $(1,2)$, $(3,2)$ *and* $(2,2)$. *We now have different situations:*

1. *in each of the rectangles formed by the horizontal and vertical lines, the spline is a polynomial, hence* C^∞.

2. *In the direction of the lines the spline is a piecewise polynomial in one variable and as long as we stay away from the knots* (blue arrow) *this curve is a* C^∞ *function as well.*

3. *Across a "knot line"* (red arrow), *things are different as now the order of differentiability depends on the multiplicity of the knot, in the example we would lose* 3 *orders. Hence, on a vertical knot line, but away from the knot itself, the function belongs to* $C^{\mu_1-\mu(t_{1,\kappa_1}),\infty}$ *where*

$$C^\gamma = \left\{ f : \frac{\partial^{|\alpha|}}{\partial x^\alpha} f \in C, \alpha \le \gamma \right\}. \tag{5.19}$$

On a horizontal line, away from the knots, the smoothness is $C^{\infty,\mu_2-\mu(t_{2,\kappa_2})}$.

4. *Around the knot* t_κ (red circle) *the spline finally belongs to* $C^{\mu-\mu(t_\kappa)}$, *so the knots are the least differentiable points.*

It is easy to imagine that this becomes even more complicated in three and more variables, but it is always the set of "active knot projetcions" that determines the differentiability.

Just for completeness ...

Definition 5.17 *The space* C^γ *from* (5.19) *is called* ***anisotropic smoothness space*** *as the order of differentiability can be different in different variables.*

In general, derivatives of tensor products are easy to compute: Whenever a function $f(x)$ can be decomposed into $f(x) = f_1(x_1)\cdots f_s(x_s)$, we have

$$\frac{\partial f}{\partial x_j}(x) = f_1(x_1)\cdots f_{j-1}(x_{j-1}) \underbrace{\frac{\partial f}{\partial x_j} f_j(x_j)}_{=f_j'(x_j)} f_{j+1}(x_{j+1})\cdots f_s(x_s), \tag{5.20}$$

partial derivatives become univariate derivatives for components. Hence we get the following formula for a partial derivative of a tensor product spline:

$$\begin{aligned}
&\frac{\partial}{\partial x_j} N_\mu \mathbf{d}(\cdot|T) = \frac{\partial}{\partial x_j} \sum_{\kappa\le\nu} \mathbf{d}_\kappa N_\kappa^\mu(\cdot|T) = \sum_{\kappa\le\nu} \mathbf{d}_\kappa \frac{\partial}{\partial x_j} N_\kappa^\mu(\cdot|T) \\
&= \sum_{\kappa\le\nu} \mathbf{d}_\kappa N_{\kappa_1}^{m_1}(\cdot|T_1)\cdots N_{\kappa_{j-1}}^{m_{j-1}}(\cdot|T_{j-1}) \left(N_{\kappa_j}^{m_j}(\cdot|T_j)\right)' N_{\kappa_{j+1}}^{m_{j+1}}(\cdot|T_{j+1})\cdots N_{\kappa_s}^{m_s}(\cdot|T_s) \\
&= m_j \sum_{\kappa\le\nu+\epsilon_j} \mathbf{d}_\kappa N_{\kappa_1}^{m_1}(\cdot|T_1)\cdots \frac{N_{\kappa_j}^{m_j-1}(\cdot|T_j)}{t_{j,\kappa_j+m_j} - t_{j,\kappa_j}} \cdots N_{\kappa_s}^{m_s}(\cdot|T_s) \\
&\quad - m_j \sum_{\kappa\le\nu+\epsilon_j} \mathbf{d}_\kappa N_{\kappa_1}^{m_1}(\cdot|T_1)\cdots \frac{N_{\kappa_j}^{m_j-1}(\cdot|T_j)}{t_{j,\kappa_j+m_j+1} - t_{j,\kappa_j+1}} \cdots N_{\kappa_s}^{m_s}(\cdot|T_s) \\
&= m_j \sum_{\kappa\le\nu+\epsilon_j} \frac{\mathbf{d}_\kappa - \mathbf{d}_{\kappa+\epsilon_j}}{t_{j,\kappa_j+m_j} - t_{j,\kappa_j}} N_\kappa^{\mu-\epsilon_j}(\cdot|T),
\end{aligned}$$

which we record as

$$\frac{\partial}{\partial x_j} N_\mu \mathbf{d}(\cdot|T) = m_j \sum_{\kappa \le \nu + \epsilon_j} \frac{\mathbf{d}_\kappa - \mathbf{d}_{\kappa+\epsilon_j}}{t_{j,\kappa_j+m_j} - t_{j,\kappa_j}} N_\kappa^{\mu-\epsilon_j}(\cdot|T). \tag{5.21}$$

In other words: Any partial derivative turns into a partial difference applied to the coefficients weighted with the difference of the knots. If the knots are **equidistant**, i.e., $t_{j,k+1} - t_{j,k} = h_j$, $k = 0, \dots, n_j + m_j + 1$, then it is also easy to compute higher order derivatives:

$$\frac{\partial^{|\alpha|}}{\partial x^\alpha} N_\mu \mathbf{d}(\cdot|T) = \frac{(\mu-\alpha)!}{h^\alpha \mu!} \sum_{\kappa \le \nu+\alpha} \sum_{\beta \le \alpha} (-1)^\beta \binom{\alpha}{\beta} \mathbf{d}_{\kappa+\beta} N_\kappa^{\mu-\alpha}(\cdot|T), \tag{5.22}$$

where

$$(-1)^\beta := \prod_{j=1}^{s} (-1)^{\beta_j}, \qquad \text{and} \qquad \binom{\alpha}{\beta} := \prod_{j=1}^{s} \binom{\alpha_j}{\beta_j}.$$

Exercise 5.3 Prove (5.22). ◇

5.3.3 Twists

As mentioned in (Farin, 1988), there are peculiar partial derivatives with a special geometric meaning.

Definition 5.18 (Twist) *A **twist** of a function* f *at* x *is any mixed second order partial derivative* $\frac{\partial^2}{\partial x_j \partial x_k} f$, $j \ne k$. *In the case* $s = 2$ *there is only* the *twist* $\frac{\partial^2}{\partial x \partial y}$.

To find out about the geometric meaning, we apply (5.21) twice and get

$$\begin{aligned}
\frac{\partial^2}{\partial x_j \partial x_k} N_\mu \mathbf{d} &= \frac{\partial}{\partial x_k} m_j \sum_{\kappa \le \nu+\epsilon_j} \frac{\mathbf{d}_\kappa - \mathbf{d}_{\kappa+\epsilon_j}}{t_{j,\kappa_j+m_j} - t_{j,\kappa_j}} N_\kappa^{\mu-\epsilon_j}(\cdot|T) \\
&= m_j m_k \sum_{\kappa \le \nu+\epsilon_j+\epsilon_k} \frac{\mathbf{d}_\kappa - \mathbf{d}_{\kappa+\epsilon_j} - \mathbf{d}_{\kappa+\epsilon_k} + \mathbf{d}_{\kappa+\epsilon_j+\epsilon_k}}{(t_{j,\kappa_j+m_j} - t_{j,\kappa_j})(t_{k,\kappa_k+m_k} - t_{k,\kappa_k})} N_\kappa^{\mu-\epsilon_j-\epsilon_k}(\cdot|T) \\
&=: m_j m_k \sum_{\kappa \le \nu+\epsilon_j+\epsilon_k} \frac{\Delta^{\epsilon_j+\epsilon_k} \mathbf{d}_\kappa}{(t_{j,\kappa_j+m_j} - t_{j,\kappa_j})(t_{k,\kappa_k+m_k} - t_{k,\kappa_k})} N_\kappa^{\mu-\epsilon_j-\epsilon_k}(\cdot|T)
\end{aligned}$$

The three points $\mathbf{d}_\kappa$, $\mathbf{d}_{\kappa+\epsilon_j}$, $\mathbf{d}_{\kappa+\epsilon_k}$ define a two dimensional plane $[\![\mathbf{d}_\kappa, \mathbf{d}_{\kappa+\epsilon_j}, \mathbf{d}_{\kappa+\epsilon_k}]\!]_*$ and in this plane there is the **parallelogram point** $\mathbf{p}_\kappa$ defined by

$$\mathbf{p}_\kappa - \mathbf{d}_{\kappa+\epsilon_k} = \mathbf{d}_{\kappa+\epsilon_j} - \mathbf{d}_\kappa, \tag{5.23}$$

which is nothing but the definition of a parallelogram: two opposite faces are parallel and of equal length. Hence,

$$\Delta^{\epsilon_j+\epsilon_k} \mathbf{d}_\kappa = 0 \qquad \Leftrightarrow \qquad \mathbf{d}_{\kappa+\epsilon_j+\epsilon_k} = \mathbf{p}_\kappa$$

which means that the twist of the surface is related to the **planarity** of the two dimensional faces.

Corollary 5.19 (Twist) *A bivariate spline surface has twist zero if and only if all the **quadrilateral** faces of the control polyhedron are planar.*

Proof: The spline surface

$$\begin{aligned} & \frac{\partial^2}{\partial x \partial y} N_{(m_1,m_2)} \mathbf{d} \\ = \ & m_j m_k \sum_{k_1,k_2=0}^{n_1+1,n_2+1} \frac{\Delta^{(1,1)} \mathbf{d}_{(k_1,k_2)}}{(t_{1,k_1+m_1} - t_{1,k_1})(t_{2,k_2+m_2} - t_{2,k_2})} N_\kappa^{(m_1-1,m_2-1)}(\cdot|T) \end{aligned}$$

is identically zero if and only if $\Delta^{(1,1)} \mathbf{d}_{(k_1,k_2)} = 0$. □

5.3.4 Interpolation by tensor product splines

If we want to interpolate with tensor product splines, the most natural thing to do is to take, for each coordinate x_j, $j = 1, \dots, s$, a set $X_j = \{x_{j,k} : k = 0, \dots, n\}$ of interpolation points that satisfy the **Schoenberg–Whitney condition**

$$t_{j,k} < x_{j,k} < t_{j,k+m_j+1}, \qquad k = 0, \dots, n_j, \quad j = 1, \dots, s, \tag{5.24}$$

from Theorem 4.56 and to form their tensor product or the **grid**

$$X = \bigotimes_{j=1}^{s} X_j = \{x_\alpha = (x_{1,\alpha_1}, \dots, x_{s,\alpha_s}) : \alpha \le \nu\}. \tag{5.25}$$

Such a grid always allows for unique interpolation.

Theorem 5.20 (Schoenberg–Whitney for tensor product) *The spline space* $\mathbb{S}_\mu(T)$ *allows for unique interpolation from the set* X *in* (5.25) *if the coordinate projections satisfy the respective Schoenberg–Whitney condition* (5.24).

Proof: Because of the univariate Schoenberg–Whitney theorem there exist[107] splines $s_{j,k} \in \mathbb{S}_{m_j}(T_j)$ such that

$$s_{j,k}(x_{j,k'}) = \delta_{k,k'}, \qquad k, k' = 0, \dots, n_j, \quad j = 1, \dots, s, \tag{5.26}$$

and therefore the spline functions

$$\mathbb{S}_\mu(T) \ni s_\alpha(x) := \prod_{j=1}^{s} s_{j,\alpha_j}(x_j), \qquad \alpha \le \nu,$$

satisfy

$$s_\alpha(x_\beta) = \prod_{j=1}^{s} s_{j,\alpha_j}(x_{j,\beta_j}) = \prod_{j=1}^{s} \delta_{\alpha_j,\beta_j} = \delta_{\alpha,\beta}, \qquad \alpha, \beta \le \nu, \tag{5.27}$$

[107] The splines defined in (5.26) are solutions of special interpolation problems.

from which we can conclude that

$$s_f := \sum_{\alpha \le \nu} f(x_\alpha) s_\alpha$$

interpolates f on X. Indeed, by (5.27),

$$s_f(x_\beta) = \sum_{\alpha \le \nu} \underbrace{f(x_\alpha) s_\alpha(x_\beta)}_{=\delta_{\alpha,\beta}} = f(x_\beta), \qquad \beta \le \nu.$$

For uniqueness we have to show that $s_f = s_g$ implies that $f(X) = g(X)$, that is $f(x_\alpha) = g(x_\alpha)$, $\alpha \le \nu$. So suppose that $s_f = s_g$ or, equivalently,

$$0 = s_f - s_g = \sum_{\alpha \le \nu} \left(f(x_\alpha) - g(x_\alpha) \right) s_\alpha.$$

Substituting x_β into this identity, again (5.27) yields that $0 = f(x_\beta) - g(x_\beta)$ which completes the proof. □

Remark 5.21 *That the points lie on a tensor grid of points that satisfy the Schoenberg–Whitney condition is* sufficient *for unique interpolation but in no way necessary. To see that, note that unqiue solvability of the interpolation problem is equivalent to the nonsigularity of the* ***collocation matrix***

$$N_\mu(X|T) := \left[N_\kappa^\mu(x_\beta|T) : \begin{array}{c} \beta \le \nu \\ \kappa \le \nu \end{array} \right].$$

In other words, $\det N_\mu(X|T) \neq 0$. *Since the determinant is a continuous function in all the $x_{j,k}$, the matrix remains nonsingular if to each of them a sufficiently small perturbation is applied. However, the grid structure can be destroyed by arbitrarily small perturbations, hence there are many more configurations that are* not *tensor grids of "good" points.*

There is, however more interesting structure behind tensor product interpolation which uses a very nice concept from linear algebra.

Definition 5.22 (Kronecker product) *The* ***Kronecker product*** *or* ***Zehfuss product***[108] *of two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$ is defined as the* ***block matrix***

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{mp\times nq}. \tag{5.28}$$

The multiple Kronecker product of $A_1, \dots, A_n$ is then give as

$$A_1 \otimes \cdots \otimes A_n = (A_1 \otimes \cdots \otimes A_{n-1}) \otimes A_n. \tag{5.29}$$

[108]For this interesting story see the very nice set of slides (Van Loan, 2009).

The definition (5.29) makes sense because the Kronecker product is **associative**:

$$
\begin{aligned}
&(A \otimes B) \otimes C \\
&= \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \otimes C \\
&= \left[\begin{array}{ccc|c|ccc}
a_{11}b_{11}C & \dots & a_{11}b_{1q}C & \dots & a_{1n}b_{11}C & \dots & a_{1n}b_{1q}C \\
\vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
a_{11}b_{p1}C & \dots & a_{11}b_{pq}C & \dots & a_{1n}b_{p1}C & \dots & a_{1n}b_{pq}C \\
\hline
\vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\hline
a_{m1}b_{11}C & \dots & a_{m1}b_{1q}C & \dots & a_{mn}b_{11}C & \dots & a_{mn}b_{1q}C \\
\vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
a_{m1}b_{p1}C & \dots & a_{m1}b_{pq}C & \dots & a_{mn}b_{p1}C & \dots & a_{mn}b_{pq}C
\end{array}\right] \\
&= \begin{bmatrix} a_{11}B \otimes C & \dots & a_{1n}B \otimes C \\ \vdots & \ddots & \vdots \\ a_{m1}B \otimes C & \dots & a_{mn}B \otimes C \end{bmatrix} = A \otimes (B \otimes C).
\end{aligned}
$$

However, the Kronecker product is *not* **comuutative**, that is, in general $A \otimes B \neq B \otimes A$.

Exercise 5.4 Prove that $(A \otimes B)^T = A^T \otimes B^T$. ◊

Lemma 5.23 *For* $A \in \mathbb{R}^{m \times n}$, $A' \in A \in \mathbb{R}^{n \times n'}$, $B \in \mathbb{R}^{p \times q}$, $B' \in \mathbb{R}^{q \times q'}$ *we have*

$$(A \otimes B)(A' \otimes B') = (AA') \otimes (BB') \tag{5.30}$$

and for nonsingular matrices

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}. \tag{5.31}$$

Proof: We can write (5.28) explicitly as

$$(A \otimes B)_{(j-1)p+r,(k-1)q+s} = a_{jk}\, b_{rs}, \qquad \begin{array}{ll} j = 1, \dots, m & r = 1, \dots, p \\ k = 1, \dots, n, & s = 1, \dots, q, \end{array} \tag{5.32}$$

for which the multiplication formula for matrices yields

$$
\begin{aligned}
&((A \otimes B)(A' \otimes B'))_{(j-1)p+r,(j'-1)p'+r'} \\
&= \sum_{\ell=1}^{nq} (A \otimes B)_{(j-1)p+r,\ell} (A' \otimes B')_{\ell,(j'-1)p'+r'} \\
&= \sum_{k=1}^{n} \sum_{s=1}^{q} (A \otimes B)_{(j-1)p+r,(k-1)q+s} (A' \otimes B')_{(k-1)q+s,(j'-1)p'+r'} \\
&= \sum_{k=1}^{n} \sum_{s=1}^{q} a_{jk} b_{rs} a'_{kj'} b'_{sr'} = \underbrace{\sum_{k=1}^{n} a_{jk} a'_{kj'}}_{=(AA')_{jj'}} \underbrace{\sum_{s=1}^{q} b_{rs} b'_{sr'}}_{=(BB')_{rr'}} \\
&= ((AA') \otimes (BB'))_{(j-1)p+r,(j'-1)p'+r'},
\end{aligned}
$$

which proves (5.30) from which[109]

$$(A^1 \otimes B^{-1})(A \otimes B) = (A^{-1}A) \otimes (B^{-1}B) = I \otimes I = I$$

allows us to conclude (5.31) as well. □

These strange identities are relevant since, when ordered appropriately, the collocation matrices for tensor product functions on a grid are Kronecker products.

Definition 5.24 (Lexicographic ordering) *The* ***lexicographic ordering*** $\prec$ *is defined as*

$$\alpha \prec \beta \qquad \Leftrightarrow \qquad \alpha_j = \beta_j, \quad j = 1, \dots, k-1, \quad \alpha_k < \beta_k. \tag{5.33}$$

Exercise 5.5 Prove that the lexicographical ordering is a total ordering on the set $\mathbb{N}_0^s$ of multiindices. ◊

Proposition 5.25 *If the multiindices are arranged in lexicographical order, the* ***collocation matrix*** *of the tensor product B–splines with respect to the grid* $X_1 \otimes \cdots \otimes X_s$ *takes the form*

$$N_\mu(X|T) = N_{m_1}(X_1|T_1) \otimes \cdots \otimes N_{m_s}(X_s|T_s). \tag{5.34}$$

Proof: [110] The lexicographical ordering arranges the multiindices as

$$(0, \alpha_{0,1}), \dots, (0, \alpha_{0,N}), (1, \alpha_{1,1}), \dots, (n_1, \alpha_{n_1,N}), \qquad N = n_2 \cdots n_s,$$

where $\alpha_{j,k} \prec \alpha_{j,k+1}$. Hence, the collocation matrix is of the form

$$\begin{aligned}
&\begin{bmatrix} N_0^{m_1}(x_0|T_1)\, N_{\widehat{\mu}}(\widehat{X}|\widehat{T}) & \dots & N_{n_1}^{m_1}(x_0|T_1)\, N_{\widehat{\mu}}(\widehat{X}|\widehat{T}) \\ \vdots & \ddots & \vdots \\ N_0^{m_1}(x_{n_1}|T_1)\, N_{\widehat{\mu}}(\widehat{X}|\widehat{T}) & \dots & N_{n_1}^{m_1}(x_{n_1}|T_1)\, N_{\widehat{\mu}}(\widehat{X}|\widehat{T}) \end{bmatrix} \\
&= \quad N_{m_1}(X_1|T_1) \otimes N_{\widehat{\mu}}(\widehat{X}|\widehat{T}) = \cdots = N_{m_1}(X_1|T_1) \otimes \cdots \otimes N_{m_s}(X_s|T_s),
\end{aligned}$$

which is formally proved by induction on s. Here $\widehat{\mu} = (\mu_2, \dots, \mu_s)$ corresponds to cancellation of the first index. □

By (5.34) and (5.31) we can now "easily solve" the interpolation problem on gridded data. Given $\mathbf{y} = (\mathbf{y}_\kappa : \kappa \le \nu) \in \mathbb{R}^{d \times n}$, $n := \prod n_j$, the linear system[111] to solve is

$$\mathbf{y}^T = N_\mu(X|T)\mathbf{d}^T = \left(N_{m_1}(X_1|T_1) \otimes \cdots \otimes N_{m_s}(X_s|T_s)\right) \mathbf{d}^T,$$

hence

$$\mathbf{d} = \left(N_{m_1}(X_1|T_1)^{-T} \otimes \cdots \otimes N_{m_s}(X_s|T_s)^{-T}\right) \mathbf{y}. \tag{5.35}$$

Some remarks:

1. Although we solve the huge $n \times n$ system, where again $n = n_1 \cdots n_s$, we only have to invert *small* matrices of size $n_j \times n_j$. This is the good news.

[109] The identity matrices in the following equation are of different size, keep that in mind.

[110] This is not really a proof, it is more bookkeeping.

[111] The transposition is only used to make dimensions fit.

2. Nevertheless, the whole thing would become pointless if we still would have to expand the Kronecker product into the full matrix.

3. And the main warning: Noone who has the slightest idea of **Numerical Linear Algebra** would compute the inverse of a matrix explicitly, see (Golub & van Loan, 1996; Higham, 2002; Sauer, 2013).

4. In summary: we have to find a smarter way to evaluate (5.35).

Definition 5.26 *The **vectorization** of a matrix $A \in \mathbb{R}^{m\times n}$ is the vector*

$$v(A) = \begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \\ \vdots \\ a_{1n} \\ \vdots \\ a_{mn} \end{bmatrix} \in \mathbb{R}^{mn}$$

of stacked column vectors.

There exists a cute formula for Kronecker products that can be found, for example, in (Horn & Johnson, 1991; Marcus & Minc, 1969).

Proposition 5.27 *For $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times q}$ and $X \in \mathbb{R}^{n\times p}$ we have*

$$v(AXB) = (B^T \otimes A)v(X). \tag{5.36}$$

Before proving the proposition, let us first check that the dimensions coincide on both sides of (5.36). Since $AXB \in \mathbb{R}^{m\times q}$ the expression on the left hand side is a vector of size mq, the Kronecker product on the right hand side belongs to $\mathbb{R}^{qm\times pn}$ and $v(X)$ is an np–vector, so the right hand side indeed also gives a vector of size mq.

Proof: We write $X = [x_1 \dots x_p]$, $B = \left[b_{jk} : \begin{array}{c} j = 1,\dots,p \\ k = 1,\dots,q \end{array} \right]$ and get for the ℓth column of the product that

$$\begin{aligned}
(AXB)_\ell &= AXBe_\ell = AX\,[b_{j\ell} : j = 1,\dots,p] \\
&= A\left(\sum_{j=1}^{p} x_j b_{j\ell}\right) = \sum_{j=1}^{p} b_{j\ell}\, Ax_j = [b_{1\ell}A \dots b_{p\ell}A]\, v(X) \\
&= \left((Be_\ell)^T \otimes A\right) v(X)
\end{aligned}$$

and therefore

$$\begin{aligned}
v(AXB) &= \begin{bmatrix} (AXB)_1 \\ \vdots \\ (AXB)_q \end{bmatrix} = \begin{bmatrix} ((Be_1)^T \otimes A)\, v(X) \\ \vdots \\ ((Be_p)^T \otimes A)\, v(X) \end{bmatrix} = \begin{bmatrix} ((Be_1)^T \otimes A) \\ \vdots \\ ((Be_p)^T \otimes A) \end{bmatrix} v(X) \\
&= \left(\begin{bmatrix} (Be_1)^T \\ \vdots \\ (Be_p)^T \end{bmatrix} \otimes A \right) v(X) = (B^T \otimes A)v(X)
\end{aligned}$$

as claimed. □

The "Kronecker trick" (5.36) allows us to compute the product of a Kronecker product $A_1 \otimes \cdots \otimes A_s$, $A_j \in \mathbb{R}^{m_j \times n_j}$ and a vector x. To that end, we interprete $x \in \mathbb{R}^n$ as $\nu(X)$ for a matrix $X \in \mathbb{R}^{n_2 \cdots n_s \times n_1}$ and obtain

$$\begin{aligned}(A_1 \otimes \cdots \otimes A_s)x &= (A_1 \otimes \cdots \otimes A_s)\nu(X) = (A_2 \otimes \cdots \otimes A_s)XA_1^T \\ &=: (A_2 \otimes \cdots \otimes A_s)X_1\end{aligned}$$

where $X_1 \in \mathbb{R}^{n_2 \cdots n_s \times m_1}$ and the expression can be evaluated recursively for the columns $[x_{11} \dots x_{1,m_1}]$ of X_1 which are vectors in $\mathbb{R}^{n_2 \cdots n_s}$.

Let us apply this to (5.35) and assume that $d = 1$, i.e., $y \in \mathbb{R}^n$. Here, we are dealing with the *square matrices* $N_{m_j}(X_j|T_j)$. The first step cuts y into a matrix $Y_0 \in \mathbb{R}^{n_2 \cdots n_s \times n_1}$ with $y = \nu(Y_0)$ and computes the product

$$Y_1 = Y_0\, N_1^{m_1}(X_1|T_1)^{-1} \qquad \Leftrightarrow \qquad N_1^{m_1}(X_1|T_1)^T\, Y_1^T = Y_0^T. \tag{5.37}$$

The linear system on the right hand side can be solved by any standard method from Numerical Linear Algebra and just requires the solution of a system of size $n_1 \times n_1$ for each column of Y_0^T of which we have $n_2 \cdots n_s$. For each of the n_1 columns $y_{1,j}$ of Y_1 we form matrices $Y_{1,j} \in \mathbb{R}^{n_3 \cdots n_s \times n_2}$ and solve systems

$$N_2^{m_2}(X_2|T_2)\, Y_{2,j}^T = Y_{1,j}^T, \qquad j = 1, \dots, n_1, \tag{5.38}$$

which can be packed into

$$N_2^{m_2}(X_2|T_2)\, \left[Y_{2,j}^T : j = 1, \dots, n_1\right] = \left[Y_{1,j}^T : j = 1, \dots, n_1\right]. \tag{5.39}$$

This leads to a combination of solving univariate linear systems and rearrangements of a vector of size n that can be used to solve the interpolation without even having to compute the collocation matrix explicitly.

We can even estimate the complexity of this algorithm. Storage is quite cheap as the memory requirement for the coordinate collocation matrices is

$$\sum_{j=1}^{s} n_j^2 \ll n^2, \qquad n = n_1 \cdots n_s,$$

and the cost for solving (5.39) in terms of **flops**[112] is $C\, n_j^3$ for the decomposition of the matrices[113] and[114] $C\, n_j^2$ for solving for each column of the reshaped Y_j. Since this matrix has n/n_j columns, the total effort in a single step is $C\, n n_j$ and the total computational effort is bounded by

$$\sum_{j=1}^{s} n_j^3 + n \sum_{j=1}^{s} n_j \qquad \sim sp^3 + sp^{s+1}, \quad n_j = p. \tag{5.40}$$

[112] This is an abbreviation for "**flo**ating point **op**eratrion**s**".

[113] Usually by means of Gauß elimination, see (Sauer, 2013), which can even be done in a very stable way since the matrix is **totally nonnegative**. And, by the way, it's even cheaper, the effort is only $Cm_j n_j^2 \ll Cn_j^2$ as long as $m \ll n$.

[114] The two optimal constants are different but both reasonable, just take the larger one.

This is really cheap[115] for solving a system of size $n \times n$ which would usually cost $\sim n^3$ or $\sim p^{3s}$ provided that the univariate dimensions are all the same.

This algorithm is due to de Boor (Boor, 1979a; Boor, 1979b), however, with a slightly different proof, the application of the Kronecker trick for matrix-vector multiplication can be found in different forms in (Lamping *et al.*, 2015; Van Loan & Pitsianis, 1993).

Remark 5.28 (Tensor product interpolation) *It seems as if tensor product interpolation almost overcomes the "curse of dimension", but there are still two huge objects with $n \sim p^s$ components:*

1. *the coefficients of the resulting spline.*
2. *the vector of data values.*

In particular, this means that in order to interpolate a function with a spline surface in higher dimensions, one has to know that functions at many locations which is not always easy in practical applications, cf. (Votsmeier et al.*, 2010).*

Theorem 5.29 (Tensor product splines on grids) *Tensor product spline interpolation on grids has a unique solution if and only if the coordinate projections satisfy the respective Schoenberg–Whitney condition. The coefficients of the interpolant can be computed efficiently.*

Proof: The only thing still left to prove is that nonsingularity of $N_\mu(X|T)$ implies the nonsingularity of the "Kronecker factors" $N_{m_j}(X_j|T_j)$, $j = 1, \dots, s$. This is a general Kronecker thing, however: Suppose for some j there exists x_j such that $A_j x_j = 0$, then, by (5.30), we have, for any $x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_s$ that

$$(A_1 \otimes \cdots \otimes A_s)(x_1 \otimes \cdots \otimes x_s) = (A_1 x_1) \otimes \cdots \otimes \underbrace{(A_j x_j)}_{=0} \otimes \cdots \otimes (A_s x_s) = 0 \tag{5.41}$$

since

$$0 \times A = \begin{bmatrix} 0A & \dots & 0A \\ \vdots & \ddots & \ddots \\ 0A & \dots & 0A \end{bmatrix} = 0 \qquad \text{and} \qquad A \otimes 0 = \begin{bmatrix} a_{11}\,0 & \dots & a_{1n}\,0 \\ \vdots & \ddots & \ddots \\ a_{m1}\,0 & \dots & a_{mn}\,0 \end{bmatrix} = 0,$$

so that any Kronecker product that contains a zero factor must be entirely zero. Taking that into account, (5.41) shows that the Kronecker product cannot be nonsingular if a s single factor is nonsingular and the converse of this statement is given by the formula (5.31). □

5.4 Surfaces from boundary curves: Coons patches

Another simple method to create surfaces from curves is to do it by **blending** the boundary curves of a four–sided surface in $\mathbb{R}^3$. To do so, we start with four curves which together form a **closed curve**

$$\mathbf{f}_j : [0,1] \to \mathbb{R}^3, \quad j = 1, \dots, 4, \qquad \mathbf{f}_j(1) = \mathbf{f}_{j+1}(0), \quad \mathbf{f}_5 := \mathbf{f}_1, \tag{5.42}$$

This curve encloses a four–sided patch and forms its boundary curve.

[115]Besides the fact that even the full matrix even cannot be stored at all.

5.4.1 Coons patches

Now we connect the boundary curves on opposition sides by means of another curve, see Fig. 5.3:

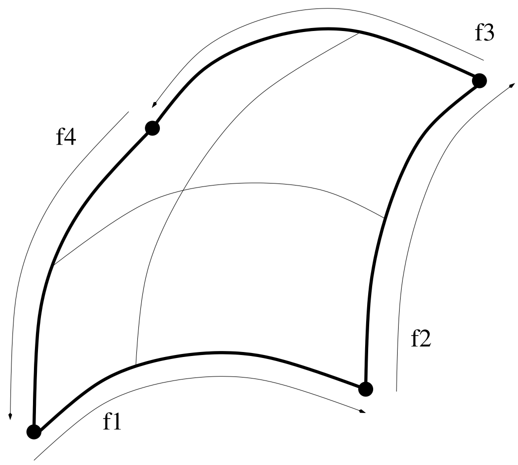

Figure 5.3: Boundary curves of the four sided patch and two blending curves.

$$\begin{aligned} \mathbf{F}_1(x,y) &= (1-g_2(y))\,\mathbf{f}_1(x) + g_2(y)\,\mathbf{f}_3(1-x), \\ \mathbf{F}_2(x,y) &= (1-g_1(x))\,\mathbf{f}_4(1-y) + g_1(x)\,\mathbf{f}_2(y), \end{aligned} \qquad (x,y)\in[0,1]^2, \tag{5.43}$$

where the two *scalar* **blending curves** g_1 and g_2 satisfy $g_j(0) = 0$ and $g_j(1) = 1$. The idea is that $\mathbf{f}_1$ and $\mathbf{f}_3$ are connected in y–direction and thus form the boundary curves $F(\cdot,0)$ and $F(\cdot,1)$ of some surface F and that $\mathbf{f}_2$ and $\mathbf{f}_4$ are likewise connected in x–direction. Note that for that purpose the opposite curves have to be parametrized in opposite directions since the "boundary curve" was defined in a closed form.

If we restrict the sum $\mathbf{F}_+ = \mathbf{F}_1 + \mathbf{F}_2$ two the boundary $[0,1]\times 0$, we get

$$\begin{aligned} \mathbf{F}_+(x,0) &= \underbrace{(1-g_2(0))}_{=1}\,\mathbf{f}_1(x) + \underbrace{g_2(0)}_{=0}\,\mathbf{f}_3(1-x) + (1-g_1(x))\,\mathbf{f}_4(1) + g_1(x)\,\mathbf{f}_2(0) \\ &= \mathbf{f}_1(x) + (1-g_1(x))\,\mathbf{f}_4(1) + g_1(x)\,\mathbf{f}_2(0). \end{aligned}$$

In the same fashion,

$$\begin{aligned} \mathbf{F}_+(x,1) &= \mathbf{f}_3(1-x) + (1-g_1(x))\,\mathbf{f}_4(0) + g_1(x)\,\mathbf{f}_2(1), \\ \mathbf{F}_+(0,y) &= \mathbf{f}_4(1-y) + (1-g_2(y))\,\mathbf{f}_1(0) + g_2(y)\,\mathbf{f}_3(1), \\ \mathbf{F}_+(1,y) &= \mathbf{f}_2(y) + (1-g_2(y))\,\mathbf{f}_1(1) + g_2(y)\,\mathbf{f}_3(0). \end{aligned}$$

If we know define the tensor function

$$\begin{aligned} \mathbf{G}(x,y) &= \mathbf{f}_1(0)(1-g_1(x))(1-g_2(y)) + \mathbf{f}_2(0)\,g_1(x)(1-g_2(y)) \\ &\quad + \mathbf{f}_4(0)\,(1-g_1(x))g_2(y) + \mathbf{f}_3(0)\,g_1(x)g_2(y). \end{aligned} \tag{5.44}$$

which satisfies

$$\mathbf{G}(0,0) = \mathbf{f}_1(0), \quad \mathbf{G}(1,0) = \mathbf{f}_2(0), \quad \mathbf{G}(1,1) = \mathbf{f}_3(0), \quad \mathbf{G}(0,1) = \mathbf{f}_4(0),$$

and set $\mathbf{F} := \mathbf{F}_+ - \mathbf{G}$, then we obtain

$$\begin{aligned}
\mathbf{F}(x,0) &= \mathbf{f}_1(x) + (1-g_1(x))\,\mathbf{f}_4(1) + g_1(x)\,\mathbf{f}_2(0) \\
&\quad -\mathbf{f}_1(0)(1-g_1(x))\,\underbrace{(1-g_2(y))}_{=1} - \mathbf{f}_2(0)\,g_1(x)\,\underbrace{(1-g_2(y))}_{=1} \\
&\quad -\mathbf{f}_4(0)\,(1-g_1(x))\,\underbrace{g_2(y)}_{=0} - \mathbf{f}_3(0)\,g_1(x)\,\underbrace{g_2(y)}_{=0} \\
&= \mathbf{f}_1(x) + (1-g_1(x))\,\underbrace{(\mathbf{f}_4(1) - \mathbf{f}_1(0))}_{=0} + g_1(x)\,\underbrace{(\mathbf{f}_2(0) - \mathbf{f}_2(0))}_{=0} = \mathbf{f}_1(x)
\end{aligned}$$

and, with precisely the same computations,

$$\mathbf{F}(x,1) = \mathbf{f}_3(1-x), \qquad \mathbf{F}(0,y) = \mathbf{f}_4(1-y), \qquad \mathbf{F}(1,y) = \mathbf{f}_2(y).$$

We summarize these observations in the following theorem.

Theorem 5.30 (Coons patch) *Given four curves $\mathbf{f}_j : [0,1] \to \mathbb{R}^3$ that satisfy* (5.42) *and two **blending functions** $g_1, g_2 : [0,1] \to \mathbb{R}$ with $gj(0) = 0$, $g_j(1) = 1$, the **Boolean sum***

$$\mathbf{F}(x) = \mathbf{F}_1(x,y) \oplus \mathbf{F}_2(x,y) := \mathbf{F}_1(x,y) + \mathbf{F}_2(x,y) - \mathbf{G}(x,y), \tag{5.45}$$

with $\mathbf{F}_1, \mathbf{F}_2$ and $\mathbf{G}$ defined in (5.43) *and* (5.44) *is called the **Coons patch** and has $\mathbf{f}_j$ as boundary curves:*

$$\mathbf{F}(x,0) = \mathbf{f}_1(x), \quad \mathbf{F}(x,1) = \mathbf{f}_3(1-x), \quad \mathbf{F}(0,y) = \mathbf{f}_4(1-y), \quad \mathbf{F}(1,y) = \mathbf{f}_2(y).$$

The strength of the Coons patches lies in the flexibility provided by the blending functions g_1, g_2. It even allows to model different types of blending in x- and y–direction, but normally the first choice is a symmetric one, namely $g_1 = g_2 = g$.

Example 5.31 (Coons patches) *The two most prominent examples of Coons patches are*

1. ***bilinear Coons patches** where $g_1(x) = x$. Here $\mathbf{F}_1$ and $\mathbf{F}_2$ are ruled surfaces formed from the boundary curves and $\mathbf{G}$ is the **bilinear interpolant** of the corners. The function g obviously satisfies*

$$g(0) = 0, \qquad g(1) = 1, \qquad g'(0) = g'(1) = 1.$$

2. ***bicubic Coons patches** where*

$$g(x) = x^2(3-2x);$$

this function satisfies

$$g(0) = 0, \qquad g(1) = 1, \qquad g'(0) = g'(1) = 0,$$

*hence it is a **sigmoidal function**.*

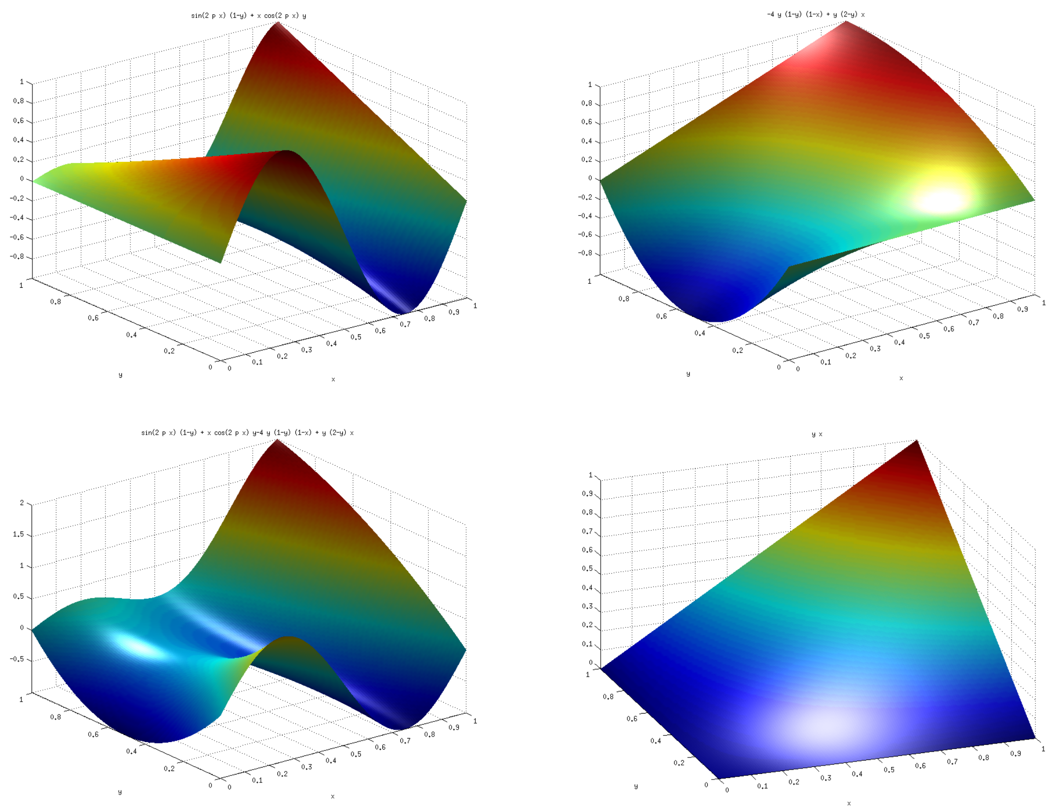

Figure 5.4: Example for a bilinearly blended Coons patch. On top the two ruled surfaces F_1 *(left)* and F_2 *(right)* and on the bottom the surfaces $F_+ = F_1 + F_2$ *(left)* and the bilinear interpolant G *(right)*. The resulting Coons patch is then shown in Fig. 5.5 *(left)*. All figures were created by `Matlab`.

The reason why the bicubic Coons patch is so popular is due to the fact that there is a certain control of the **cross boundary derivatives** of the patch at the boundary. To that end, let us consider

$$\begin{aligned}\frac{\partial F}{\partial y}(x,0) &= \frac{\partial}{\partial x}\left(F_1(x,y) + F_2(x,y) - G(x,y)\right)(x,0)\\ &= \underbrace{-g_2'(0)\,\mathbf{f}_1(x) + g_2'(0)\,\mathbf{f}_3(1-x)}_{F_1}\underbrace{-(1-g_1(x))\mathbf{f}_4'(1) + g_1(x)\mathbf{f}_2'(0)}_{F_2}\\ &\quad + \underbrace{g_2'(0)\left((1-g_1(x))\mathbf{f}_1(0) + g_1(x)\,\mathbf{f}_2(0) - (1-g_1(x))\mathbf{f}_3(0) - g_1(x)\mathbf{f}_4(0)\right)}_{G}\end{aligned}$$

which vanishes whenever $g_2'(0) = 0$, for example, in the case of bicubically blended patches. Similar computations hold for the other three parts of the boudary curve, hence, the surface is **flat** across the boundary. In general, the blending functions g_1 and g_2 can be chosen arbitrarily and this can be used to obtain different blends.

One application of Coons patches is approximation of quadrilateral surface

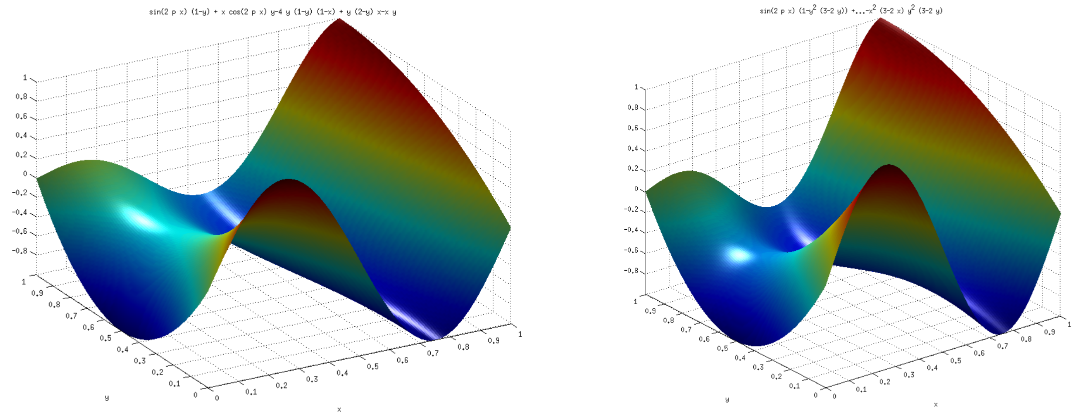

Figure 5.5: The final Coons patch with the ingredients from Fig. 5.4 *(left)* and its bicubically blended cousin *(right)*.

networks: given a set $\mathbf{y}_\alpha \in \mathbb{R}^3$, $\alpha \le \nu$, one can interpolate along the x- and y–lines, for example with a cubic natural spline as in Section 4.3.5 to obtain a set of quadrilaterals bounded by curves. These can be blended by the above method into an overall surface whose smoothness can be controlled by a proper choice of the function g. The advantage of such a method is that it needs far less points than a tensor product bicubic natural spline.

5.4.2 Gordon patches

The concept of **Gordon patches** generalizes that of a Coons patch by interpolating **isoparametric curves** or **isocurves**[116]. An isocuve $\mathbf{f}$ for a surface $\mathbf{F}$ is a curve such that there exists x^* or y^* such that

$$\mathbf{F}(x^*, y) = \mathbf{f}(y) \qquad \text{or} \qquad \mathbf{F}(x, y^*) = \mathbf{f}(x). \tag{5.46}$$

The Gordon patch starts with given sites x_j, $j = 0, \dots, m$, as well as y_k, $k = 0, \dots, m$, and respective isocurves $\mathbf{f}_{x,j}$ and $\mathbf{f}_{y,k}$. The goal is to construct a surface $\mathbf{F}$ with

$$\mathbf{F}(x_j, \cdot) = \mathbf{f}_{x,j}, \quad j = 0, \dots, m \qquad \text{and} \qquad \mathbf{F}(\cdot, y_k) = \mathbf{f}_{y,k}, \quad k = 0, \dots, n. \tag{5.47}$$

At the intercetion points the isocurves have to satisfy the **compatibility condition**

$$\mathbf{f}_{x,j}(y_k) = \mathbf{f}_{y,k}(x_j), \qquad j = 0, \dots, m,\ k = 0, \dots, n. \tag{5.48}$$

To build $\mathbf{F}$ from the isocurves we make use of univariate interpolation and pick any set of *scalar* functions $\ell_{x,j}$, $\ell_{y,k}$ such that

$$\ell_{x,j}(x_{j'}) = \delta_{jj'}, \quad j, j' = 0, \dots, m, \qquad \ell_{y,k}(y_{k'}) = \delta_{kk'}, \quad k, k' = 0, \dots, n.$$

[116] The two names are usually used synonymously where "isocurve" sounds cooler, of course.

Canonical choices for such functions would be splines or polyomials but in fact *any* set of such functions would do, but nevertheless it should satisfy the following condition.

Definition 5.32 *A set of functions $\ell_0, \dots, \ell_n$ is called a* ***Lagrange basis***[117] *for $x_0, \dots, x_n$ if*

$$\ell_j(x_k) = \delta_{jk}, \qquad j, k = 0, \dots, n. \tag{5.49}$$

A Lagrange basis is said to be of ***order zero*** *if the interpolation preserves constants or, equivalently, if the functions form a* ***partition of unity****, that is,*

$$\sum_{j=0}^{n} \ell_j(x) = 1. \tag{5.50}$$

Example 5.33 *A simple example for a Lagrange basis that is not of order zero can be constructed as follows: take $n+2$ distinct points and the unique Lagrange basis within the polynomials of degree $n+1$. They are of order zero and each of them is of the form $ax^{n+1} + \cdots$, hence*

$$\sum_{j=0}^{n} \ell_j = 1 - \ell_{n+1} \neq 0.$$

Now we form

$$\mathbf{F}_1(x, y) \quad = \quad \sum_{j=0}^{m} \ell_{x,j}(x)\, \mathbf{f}_{x,j}(y), \tag{5.51}$$

$$\mathbf{F}_2(x, y) \quad = \quad \sum_{k=0}^{n} \ell_{y,k}(y)\, \mathbf{f}_{y,k}(x), \tag{5.52}$$

and obtain our final surface as

$$\mathbf{F}(x, y) = \mathbf{F}_1(x, y) + \mathbf{F}_2(x, y) - \sum_{j,k=0}^{m,n} \mathbf{f}_{x,j}(y_k)\, \ell_{x,j}(x) \ell_{y,k}(y). \tag{5.53}$$

Theorem 5.34 *The* ***Gordon surface*** *from* (5.53) *satisfies* (5.47)*.*

Proof: Essentially by substitution and the compatibility condition (5.48):

$$\begin{aligned}
\mathbf{F}(x_p, y) &= \sum_{j=0}^{m} \underbrace{\ell_{x,j}(x_p)}_{=\delta_{jp}}\, \mathbf{f}_{x,j}(y) + \sum_{k=0}^{n} \ell_{y,k}(y)\, \mathbf{f}_{y,k}(x_p) - \sum_{j,k=0}^{m,n} \mathbf{f}_{x,j}(y_k)\, \underbrace{\ell_{x,j}(x_p)}_{=\delta_{jp}}\, \ell_{y,k}(y) \\
&= \mathbf{f}_{x,p}(y) + \sum_{k=0}^{n} \ell_{y,k}(y)\, \mathbf{f}_{y,k}(x_p) - \sum_{k=0}^{n} \underbrace{\mathbf{f}_{x,p}(y_k)}_{=\mathbf{f}_{y,k}(x_p)}\, \ell_{y,k}(y) \\
&= \mathbf{f}_{x,p}(y) + \sum_{k=0}^{n} \ell_{y,k}(y)\, (\mathbf{f}_{y,k}(x_p) - \mathbf{f}_{y,k}(x_p)) = \mathbf{f}_{x,p}.
\end{aligned}$$

The proof for $\mathbf{F}(x, y_p)$ works exactly the same way. □

[117]The name is derived from the name **Lagrange interpolation** used for interpolation of function values, interpolation of consecutive derivatives is called **Hermite interpolation** and the more general case of "gaps" among the derivatives bears the name '**Birkhoff interpolation**.

Corollary 5.35 *If the blending functions $\ell_{x,j}$ and $\ell_{y,k}$ are of order zero, then the Gordon surface is a translational surface if all the isocurves $\mathbf{f}_{x,j}$ or $\mathbf{f}_{y,k}$ coincide, respectively.*

Remark 5.36 *A **Coons patch** is a **Gordon surface** with $x_0 = y_0 = 0$, $x_1 = y_1 = 1$ and*

$$\mathbf{f}_{x,0} = \mathbf{f}_1, \qquad \mathbf{f}_{x,1} = \mathbf{f}_3(1-\cdot), \qquad \mathbf{f}_{y,0} = \mathbf{f}_4(1-\cdot), \qquad \mathbf{f}_{y,1} = \mathbf{f}_2.$$

Ratio fatum vincere nulla valet.

Ovid

6 Rational curves and surfaces

The standard freeform object in CAD nowaydays, also integrated in every geometric file standard like **IGES** or **Step**, are **NURB**s. This acronym is the abbreviation for "**N**on **U**niform **R**rational **B**–spline" and means either curves or tensor product surfaces generated from these curves.

6.1 Rational functions

Definition 6.1 *A **rational function** $f = \frac{p}{q}$ is the quotient of two polynomials $p \in \Pi_m$, $q \in \Pi_n$. We denote all rational functions of that type by*[118] $\mathscr{R}_{m,n}$.

Remark 6.2 *Since of $p/q, \tilde{p}/\tilde{q} \in \mathscr{R}_{m,n}$ we have*

$$\frac{p}{q} + \frac{\tilde{p}}{\tilde{q}} = \frac{p\tilde{q} + \tilde{p}q}{q\tilde{q}}$$

which usually does not belong to $\mathscr{R}_{m,n}$ any more, this space is neither linear[119] *and not even convex. This makes **rational approximation**, for example, a totally different field, see (Braess, 1986).*

Life becomes significantly easier if the rational functions to be considered are from the space

$$\mathscr{R}_{m,n} \supset \mathscr{R}_{m,q} = \frac{\Pi_n}{q} = \left\{ \frac{p}{q} : p \in \Pi_m \right\}, \qquad q \in \Pi_n,$$

which almost trivially is a linear space. We can represent the numerator and the denominator by means of the Bernstein–Bézier basis, assume that the have the same degree[120] and obtain the following definition of a rational curve.

[118]Notationally, a greek letter would be almost more appropriate but since the proper one would be "Rho", written as P, this would lead to even more confusion. Therefore $\mathscr{R}$ is appropriate, in particular since this set has a totally different structure.
[119]Which means that $\mathscr{R}_{m,n}$ is **no** vector space.
[120]Otherwise we "artificially" write the polynomial of smaller degree in terms of the higher degree.

Definition 6.3 *A **rational curve** on* $[0,1]$ *is given as*

$$R_n(\mathbf{c}, w) = \frac{B_n \mathbf{c}}{B_n w} = \frac{\sum_{j=0}^{n} \mathbf{c}_j \, B_j^n(u)}{\sum_{j=0}^{n} w_j \, B_j^n(u)}, \tag{6.1}$$

where

$$B_j^n(u) := B_{(n-j),j}(u) = \binom{n}{j} (1-u)^{n-j} u^j. \tag{6.2}$$

The coefficients $\mathbf{c}_j$ *are called **control points** again, the* w_j *are called the **weights** of the rational curve.*

Remark 6.4 *The notion of a rational curve can be extended to a rational triangular surface in a very straightforward manner. Just keep in mind that the numerator function can be vector valued while the denominator is always scalar. The representation* (6.1) *is unique up to a common nonzero factor of the coefficients.*

To efficiently compute with rational curves, we can define

$$\widehat{\mathbf{c}_j} := \begin{bmatrix} w_j \\ \mathbf{c}_j \end{bmatrix} \in \mathbb{R}^{d+1}, \qquad j = 0, \dots, n,$$

and then evaluate the **polynomial curve**

$$B_n \widehat{\mathbf{c}}(u) = \sum_{j=0}^{n} \begin{bmatrix} w_j \\ \mathbf{c}_j \end{bmatrix} B_j^n(u) = \begin{bmatrix} B_n w(u) \\ B_n \mathbf{c}(u) \end{bmatrix} =: \widehat{\mathbf{p}}(u)$$

from which the rational curve is obtained as

$$\mathbf{r}(u) = \frac{\mathbf{p}(u)}{p_0(u)} \equiv \begin{bmatrix} 1 \\ \dfrac{p_1(u)}{p_0(u)} \\ \vdots \\ \dfrac{p_d(u)}{p_0(u)} \end{bmatrix}$$

This concept, however, is very well known in Mathematics, it is the **projective space** from Definition 2.12.

Remark 6.5 *This is the main concept for rational curves and surfaces in* $\mathbb{R}^d$*: Embed them into* $\mathbb{R}^{d+1}$*, use the standard affine algorithms there and divide by the additional component. Note, however, that the coefficients* $\widehat{\mathbf{c}_j}$ *do not form a linear space,*

$$\widehat{\mathbf{c}_j} + \widehat{\mathbf{b}_j} = \begin{bmatrix} w_j \\ \mathbf{c}_j \end{bmatrix} + \begin{bmatrix} v_j \\ \mathbf{b}_j \end{bmatrix} = \begin{bmatrix} v_j + w_j \\ \mathbf{b}_j + \mathbf{c}_j \end{bmatrix}$$

does not make sense, at least not if it is to be interpreted as the sum of the two rational curves in $\mathbb{R}^d$.

Lemma 6.6 *Two rational curves can be added if*[121] *they have the same weights and the resulting curve is*

$$R_n(\mathbf{c}, w) + R_n(\mathbf{c}', w) = R_n(\mathbf{c} + \mathbf{c}', w).$$

Exercise 6.1 Formulate and prove the rational de Casteljeau algorithm. ◇

Rational functions can have "bad points".

Definition 6.7 *A* ***pole***[122] *of a rational function $f = \frac{p}{q}$ is a zero of the denominator. A* ***removable pole*** *is a zero of the denominator that is also a zero of the numerator of at least the same multiplicity*[123].

In the projective terminology a pole of the curve has a nice interpretation as it is a point where

$$B_n\widehat{\mathbf{c}}(u) = \left[\begin{array}{c} 0 \\ B_n\mathbf{c}(u) \end{array} \right],$$

which is one of the many representations of the point ∞ in $\mathbb{P}^d$.

In contrast to complex analysis[124] , poles in curves are not desirable in practical applications and should be avoided by proper choice of the weights. The standard choice is to set

$$w_j \geq 0, \qquad w_0\, w_n > 0, \qquad \sum_{j=0}^{n} w_j = 1. \tag{6.3}$$

The last condition in (6.3) is only a normalization, the other two are the significant ones.

Lemma 6.8 *If the weights satisfy* (6.3), *the rational curve* $R_n(\mathbf{c}, w)$ *has no poles in* $[0, 1]$.

Proof: Due to the endpoint interpolation (4.10), the weight function in the denominator satisfies

$$B_n w(0) = w_0 > 0, \qquad B_n w(1) = w_n > 0,$$

while for arbitrary $u \in (0, 1)$ we get

$$B_n w(u) = \sum_{j=0}^{n} \underbrace{w_j}_{\geq 0} \underbrace{B_j^n(u)}_{\geq 0} \geq w_0 \underbrace{(1-u)^n}_{>0} + w_n \underbrace{u^n}_{>0} > 0,$$

hence $B_n w(u) > 0$, $u \in [0, 1]$, and the curve has no pole. □

---

[121] We are *not* talking about "only if" here, but if the condition is not satisfied, the addition is more complicated, of course:

$$\frac{B_n\mathbf{c}}{B_n w} + \frac{B_n\mathbf{c}'}{B_n w'} = \frac{B_n w\, B_n\mathbf{c}' + B_n w'\, B_n\mathbf{c}}{B_n w\, B_n w'} \in \mathscr{R}_{2n,2n},$$

and even though all the quantities can be computed, this is not a desirable behavior.

[122] In CAD terminology, the word *pole* is also used for the control points which helps to increase the amount of confusion.

[123] Removable poles are always artificial as they could be divided off.

[124] Aka "**Funktionentheorie**"

6.2 Ratios, cross ratios and projections

In this section we follow (Farin, 1988) and introduce some *geometric* invariants of projective maps that we will need to define and manage our rational curves.

Definition 6.9 *The **ratio** of three **collinear points** $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^d$ is defined as*

$$r(\mathbf{a}, \mathbf{b}, \mathbf{c}) := \frac{u_1(\mathbf{b}|[\mathbf{a}, \mathbf{c}])}{u_0(\mathbf{b}|[\mathbf{a}, \mathbf{c}])} = \frac{\mathrm{vol}_1([\mathbf{a}, \mathbf{b}])}{\mathrm{vol}_1([\mathbf{b}, \mathbf{b}])}, \tag{6.4}$$

*and the **cross ratio** of four collinear points $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^d$ as*

$$c(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) := \frac{r(\mathbf{a}, \mathbf{b}, \mathbf{d})}{r(\mathbf{a}, \mathbf{b}, \mathbf{c})}. \tag{6.5}$$

Remark 6.10 *Since **barycetric coordinates** are invariant under any **affine transformation**, so is the ratio of three collinear points.*

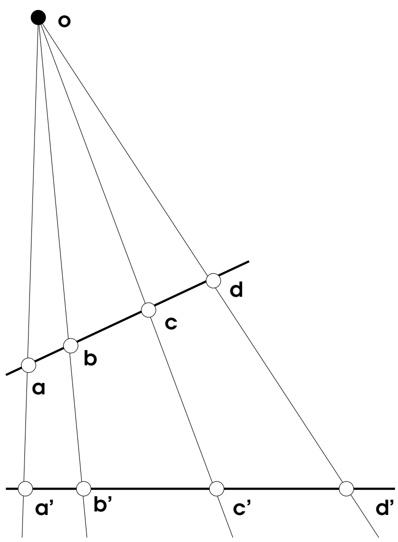

Figure 6.1: Example of a projective map on the line through $\mathbf{a}, \mathbf{b}, c\mathbf{b}, \mathbf{d}$ with the center $\mathbf{o}$. The image point $\mathbf{a}$ of $\mathbf{a}'$, for example, is obtained by connecting $\mathbf{a}'$ with $\mathbf{o}$ and then form the intersection between this connection and the "target" line.

Like (Farin, 1988), we will not give a formal definition of the projection on a straight line, but use the "picture definition" of Fig 6.1. It can, however, be shown, that projective maps can be written as *rational linear transformations*. One can see in this figure that obviously ratios are not preserved but the following theorem shows that cross ratios are.

Theorem 6.11 (Cross ration theorem) *For the points in Fig 6.1 we have*

$$c(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) = c(\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}'). \tag{6.6}$$

Proof: Interpreting the 1-d barycentric coordinates as 2-d coordinates, we get that

$$u_0(\mathbf{b}|[\mathbf{a},\mathbf{c}]) = u_0(\mathbf{b}|[\mathbf{a},\mathbf{c},\mathbf{o}]) = \frac{\mathrm{vol}_2([\mathbf{a},\mathbf{b},\mathbf{o}])}{\mathrm{vol}_2([\mathbf{a},\mathbf{c},\mathbf{o}])}$$

and

$$u_1(\mathbf{b}|[\mathbf{a},\mathbf{c}]) = u_1(\mathbf{b}|[\mathbf{a},\mathbf{c},\mathbf{o}]) = \frac{\mathrm{vol}_2([\mathbf{b},\mathbf{c},\mathbf{o}])}{\mathrm{vol}_2([\mathbf{a},\mathbf{c},\mathbf{o}])},$$

hence

$$r(\mathbf{a},\mathbf{b},\mathbf{d}) = \frac{\dfrac{\mathrm{vol}_2([\mathbf{b},\mathbf{d},\mathbf{o}])}{\mathrm{vol}_2([\mathbf{a},\mathbf{d},\mathbf{o}])}}{\dfrac{\mathrm{vol}_2([\mathbf{a},\mathbf{b},\mathbf{o}])}{\mathrm{vol}_2([\mathbf{a},\mathbf{d},\mathbf{o}])}} = \frac{\mathrm{vol}_2([\mathbf{b},\mathbf{d},\mathbf{o}])}{\mathrm{vol}_2([\mathbf{a},\mathbf{b},\mathbf{o}])}.$$

Likewise, we find that

$$r(\mathbf{a},\mathbf{c},\mathbf{d}) = \frac{\mathrm{vol}_2([\mathbf{c},\mathbf{d},\mathbf{o}])}{\mathrm{vol}_2([\mathbf{a},\mathbf{c},\mathbf{o}])},$$

hence

$$c(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}) = \frac{\mathrm{vol}_2([\mathbf{b},\mathbf{d},\mathbf{o}])\,\mathrm{vol}_2([\mathbf{a},\mathbf{c},\mathbf{o}])}{\mathrm{vol}_2([\mathbf{a},\mathbf{b},\mathbf{o}])\,\mathrm{vol}_2([\mathbf{c},\mathbf{d},\mathbf{o}])}.$$

By the elementary area formula for triangles[125], we get, with

$$\alpha = \angle(\mathbf{a},\mathbf{o},\mathbf{b}), \qquad \beta = \angle(\mathbf{b},\mathbf{o},\mathbf{c}), \gamma = \angle(\mathbf{c},\mathbf{o},\mathbf{d}),$$

that[126]

$$c(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}) \;=\; \frac{(\ell_{\mathbf{b}}\ell_{\mathbf{d}}\sin(\beta+\gamma))\,(\ell_{\mathbf{a}}\ell_{\mathbf{c}}\sin(\alpha+\beta))}{(\ell_{\mathbf{a}}\ell_{\mathbf{b}}\sin\alpha)\,(\ell_{\mathbf{c}}\ell_{\mathbf{d}}\sin\gamma)} = \frac{\sin(\beta+\gamma)\,\sin(\alpha+\beta)}{\sin\alpha\,\sin\gamma},$$

and since this expression depends only of the angles between the **projection lines** at $\mathbf{o}$, it is the same for $c(\mathbf{a}',\mathbf{b}',\mathbf{c}',\mathbf{d}')$. □

Cross ratios are interesting objects by themselves and have even been used in the construction of musical instruments. There exists a construction by Stråhle (Stråhle, 1743) for how to place the positions for an approximately tempered scale by means of a geometric construction based on projections, see Fig. 6.2. The story about this construction, its incorrect "correction" by the Swedish mathematician Faggot and what all this has to do with continued fractions, is nicely told in I. Steward's article *Faggot's fretful fiasco* in (Fauvel *et al.*, 2003). The trick is to get the angle right, but also that the cross ratio between the fret distances (or also tone hole distances for woodwind instruments) is always constant due to Theorem 6.11. This has nothing to do with splines directly except that there is also a paper by Schoenberg on how to place the frets on guitars.

[125]The area is $\frac{1}{2}\ell_1\ell_2\sin\alpha$, where α is one angle in the triangle and ℓ_1, ℓ_2 are the lengths of the adjacent edges.

[126]We immediately cancel all the $\frac{1}{2}$ terms.

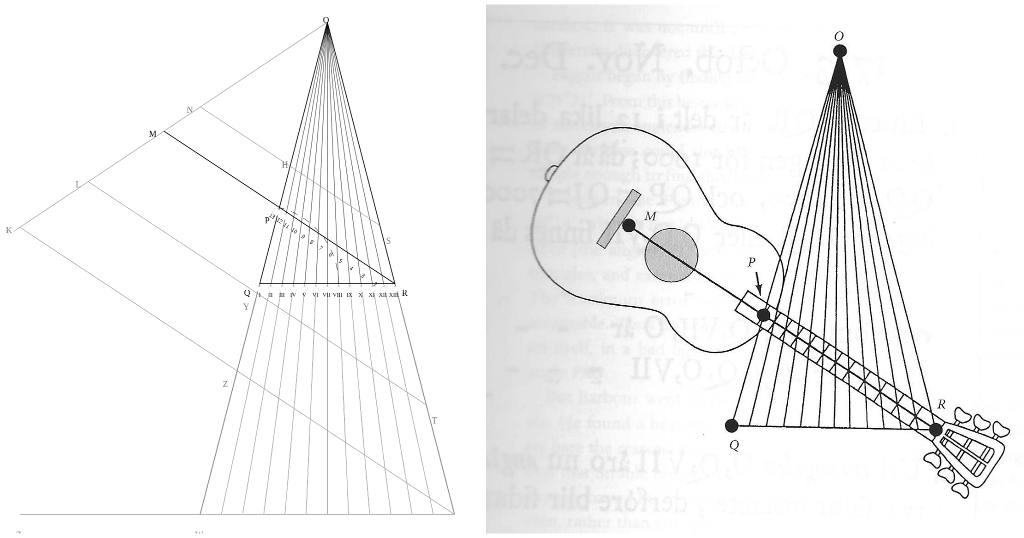

Figure 6.2: The scheme of Straehles construction and the explanation in the article from (Fauvel *et al.*, 2003). Left image from Wikipedia, right image from (Fauvel *et al.*, 2003).

6.3 Conics

Conics are fundamental objects in CAD as they include classical geometric objects like circles, ellipses[127], parabolas and hyperbolas. Clasically, the are defined as intersections of a cone[128] with a plane and they are categorized depending on how this intersection takes place. For our purposes, the following equivalent definition will be more handy, however.

Definition 6.12 *A **conic section**[129] is the projection of a parabola from $\mathbb{R}^3$ to $\mathbb{R}^2$.*

In particular, conics include circles, more precisely, parts of circles. Just keep in mind that circles as well as parabolas are obtained by intersections of the cone with a plane, then the projection of one of them with the tip of the cone as projection center yields the other one.
To see that this definition of conics with rational quadratics, hence a special class of rational Bézier curves, we recall that a point in $\mathbb{P}^2$ is the equivalence class of the point $\mathbf{x} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \in \mathbb{R}^3$, given as

$$\left\{ \widehat{\mathbf{x}} = \begin{bmatrix} w \\ w\,\mathbf{x} \end{bmatrix} : w \in \mathbb{R} \setminus \{0\} \right\}.$$

Any such element on that line through the origin and $\begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}$ in $\mathbb{R}^3$ is a representer of the same projective point.

[127]OK, a circle is just a special case of an ellipse.
[128]More precisely, the "double cone".
[129]Which may be only a part of a conic.

Theorem 6.13 *For any conic section* $t \mapsto \mathbf{f}$, $t \in [0,1]$, *there exist coefficients* $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2$ *and weights* w_0, w_1, w_2, *not all of them equal to zero, such that*

$$\mathbf{f}(t) = \frac{\sum_{j=0}^{2} \mathbf{c}_j \, B_j^2(t)}{\sum_{j=0}^{2} w_j \, B_j^2(t)}. \tag{6.7}$$

Proof: Each image point of the conic function $\mathbf{f}(t) \in \mathbb{R}^2$ can be lifted to $\begin{bmatrix} 1 \\ \mathbf{f}(t) \end{bmatrix}$ which is the projection of a point $\begin{bmatrix} w(t) \\ \mathbf{w}(t)\mathbf{f}(t) \end{bmatrix}$ which lies on a **parabola** in $\mathbb{R}^3$, parametrized as[130]

$$\mathbf{p}(t) = \begin{bmatrix} w(t) \\ \mathbf{q}(t) \end{bmatrix}, \qquad t \in [0,1],$$

where w and $\mathbf{q}$ are quadratic polynomials. In particular, there exist w_0, w_1, w_2 such that

$$w(t) = \sum_{j=0}^{2} w_j \, B_j^2(t), \qquad t \in [0,1].$$

Since $\mathbf{q}$ is a parabola, this implies that

$$w(t) \begin{bmatrix} 1 \\ \mathbf{f}(t) \end{bmatrix} = \mathbf{p}(t) = \sum_{j=0}^{2} \begin{bmatrix} w_j \\ \mathbf{c}_j \end{bmatrix} B_j^2(t),$$

hence

$$\mathbf{f}(t) = \frac{\sum_{j=0}^{2} \mathbf{c}_j B_j^2(t)}{w(t)},$$

which is (6.7). □

Corollary 6.14 *If all weights are nonzero, the conic can be written as*

$$\mathbf{f}(t) = \frac{\sum_{j=0}^{2} w_j \mathbf{b}_j \, B_j^2(t)}{\sum_{j=0}^{2} w_j \, B_j^2(t)}. \tag{6.8}$$

[130] That the parametrization runs over $[0,1]$ can always be ensured by an **affine reparametrization** which transforms polynomials to polynomials *of the same degree*.

Remark 6.15 *The advantage of the the representation* (6.8) *lies in the fact that the 3 dimensional representations*

$$\begin{bmatrix} w_j \\ w_j\,\mathbf{b}_j \end{bmatrix}, \qquad j = 0,1,2,$$

are all projectively equivalent to the point $\begin{bmatrix} 1 \\ \mathbf{b}_j \end{bmatrix} \sim \mathbf{b}_j$.

Every conic can also be written in **implicit** form as

$$\{\mathbf{x} \in \mathbb{R}^2 : f(\mathbf{x}) = 0\}, \qquad f \in \Pi_2,$$

where f is a quadratic polynomial. Implicit forms are useful for intersections and for checking whether a point lies on a conic, but the implicit form cannot distinguish between the "full" conic and some conic section that forms a part of it.

Example 6.16 *The explicit form for a circle with center* $\mathbf{c}$ *and radius* r *is*

$$f(\mathbf{x}) = \|\mathbf{x} - \mathbf{c}\|_2^2 - r^2 = (x_1 - c_1)^2 + (x_2 - c_2)^2 - r^2.$$

The implicit form of a conic is now easily determined for a nondegenerate conic, i.e., a conic that is not a straight line. This in turn is equivalent to $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ being in general position and forming a nondegenerate triangle. In other words,

$$\det \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 \end{bmatrix} \neq 0.$$

Now we write the point in terms of barycentric coordinates with respect this triangle as

$$\mathbf{f}(t) = \sum_{j=0}^{2} u_j(t)\,\mathbf{b}_j, \qquad \text{i.e.,} \qquad u_j(t) = w_j \frac{B_j^2(t)}{w(t)},$$

that is,

$$\begin{aligned} w(t)u_0(t) &= w_0(1-t)^2 \\ w(t)u_1(t) &= 2w_1 t(1-t) \\ w(t)u_1(t) &= w_1 t^2 \end{aligned}$$

If we square the middle equation and substitute the other two, this gives

$$w^2(t) w_1^2 u_1^2(t) = 4\,w(t) w_0 u_0(t)\,w(t) w_2 u_2(t) \qquad \Leftrightarrow \qquad w_1^2 u_1^2 - 4 w_0 w_2\, u_0 u_2 = 0$$

and the explicit formula for barycentric coordinates,

$$u_0 = \frac{\det \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{x} & \mathbf{b}_1 & \mathbf{b}_2 \end{bmatrix}}{\det \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 \end{bmatrix}}, \quad u_1 = \frac{\det \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{b}_0 & \mathbf{x} & \mathbf{b}_2 \end{bmatrix}}{\det \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 \end{bmatrix}}, \quad u_2 = \frac{\det \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{x} \end{bmatrix}}{\det \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 \end{bmatrix}},$$

yields the implicit formula

$$f(\mathbf{x}) = w_1^2 \det \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{b}_0 & \mathbf{x} & \mathbf{b}_2 \end{bmatrix}^2 - 4w_0w_2 \det \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{x} & \mathbf{b}_1 & \mathbf{b}_2 \end{bmatrix} \det \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{x} \end{bmatrix}, \quad (6.9)$$

which is a quadratic polynomial in $\mathbf{x}$.

Example 6.17 *We want to determine the circular segment with* $\mathbf{b}_0 = (1,0)$, $\mathbf{b}_1 = (1,1)$, $\mathbf{b}_2 = (0,1)$, *hence* $\mathbf{c} = 0$ *and* $r = 1$. *Since*

$$\det \begin{bmatrix} 1 & 1 & 1 \\ 1 & x & 0 \\ 0 & y & 1 \end{bmatrix} = x+y-1, \quad \det \begin{bmatrix} 1 & 1 & 1 \\ x & 1 & 0 \\ y & 1 & 1 \end{bmatrix} = 1-y, \quad \det \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & x \\ 0 & 1 & y \end{bmatrix} = (1-x),$$

we get that

$$\begin{aligned} f(x,y) &= w_1^2(x+y-1)^2 - 4w_0w_1(1-x)(1-y) \\ &= w_1^2x^2 + w_1^2y^2 + (2w_1^2 - 4w_0w_1)xy - (2w_1^2 - 4w_0w_1)(x+y) + w_1^2 - 4w_0w_1, \end{aligned}$$

which becomes the implicit equation $x^2 + y^2 - 1$ *if and only if* $w_1^2 = 1$, *say* $w_1 = 1$, *and*

$$2 - 4w_0w_1 = 0 \qquad \text{and} \qquad 1 - 4w_0w_1 = -1.$$

The second reuqirement follows directly from the first and we only have to choose $w_0w_1 = \frac{1}{2}$ *which is symmetrically chosen as* $w_0 = w_1 = \frac{1}{\sqrt{2}}$. *Hence, the rational control points for the* exact ***quarter circle*** *are*

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

6.4 Rational Bézier und spline curves

Let us give a formal definition of rational Bézier curves.

Definition 6.18 (Rational Bézier curve) *A* ***rational Bézier curve*** *of degree* n *with* ***weights*** $w_j \neq 0$ *and* ***control points*** $\mathbf{b}_j \in \mathbb{R}^d$ *is the curve*

$$B_n\widehat{\mathbf{b}}(t) := \frac{\sum_{j=0}^{n} w_j \mathbf{b}_j \, B_j^n(t)}{\sum_{j=0}^{n} w_j B_j^n(t)}, \qquad \widehat{\mathbf{b}}_j := \begin{bmatrix} w_j \\ w_j \mathbf{b}_j \end{bmatrix}. \quad (6.10)$$

Such a curve is called ***polynomial***[131] *if* $w_j = 1$, $j = 0, \dots, n$.

[131] Often it falsely called "nonrational" or even "irrational", but such curves only belong to a particular **subclass** of rational functions

Remark 6.19

1. *In* (6.10)*, we defined the control points to be the affine representer of the projective equivalence class.*
2. *The only restriction on the weight is that they are nonzero. Positive or negative weights are still possible, but weights with different signs bear the danger of poles.*
3. *The weights can be normalized such that they sum to* 1 *or* 0*. This is not necessary, just a matter of convention.*

6.4.1 Properties of rational Bézier curves

Some properties carry over almost directly from the Bézier case. Since $B_j^n(0) = \delta_{j0}$, we find, for example, that

$$B_n \widehat{\mathbf{b}}(0) = \frac{w_0 \mathbf{b}_0}{w_0} = \mathbf{b}_0,$$

so that also rational curves provide **endpoint interpolation**. Since, for $w_j > 0$ we have that[132]

$$\frac{w_j B_j^n(t)}{\sum_{j=0}^n w_j B_j^n(t)} \geq 0 \qquad \text{and} \qquad \sum_{j=0}^n \frac{w_j B_j^n(t)}{\sum_{j=0}^n w_j B_j^n(t)} = 1,$$

every point on the curve is again a **convex combination** of the points $\mathbf{b}_j$, yielding the **convex hull property** again: the curve runs within the convex hull of the **control points**.

Derivatives are a bit more complicated. In fact,

$$\frac{d}{dt}\frac{\mathbf{p}(t)}{w(t)} = \frac{\mathbf{p}'(t) w(t) - \mathbf{p}(t) w'(t)}{w^2(t)}$$

yields that

$$B_n \widehat{\mathbf{b}}'(t) = n \, \frac{\left(\sum_{j=0}^{n-1} \Delta(w_j \mathbf{b}_j)\, B_j^{n-1}(t)\right)\left(\sum_{j=0}^{n} w_j B_j^{n-1}(t)\right) - \left(\sum_{j=0}^{n} w_j \mathbf{b}_j \, B_j^n(t)\right)\left(\sum_{j=0}^{n-1} \Delta w_j B_j^{n-1}(t)\right)}{\left(\sum_{j=0}^{n} w_j B_j^{n-1}(t)\right)^2},$$

which is not so nice any more, but at the end points we get

$$\begin{aligned} B_n \widehat{\mathbf{b}}'(0) &= \frac{(w_1 \mathbf{b}_1 - w_0 \mathbf{b}_0) w_0 - w_0 \mathbf{b}_0 (w_1 - w_0)}{w_0^2} \\ &= \frac{w_0 w_1 \mathbf{b}_1 - w_0^2 \mathbf{b}_0 - w_0 w_1 \mathbf{b}_0 + w_0^2 \mathbf{b}_0}{w_0^2} = \frac{w_1}{w_0} \Delta \mathbf{b}_0, \end{aligned}$$

so that the geometric interpretation of the end segments of the control polygon as tangents persists as well.

[132] In fact, this holds true as long as all w_j have the same sign.

Remark 6.20 *All these results and formulas show that the "projective" definition from* (6.10) *is indeed the right one.*

Literatur | 6

Bauschinger, J. (1900). Interpolation. In *Encyklopädie der Mathematischen Wissenschaften, Bd. I, Teil 2*, pages 800–821. B. G. Teubner, Leipzig.

Bernstein, S. N. (1912). Démonstration du théorème de Weierstrass, fondée su le calcul des probabilitiés. *Commun. Soc. Math. Kharkov*, **13**:1–2.

Bézier, P. (1972). *Numerical Control. Mathematics and Applications*. J. Wiley and Sons.

Bézier, P. (1986). *The mathematical basis of the UNISURF CAD system*. Butterworth & Co Ltd.

Boor, C. d. (1972). On calculating with B–splines. *J. Approx. Theory*, **6**:50–62.

Boor, C. d. (1979a). Efficient computer manipulation of tensor products. *ACM Transactions on Mathematical Software*, **5**:173–182.

Boor, C. d. (1979b). Efficient computer manipulation of tensor products, corrigenda. *ACM Transactions on Mathematical Software*, **5**:535.

Boor, C. d. (1990). *Splinefunktionen*. Lectures in Mathematics, ETH Zürich. Birkhäuser.

Braess, D. (1986). *Nonlinear Approximation Theory*, volume 7 of *Springer Series in Computational Mathematics*. Springer.

Brieskorn, E. (1985). *Lineare Algebra und Analytische Geometrie II*. Vieweg.

Curry, H. B., Schoenberg, I. J. (1966). On Pólya frequency functions IV: The fundamental spline functions and their limits. *J. d'Analyse Math.*, **17**:71–107.

Dinghas, A. (1951). Über einige Identitäten vom Bernsteinschen Typus. *Det Koneglige Norske Videnskabers Selskab*, **24**(21).

Farin, G. (1988). *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press.

Fauvel, J., Flood, R., Wilson, R., editors (2003). *Music and Mathematics. From Pythagoras to Fractals*. Oxford University Press.

Fischer, G. (1984). *Lineare Algebra*. Vieweg.

Forster, O. (1976). *Analysis I*. Vieweg.

Gasca, M., Sauer, T. (2000). On the history of multivariate polynomial interpolation. *J. Comput. Appl. Math.*, **122**:23–35.

Gathen, J. v. z., Gerhard, J. (1999). *Modern Computer Algebra*. Cambridge University Press.

Gelfand, I. M., Fomin, S. V. (1963). *Calculus of Variations*. Prentice–Hall. Dover reprint, 2000.

Golub, G., van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, 3rd edition.

Halmos, P. (1988). *I want to be a mathematician. An automathography*. MAA Spectrum Series. Mathematical Association of America.

Hamm, C., Handeck, J., Sauer, T. (2014). Spline multiresolution and wavelet–like decompositions. *Comp. Aided Geom. Design*, **31**:521–530.

Heuser, H. (1983). *Lehrbuch der Analysis. Teil 2*. B. G. Teubner, 2. edition.

Higham, N. J. (2002). *Accuracy and stability of numerical algorithms*. SIAM, 2nd edition.

Horn, R. A., Johnson, C. R. (1991). *Topics in Matrix Analysis*. Cambridge University Press.

Jordan, M. C. (1887). *Cours d' Analyse. Tome Troisiéme. Calcul Intégral, Équations Différentielles*. Gauthier–Villars.

Kirk, D. E. (1970). *Optimal Control Theory. An Introduction*. Prentice Hall. Dover reprint 2004.

Kreyszig, E. (1959). *Differential Gometry*. The University of Toronto Press. Dover reprint 1991.

Lamping, F., Peña, J. M., Sauer, T. (2015). Kronecker products and multilinear forms. *Numer. Linear Algebra Appl.*. submitted for publication.

Lorentz, G. G. (1953). *Bernstein Polynomials*. University of Toronto Press.

Lyche, T. (1987). *Knot Insertion and Deletion Algorithms for B-Spline Curves and Surfaces*. SIAM.

Marcus, M., Minc, H. (1965). *Introduction to Linear Algebra*. Macmillan, New York. Dover reprint 1989.

Marcus, M., Minc, H. (1969). *A Survey of Matrix Theory and Matrix Inequalities*. Prindle, Weber & Schmidt. Paperback reprint, Dover Publications, 1992.

Meyer, T., Steinthal, H., editors (1973). *Grund- und Aufbauwortschatz Griechisch*. Ernst Klett Verlag.

Möbius, A. F. (1827). *Der barycentrische Calcul*. Johann Ambrosius Barth.

Pogorelov, A. (1987). *Geometry*. Mir Publishers.

Ramshaw, L. (1987). Blossoming: A connect–the-dots approach to splines. Technical report, Digital Systems Research Center, Palo Alto.

Sauer, T. (1996). Ein algorithmischer Zugang zu Polynomen und Splines. *Mathem. Semesterberichte*, **43**( ):169–189. Vortrag im *Eichstätter Kolloquium zur Didaktik der Mathematik*.

Sauer, T. (2001). Computeralgebra. Vorlesungsskript, Justus–Liebig–Universität Gießen, Universität Passau.

Sauer, T. (2013). Einführung in die Numerische Mathematik. Vorlesungsskript, Universität Passau.

Sauer, T. (2014). Analysis 1. Vorlesungsskript, Universit"at Passau.

Sauer, T. (2015). Analysis 2. Vorlesungsskript, Universit"at Passau.

Schneider, H., Barker, G. P. (1973). *Matrices and Linear Algebra*. Holt, Reinehart and Winston. Paperback reprint, Dover Publications, 1989.

Schoenberg, I. J. (1973). *Cardinal Spline Interpolation*, volume 12 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM.

Seidel, H. P. (1989). A new multiaffine approach to B–splines. *Comp. Aided Geom. Design*, **6**:23–32.

Spivak, M. (1965). *Calculus on manifolds*. Perseus Books.

Stengel, R. F. (1986). *Optimal Control and Estimation*. John Wiley & Sons. Dover reprint 1994.

Stråhle, D. P. (1743). Nytt Påfund, til at finna Temperaturen, i stamningen för thonerne på claveret ock dylika Instrumenter. *Proceedings of the Royal Swedish Academy of Sciences*, pages 281–285.

Struik, D. J. (1961). *Lectures on Classical Differential Geometry*. Addison–Wesley, 2nd edition. Dover reprint, 1988.

Van Loan, C. F. (2009). The Kronecker product. A product of times. `http://www.siam.org/meetings/la09/talks/vanloan.pdf`.

Van Loan, C. F., Pitsianis, N. (1993). Approximation with Kronecker products. In Moonen, M. S., Golub, G., editors, *Linear Algebra for Large Scale and Real Time Applications*, pages 293–314. Kluwer.

Votsmeier, M., Scheuer, A., Drochner, A., Vogel, H., Gieshoff, J. (2010). Simulation of automotive NH_3 oxidation catalysts based on pre–computed rate data from mechanistic surface kinetics. *Catalysis Today*, **151**:271–277.