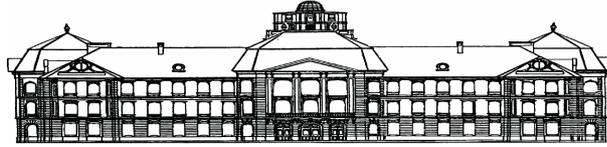


dc_933_14



Eszterházy Károly College
Institute of Mathematics and Computer Science

Miklós Hoffmann

New methods in computer aided modeling of curves and surfaces

Eger, 2014

Contents

1	Knots of B-spline and NURBS curves and surfaces	4
1.1	Basic definitions	4
1.2	Geometric effects of the modification of a knot	5
1.2.1	The envelope of the family of B-spline curves	8
1.2.2	The envelope of the paths of curve points	10
1.2.3	Extension of paths	13
1.2.4	The case of higher order contact and higher multiplicity of knots	14
1.3	Shape control of B-spline and NURBS curves by knot modification	17
1.3.1	B-spline curve passing through a point	17
1.3.2	NURBS curve passing through a point	19
1.3.3	Modification of two weights and a knot value of a NURBS curve	20
1.3.4	Further constrained modification tools by knots	20
1.4	Extension to surfaces	29
2	New curve types in geometric modeling	31
2.1	Linear blending of curves - the quartic curve of Han	31
2.1.1	Definition and linear blending description	32
2.1.2	Linear blending on a common basis	33
2.1.3	Shape parameter alteration	35
2.1.4	Constrained shape modification	38
2.2	C-curves	38
2.2.1	Paths of C-Bézier curves and their extensions	39
2.2.2	Paths of C-B-spline curves and their approximate lines	41
2.3	FB-spline curves	43
2.3.1	Control point based methods	45
2.3.2	Shape parameter based methods	47
2.4	A trigonometric curve with exponential shape parameters	50
2.4.1	New basis functions	51
2.4.2	Nonnegativity	52
2.4.3	Linear independence	55
2.4.4	The curve	62
3	Non-control-point-based methods	65
3.1	Modeling unorganized points by artificial neural networks	65
3.1.1	The Kohonen neural network and its application	66
3.1.2	Numerical control of the training: the gain term and the neighborhood function	70
3.1.3	Ruled surfaces from a set of rulings	72
3.2	Sphere-based modeling	73
3.2.1	Previous work	76

dc_933_14

3.2.2	Skinning of circles	78
3.2.3	Extension to spheres	83
3.2.4	Surfaces with multiple branches	87
3.2.5	Results and comparison with other methods	92
	Bibliography	96

Introduction

New results and methods are discussed in this dissertation in the field of computer aided curve and surface modeling. Three closely related topics are covered by the three chapters, where short overview of the previous works and brief introduction to the topics are also provided. Apart from these introductory notes, further parts of the dissertation contain exclusively new results of the author and his co-authors.

In the first chapter one can find results in terms of knot modification of B-spline and NURBS curves and surfaces, based on the papers [35, 37, 40, 57, 58, 59, 60]. New curve types and their properties are discussed in chapter 2, discussing the contributions provided in [41, 42, 43, 45, 46, 61]. The last chapter provides an overview on non-control-point-based methods, based on the results published in [2, 32, 33, 36, 38, 69, 110].

Here I would like to thank my distinguished colleagues and co-authors for the inspiring atmosphere of our scientific discussions. This dissertation would not have been possible without their help.

1 Knots of B-spline and NURBS curves and surfaces

1.1 Basic definitions

B-spline and NURBS curves are standard description methods and hence widely used in computer aided design today. There are several books and papers on these curves describing their properties, with the help of which one can apply them as powerful design tools. The basic definitions (as one can find e.g. in [90]) are the following:

Definition 1.1. The recursive function N_j^k given by the equations

$$N_j^1(u) = \begin{cases} 1 & \text{if } u \in [u_j, u_{j+1}), \\ 0 & \text{otherwise} \end{cases}$$

$$N_j^k(u) = \frac{u - u_j}{u_{j+k-1} - u_j} N_j^{k-1}(u) + \frac{u_{j+k} - u}{u_{j+k} - u_{j+1}} N_{j+1}^{k-1}(u)$$

is called normalized B-spline basis function of order k (degree $k-1$). The numbers $u_j \leq u_{j+1} \in \mathbb{R}$ are called knot values or simply knots, where $0/0 \doteq 0$ by definition.

Definition 1.2. The curve $\mathbf{s}(u)$ defined by

$$\mathbf{s}(u) = \sum_{l=0}^n \mathbf{d}_l N_l^k(u), \quad u \in [u_{k-1}, u_{n+1}]$$

is called B-spline curve of order k (degree $k-1$), where $N_l^k(u)$ is the l^{th} normalized B-spline basis function, for the evaluation of which the knots u_0, u_1, \dots, u_{n+k} are necessary. The points \mathbf{d}_i are called control points or de Boor-points, while the polygon formed by these points is called control polygon. The arcs of this B-spline curve are called spans. The j^{th} span can be written as

$$\mathbf{s}_j(u) = \sum_{l=j-k+1}^j \mathbf{d}_l N_l^k(u), \quad u \in [u_j, u_{j+1}).$$

As one can observe from the definitions above, B-spline curve is uniquely defined by its degree, control points and knot values, while in terms of NURBS curves the weight vector has to be specified in addition. It is an obvious fact that the modification of each of these data will affect the shape of the curve and some of its geometric properties. The modification of a curve plays central role in CAD systems, hence numerous methods are presented to control the shape of a curve by modifying one of its data mentioned above. The most basic possibilities can be found in any book of the field. Further control point-based shape modification is discussed in [88] and [26], weight-based modification is described e.g. in [88] and [55], while others present shape control by simultaneous modification of control points and weights (see [1], [101]).

The effect of a change of the knot vector on the shape of the curve, however, has not been described yet. Even in one of the most comprehensive books ([90]) one can read the following: "Although knot locations also affect shape, we know of no geometrically intuitive or mathematically simple interpretation of this effect...". The aim of this section is to present the geometrical and mathematical representation of the effects of knot modification for B-spline curves and surfaces, based on the contributions of Hoffmann and Juhász, presented in [35, 37, 40, 57, 58, 59, 60].

Throughout this chapter the following properties of the normalized B-spline basis functions will be used:

1. $N_j^k(u)$ is equal to 0 everywhere except on the interval $[u_j, u_{j+k})$.
2. At the r^{th} step of the recursive evaluation of $N_j^k(u)$ the following functions can occur:

$$N_{j+n}^{k-r}(u), r = 0, \dots, k-1; n = 0, \dots, r.$$

3. $N_j^k(u) = (k-1) \left(\frac{1}{u_{j+k-1}-u_j} N_j^{k-1}(u) - \frac{1}{u_{j+k}-u_{j+1}} N_{j+1}^{k-1}(u) \right)$
4. The modification of the knot u_i affects only the functions $N_{i-k}^k(u), \dots, N_i^k(u)$, hence only the shape of the spans $\mathbf{s}_{i-k+1}(u), \dots, \mathbf{s}_i(u), \dots, \mathbf{s}_{i+k-2}(u)$ of the curve will be changed.

1.2 Geometric effects of the modification of a knot

When modifying the knot u_i , the basis functions and spans described in Property 4 will depend not only on u but on u_i as well. To emphasize this fact, they will be denoted by $N_i^k(u, u_i)$ and $\mathbf{s}_i(u, u_i)$. Fixing the second one of the two variables (i.e. the knot value $u_i = \tilde{u}_i$) one can receive the original basis functions $N_i^k(u, \tilde{u}_i) = N_i^k(u)$ and spans $\mathbf{s}_i(u, \tilde{u}_i) = \mathbf{s}_i(u)$, but fixing the first variable (i.e. the parameter $u = \tilde{u}$) the functions $N_i^k(\tilde{u}, u_i)$ will not remain the standard basis functions any more, but some rational functions of u_i , while $\mathbf{s}_i(\tilde{u}, u_i)$ can be interpreted as a curve on which a point of the original B-spline moves. More precisely, when modifying the knot u_i the point of the span $\mathbf{s}_j(u)$ associated with the fixed parameter value $\tilde{u} \in [u_j, u_{j+1})$ will move along the curve

$$\mathbf{s}_j(\tilde{u}, u_i) = \sum_{l=j-k+1}^j \mathbf{d}_l N_l^k(\tilde{u}, u_i), u_i \in [u_{i-1}, u_{i+1}).$$

Hereafter, we refer to this curve as the path of the point $\mathbf{s}_j(\tilde{u})$. At the first part of this section these functions and paths will be examined, especially in terms of their degree.

Lemma 1.3. $N_{i-k}^k(\tilde{u}, u_i), \tilde{u} \in [u_{i-m}, u_{i-m+1}), (m = 1, \dots, k-1), u_i \in [u_{i-1}, u_{i+1})$ is a rational function of degree $k-m$ in u_i .

Proof. In the recursive Definition 1.1

$$N_{i-k}^k(\tilde{u}, u_i) = \frac{\tilde{u} - u_{i-k}}{u_{i-1} - u_{i-k}} N_{i-k}^{k-1}(\tilde{u}, u_i) + \frac{u_i - \tilde{u}}{u_i - u_{i-k+1}} N_{i-k+1}^{k-1}(\tilde{u}, u_i)$$

the first term is independent of u_i because of Property 4, hence only the second term has to be considered. This fact is also valid for the further steps of the recursion:

$$\begin{aligned} & \vdots \\ N_{i-m-1}^{m+1}(\tilde{u}, u_i) &= \frac{\tilde{u} - u_{i-m-1}}{u_{i-1} - u_{i-m-1}} N_{i-m-1}^m(\tilde{u}, u_i) + \frac{u_i - \tilde{u}}{u_i - u_{i-m}} N_{i-m}^m(\tilde{u}, u_i) \\ N_{i-m}^m(\tilde{u}, u_i) &= \frac{\tilde{u} - u_{i-m}}{u_{i-1} - u_{i-m}} N_{i-m}^{m-1}(\tilde{u}, u_i) + \frac{u_i - \tilde{u}}{u_i - u_{i-m+1}} N_{i-m+1}^{m-1}(\tilde{u}, u_i). \end{aligned}$$

The first term of the right hand side of both equations above is constant because of Property 4. The second term of the last equation is equal to 0 due to Property 1. Thus u_i appears only in $k - m$ terms, at degree 1 everywhere, consequently, the degree of the function $N_{i-k}^k(\tilde{u}, u_i)$ in u_i is $k - m$. \square

Theorem 1.4. *The path $\mathbf{s}_{i-m}(\tilde{u}, u_i) = \sum_{l=i-m-k+1}^{i-m} N_l^k(\tilde{u}, u_i) \mathbf{d}_l$, $u_i \in [u_{i-1}, u_{i+1}]$ is a rational curve of degree $k - m$ with respect to u_i , $\forall \tilde{u} \in [u_{i-m}, u_{i-m+1}]$, ($m = 1, \dots, k - 1$).*

Proof. The lower limit of the summation can be increased to $i - k$, since u_i has no effect on $N_l^k(\tilde{u}, u_i)$ for $l < i - k$ (see Property 4). Hence only the functions

$$N_{i-k+z}^k(\tilde{u}, u_i), z = 0, \dots, k - m \quad (1.1)$$

have to be considered. The function $N_{i-k}^k(\tilde{u}, u_i)$ is of degree $k - m$ because of Lemma 1.3. Thus it is sufficient to prove that the degree of the functions (1.1) is at most $k - m$, for $z > 0$.

At the r^{th} step of the recursion those functions which have influence on the functions mentioned above can be described in the following form (see Property 2):

$$\begin{aligned} N_{i-k+z+n}^{k-r}(\tilde{u}, u_i) &= \frac{\tilde{u} - u_{i-k+z+n}}{u_{i+z+n-r-1} - u_{i-k+z+n}} N_{i-k+z+n}^{k-r-1}(\tilde{u}, u_i) + \\ &\quad \frac{u_{i+z+n-r} - \tilde{u}}{u_{i+z+n-r} - u_{i-k+z+n+1}} N_{i-k+z+n+1}^{k-r-1}(\tilde{u}, u_i) \\ r &= 0, \dots, k - 1; n = 0, \dots, r. \end{aligned}$$

In this form u_i can occur in the following cases:

1. $i - k + z + n = i$, i.e. $z + n - k = 0$, that is the function $N_{i-k+z+n}^{k-r-1}(\tilde{u}, u_i)$ appears in the first term, but this function is equal to 0 on the interval $[u_{i-m}, u_{i-m+1}]$ for all permissible values of m (see Property 1).
2. $i + z + n - r - 1 = i$, that is $z + n = r + 1$, hence the normalized B-spline basis function in the first term is $N_{i-(k-r-1)}^{k-r-1}(\tilde{u}, u_i)$. According to Lemma 1.3, the degree of this function in u_i is $k - m - r - 1$, hence the degree of the first term can at most be $k - m - r \leq k - m$.
3. $i + z + n - r = i$, that is $z + n = r$, which corresponds to case 2.
4. $i - k + z + n + 1 = i$, which corresponds to case 1.

\square

Corollary 1.5. *For $m = k - 1$, the resulted path is of degree 1, that is if u_i runs from u_{i-1} to u_{i+1} , then the points of the span $\mathbf{s}_{i-k+1}(\tilde{u}, u_i)$ move along straight lines parallel to the side $\mathbf{d}_{i-k}, \mathbf{d}_{i-k+1}$ of the control polygon.*

In this case the path has the following simple form:

$$\mathbf{s}_{i-k+1}(\tilde{u}, u_i) = N_{i-k}^k(\tilde{u}, u_i) \mathbf{d}_{i-k} + N_{i-k+1}^k(\tilde{u}, u_i) \mathbf{d}_{i-k+1},$$

while the normalized B-spline basis functions appearing in this form can be written as

$$\begin{aligned} N_{i-k}^k(\tilde{u}, u_i) &= C_1(\tilde{u}) + C_2(\tilde{u}) \frac{u_i - \tilde{u}}{u_i - u_{i-k+1}} \\ N_{i-k+1}^k(\tilde{u}, u_i) &= C_2(\tilde{u}) \frac{\tilde{u} - u_{i-k+1}}{u_i - u_{i-k+1}} \end{aligned}$$

where

$$C_1(\tilde{u}) = \frac{\tilde{u} - u_{i-k}}{u_{i-1} - u_{i-k}} N_{i-k}^{k-1}(\tilde{u}, u_i), C_2(\tilde{u}) = \frac{\tilde{u} - u_{i-k+1}}{u_{i-1} - u_{i-k+1}} N_{i-k+1}^{k-2}(\tilde{u}, u_i)$$

are constants not depending on u_i . This yields

$$\begin{aligned} \mathbf{s}_{i-k+1}(\tilde{u}, u_i) &= \left(C_1(\tilde{u}) + C_2(\tilde{u}) \left(1 - \frac{\tilde{u} - u_{i-k+1}}{u_i - u_{i-k+1}} \right) \right) \mathbf{d}_{i-k} + C_2(\tilde{u}) \frac{\tilde{u} - u_{i-k+1}}{u_i - u_{i-k+1}} \mathbf{d}_{i-k+1} \\ &= (C_1(\tilde{u}) + C_2(\tilde{u})) \mathbf{d}_{i-k} + C_2(\tilde{u}) \frac{\tilde{u} - u_{i-k+1}}{u_i - u_{i-k+1}} (\mathbf{d}_{i-k+1} - \mathbf{d}_{i-k}). \end{aligned}$$

Until now the movement of those points and parts of the B-spline curve was clarified, the parameters of which are smaller than the knot value u_i subject to change. Similar statements hold for those parts of the B-spline curve, which correspond to the parameter values succeeding the knot u_i .

Lemma 1.6. *The function $N_i^k(\tilde{u}, u_i)$, $\tilde{u} \in [u_{i+m}, u_{i+m+1})$, ($m = 0, \dots, k-2$), $u_i \in [u_{i-1}, u_{i+1}]$ is a rational function of degree $k - m - 1$ in u_i .*

Proof. The proof of this lemma is analogous to that of Lemma 1.3. □

Theorem 1.7. *The path $\mathbf{s}_{i+m}(\tilde{u}, u_i) = \sum_{l=i+m-k+1}^{i+m} N_l^k(\tilde{u}, u_i) \mathbf{d}_l$, $u_i \in [u_{i-1}, u_{i+1}]$ is a rational curve of degree $k - m - 1$ with respect to u_i , $\forall \tilde{u} \in [u_{i+m}, u_{i+m+1})$, ($m = 0, \dots, k-2$).*

Proof. The upper limit of the summation can be decreased to i , since u_i has no influence on $N_l^k(\tilde{u}, u_i)$ for $l > i$ (see Property 4). By using Lemma 1.6, the further part of the proof is analogous to that of Theorem 1.4. □

Corollary 1.8. *For $m = k-2$, the path is of degree 1, that is if u_i runs from u_{i-1} to u_{i+1} then the points of the span $\mathbf{s}_{i+k-2}(\tilde{u}, u_i)$ move along straight lines parallel to the side $\mathbf{d}_{i-1}, \mathbf{d}_i$ of the control polygon.*

Now we can summarize our results based on Theorem 1.4 and 1.7 and their corollaries. Modifying the knot value u_i the points of the spans of a k^{th} order B-spline curve move along rational curves, the degree of which decreases symmetrically from $k-1$ to 1 as the indices of the spans getting farther from i . Hence the points of the spans $\mathbf{s}_{i-k+1}(\tilde{u}, u_i)$ and $\mathbf{s}_{i+k-2}(\tilde{u}, u_i)$ move along straight lines parallel to the corresponding sides of the control polygon. Other parts of the curve remain unchanged (see Figure 1.2).

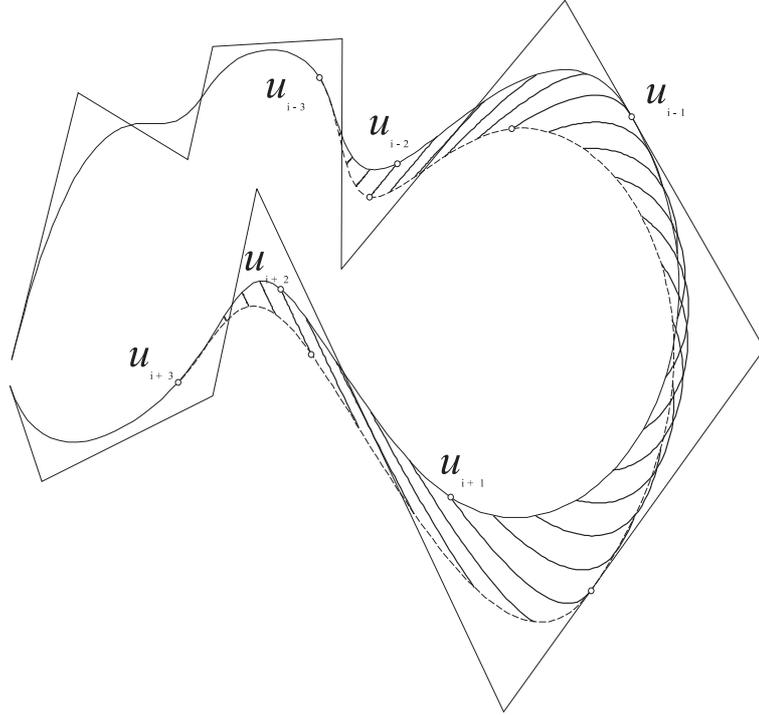


Figure 1.1. Paths of the points of a cubic B-spline curve when u_i runs from u_{i-1} to u_{i+1} and the two extreme positions of the curve: $u_i = u_{i-1}$ (solid line) and $u_i = u_{i+1}$ (dashed line)

1.2.1 The envelope of the family of B-spline curves

The modification of a knot value u_i results a one-parameter family of B-spline curves

$$\mathbf{s}(u, u_i) = \sum_{l=0}^n \mathbf{d}_l N_l^k(u, u_i), \quad u \in [u_{k-1}, u_{n+1}], u_i \in [u_{i-1}, u_{i+1}]. \quad (1.2)$$

In case of $k = 3$ the spans of the curves are parabolic arcs. It is well-known that the tangent lines of these arcs at the knot values coincide with the sides of the control polygon. Modifying a knot value u_i the tangent line remains the same, which can be interpreted as the side of the control polygon is the envelope of the family of these quadratic B-spline curves. In the following theorem the generalization of this property will be proved for arbitrary degree k .

Theorem 1.9. *The family of the k^{th} order B-spline curves $\mathbf{s}(u, u_i) = \sum_{l=0}^n \mathbf{d}_l N_l^k(u, u_i)$, $u \in [u_{k-1}, u_{n+1}]$, $u_i \in [u_{i-1}, u_{i+1}]$, $k > 2$ has an envelope. This envelope is a B-spline curve of order $(k - 1)$ and can be written in the form*

$$\mathbf{b}(v) = \sum_{l=i-k+1}^{i-1} \mathbf{d}_l N_l^{k-1}(v), \quad v \in [v_{i-1}, v_i], \quad (1.3)$$

where $v_j = \begin{cases} u_j & \text{if } j < i \\ u_{j+1} & \text{if } j \geq i \end{cases}$, that is the i^{th} knot value is removed from the knot vector (u_j) of the original curves.

Proof. We prove that the curve $\mathbf{b}(v)$ at its point corresponding to $v = u_i$ touches the curve $\mathbf{s}(u, u_i)$ at the point associated with $u = u_i$, i.e. they have a point and a tangent line on common.

Based on Definition 1.1 the span

$$\mathbf{s}_i(u, u_i) = \sum_{l=i-k+1}^i \mathbf{d}_l N_l^k(u, u_i), \quad u \in [u_i, u_{i+1}] \quad (1.4)$$

the starting point of which is $\mathbf{s}(u_i, u_i)$ can be written in the form

$$\mathbf{s}_i(u, u_i) = \sum_{l=i-k+1}^i \mathbf{d}_l \left(\frac{u - u_l}{u_{l+k-1} - u_l} N_l^{k-1}(u) + \frac{u_{l+k} - u}{u_{l+k} - u_{l+1}} N_{l+1}^{k-1}(u) \right). \quad (1.5)$$

At the specific parameter u_i the value of this function is

$$\mathbf{s}_i(u_i, u_i) = \sum_{l=i-k+1}^{i-1} \mathbf{d}_l \left(\frac{u_i - u_l}{u_{l+k-1} - u_l} N_l^{k-1}(u_i) + \frac{u_{l+k} - u_i}{u_{l+k} - u_{l+1}} N_{l+1}^{k-1}(u_i) \right) \quad (1.6)$$

(the upper limit of the summation can be decreased to $i - 1$, since $N_i^{k-1}(u_i) = N_{i+1}^{k-1}(u_i) = 0$).

Now we insert the knot value u_i between v_{i-1} and v_i ($v_{i-1} = u_{i-1} \leq u_i \leq u_{i+1} = v_i$) by the Böhm's insertion algorithm (cf. [8]). The new knot vector is

$$\hat{v}_j = \begin{cases} v_j = u_j & \text{if } j < i \\ u_i & \text{if } j = i \\ v_{j-1} = u_j & \text{if } j > i \end{cases} \quad (1.7)$$

For the normalized B-spline basis functions $N_l^{k-1}(v)$ and $\hat{N}_l^{k-1}(v)$, defined over the knot vectors (v_j) and (\hat{v}_j) respectively, the following relation holds:

$$N_l^{k-1}(v) = \begin{cases} \hat{N}_l^{k-1}(v) & \text{if } l < i - k + 1; \\ \frac{u_i - \hat{v}_l}{\hat{v}_{l+k-1} - \hat{v}_l} \hat{N}_l^{k-1}(v) + \frac{\hat{v}_{l+k} - u_i}{\hat{v}_{l+k} - \hat{v}_{l+1}} \hat{N}_{l+1}^{k-1}(v) & \text{if } l = i - k + 1, \dots, i - 1; \\ \hat{N}_{l+1}^{k-1}(v) & \text{if } l > i - 1. \end{cases}$$

Based on this fact the following form can be obtained

$$\mathbf{b}(v) = \sum_{l=i-k+1}^{i-1} \mathbf{d}_l \left(\frac{u_i - \hat{v}_l}{\hat{v}_{l+k-1} - \hat{v}_l} \hat{N}_l^{k-1}(v) + \frac{\hat{v}_{l+k} - u_i}{\hat{v}_{l+k} - \hat{v}_{l+1}} \hat{N}_{l+1}^{k-1}(v) \right) \\ v \in [\hat{v}_{i-1}, \hat{v}_{i+1}]$$

and using (1.7) this can be written in the form (since $\hat{v}_j = u_j, \forall j$)

$$\mathbf{b}(u) = \sum_{l=i-k+1}^{i-1} \mathbf{d}_l \left(\frac{u_i - u_l}{u_{l+k-1} - u_l} \dot{N}_l^{k-1}(u) + \frac{u_{l+k} - u_i}{u_{l+k} - u_{l+1}} \dot{N}_{l+1}^{k-1}(u) \right) \quad (1.8)$$

$$u \in [u_{i-1}, u_{i+1}).$$

Comparing (1.6) and (1.8) one can see that $\mathbf{b}(u_i) = \mathbf{s}_i(u_i, u_i)$ holds.

The derivative of the curve (1.8) with respect to u is

$$\dot{\mathbf{b}}(u) = \sum_{l=i-k+1}^{i-1} \mathbf{d}_l \left(\frac{u_i - u_l}{u_{l+k-1} - u_l} \dot{N}_l^{k-1}(u) + \frac{u_{l+k} - u_i}{u_{l+k} - u_{l+1}} \dot{N}_{l+1}^{k-1}(u) \right). \quad (1.9)$$

Based on (1.4), the derivative of the curve $\mathbf{s}_i(u, u_i)$ with respect to u is

$$\dot{\mathbf{s}}_i(u, u_i) = \sum_{l=i-k+1}^i \mathbf{d}_l \dot{N}_l^k(u, u_i), \quad (1.10)$$

while on the other hand, using (1.5)

$$\dot{\mathbf{s}}_i(u, u_i) = \sum_{l=i-k+1}^i \mathbf{d}_l \left(\frac{1}{u_{l+k-1} - u_l} \dot{N}_l^{k-1}(u) - \frac{1}{u_{l+k} - u_{l+1}} \dot{N}_{l+1}^{k-1}(u) + \frac{u - u_l}{u_{l+k-1} - u_l} \dot{N}_l^{k-1}(u) + \frac{u_{l+k} - u}{u_{l+k} - u_{l+1}} \dot{N}_{l+1}^{k-1}(u) \right)$$

holds. Applying Property 3, this can be written as

$$\dot{\mathbf{s}}_i(u, u_i) = \sum_{l=i-k+1}^i \mathbf{d}_l \left(\frac{u - u_l}{u_{l+k-1} - u_l} \dot{N}_l^{k-1}(u) + \frac{u_{l+k} - u}{u_{l+k} - u_{l+1}} \dot{N}_{l+1}^{k-1}(u) + \frac{1}{k-1} \dot{N}_l^k(u) \right). \quad (1.11)$$

Based on (1.10) and (1.11) one can write

$$\frac{u - u_l}{u_{l+k-1} - u_l} \dot{N}_l^{k-1}(u) + \frac{u_{l+k} - u}{u_{l+k} - u_{l+1}} \dot{N}_{l+1}^{k-1}(u) = \frac{k-2}{k-1} \dot{N}_l^k(u).$$

Hence from (1.10) and (1.9) we obtain

$$\dot{\mathbf{b}}(u_i) = \frac{k-2}{k-1} \dot{\mathbf{s}}_i(u_i, u_i).$$

□

The envelope is illustrated by Figure 2.

1.2.2 The envelope of the paths of curve points

Two families of curves have been considered so far, the paths of the points and the family of B-spline curves themselves. These two families of curves can be considered as parameter lines of

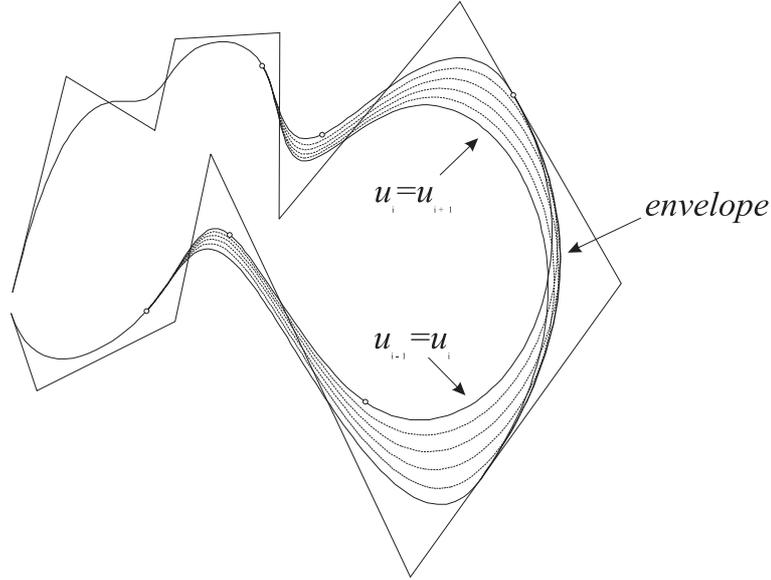


Figure 1.2. Envelope of the family of cubic B-spline curves when u_i runs from u_{i-1} to u_{i+1}

the surface patch

$$\mathbf{s}(u, u_i) = \sum_{l=0}^n \mathbf{d}_l N_l^k(u, u_i), \quad u \in [u_{k-1}, u_{n+1}], \quad u_i \in [u_{i-1}, u_{i+1}].$$

The envelope mentioned above in Theorem 1.9 is a curve on this surface, but the parameter lines behave in a singular way at the points of that curve. We have seen that it is an envelope of the family of B-spline curves. In the next subsection, where we will restrict our consideration to the cubic case ($k = 4$) we will prove, that this curve is also the envelope of the paths and both families have the same osculating plane at every point of this envelope, which plane is also the plane of the envelope itself (c.f. [34]).

Theorem 1.10. *If we consider the surface $\mathbf{s}_i(u, u_i)$, $u \in [u_{k-1}, u_{n+1}]$, $u_i \in [u_{i-1}, u_{i+1}]$ then the envelope of the family of B-spline curves $\mathbf{s}_i(u, \tilde{u}_i)$ is also the envelope of the family of paths $\mathbf{s}_i(\tilde{u}, u_i)$ at the points corresponding to $u = u_i$.*

Proof. It is sufficient to prove, that the two families of curves have points and tangent lines on common at the points corresponding to the parameter value $u = u_i$. If we fix the parameters $u = \tilde{u}$ and $u_i = \tilde{u}_i$ then a member of both families of curves has been selected. Substituting these parameters to both of the curves the existence of the common point $\mathbf{s}_i(\tilde{u}, \tilde{u}_i) = \mathbf{s}_i(\tilde{u}, \tilde{u}_i)$ immediately follows. For the proof of the common tangent lines the first derivatives of these curves will be used. Substituting the parameter $u = u_i$ to the coefficients after some calculations

one can receive, that

$$\begin{aligned} \left. \frac{\partial N_{i-3}^4}{\partial u_i} \right|_{u=u_i} &= -\frac{1}{3} \left. \frac{\partial N_{i-3}^4}{\partial u} \right|_{u=u_i} = \frac{1}{u_{i+1}-u_{i-2}} \frac{u_{i+1}-u_i}{u_{i+1}-u_{i-1}} \\ \left. \frac{\partial N_{i-2}^4}{\partial u_i} \right|_{u=u_i} &= -\frac{1}{3} \left. \frac{\partial N_{i-2}^4}{\partial u} \right|_{u=u_i} = \frac{1}{u_{i+1}-u_{i-1}} \left(\frac{u_i-u_{i-2}}{u_{i+1}-u_{i-2}} - \frac{u_{i+2}-u_i}{u_{i+2}-u_{i-1}} \right) \\ \left. \frac{\partial N_{i-1}^4}{\partial u_i} \right|_{u=u_i} &= -\frac{1}{3} \left. \frac{\partial N_{i-1}^4}{\partial u} \right|_{u=u_i} = -\frac{1}{u_{i+1}-u_{i-1}} \frac{u_i-u_{i-1}}{u_{i+2}-u_{i-1}} \\ \left. \frac{\partial N_i^4}{\partial u_i} \right|_{u=u_i} &= \left. \frac{\partial N_i^4}{\partial u} \right|_{u=u_i} = 0 \end{aligned}$$

which yields, that

$$\left. \frac{\partial \mathbf{s}_i(u, u_i)}{\partial u} \right|_{u=u_i} = -\frac{1}{3} \left. \frac{\partial \mathbf{s}_i(u, u_i)}{\partial u_i} \right|_{u=u_i}$$

i.e. the curves have also tangent lines in common at the points of the envelope. \square

With the help of the second derivatives of the coefficient functions the osculating plane of these curves can also be examined.

Theorem 1.11. *The osculating planes of the two families of curves $\mathbf{s}_i(u, \tilde{u}_i)$ and $\mathbf{s}_i(\tilde{u}, u_i)$ coincide at every point of the envelope and this plane is that of the three control points $\mathbf{d}_{i-3}, \mathbf{d}_{i-2}, \mathbf{d}_{i-1}$ for every u_i .*

Proof. The osculating plane is uniquely defined by the first and second derivatives of the curve. Since Theorem 3. holds for the first derivatives it is sufficient to prove that the second derivatives of these curves are also parallel to each other. Using the second derivatives of the coefficient functions and substituting the parameter value $u = u_i$ the following result can be obtained:

$$\begin{aligned} \left. \frac{\partial^2 N_{i-3}^4}{\partial u_i^2} \right|_{u=u_i} &= \frac{1}{3} \left. \frac{\partial^2 N_{i-3}^4}{\partial u^2} \right|_{u=u_i} = 2 \frac{1}{u_{i+1}-u_{i-2}} \frac{1}{u_{i+1}-u_{i-1}} \\ \left. \frac{\partial^2 N_{i-2}^4}{\partial u_i^2} \right|_{u=u_i} &= \frac{1}{3} \left. \frac{\partial^2 N_{i-2}^4}{\partial u^2} \right|_{u=u_i} = 2 \frac{-u_{i+1}+u_{i-2}-u_{i+2}+u_{i-1}}{(-u_{i+2}+u_{i-1})(u_{i+1}-u_{i-1})(-u_{i+1}+u_{i-2})} \\ \left. \frac{\partial^2 N_{i-1}^4}{\partial u_i^2} \right|_{u=u_i} &= \frac{1}{3} \left. \frac{\partial^2 N_{i-1}^4}{\partial u^2} \right|_{u=u_i} = 2 \frac{1}{(u_{i+1}-u_{i-1})(u_{i+2}-u_{i-1})} \\ \left. \frac{\partial^2 N_i^4}{\partial u_i^2} \right|_{u=u_i} &= \left. \frac{\partial^2 N_i^4}{\partial u^2} \right|_{u=u_i} = 0 \end{aligned}$$

which immediately yields, that

$$\left. \frac{\partial^2 \mathbf{s}_i(u, u_i)}{\partial u^2} \right|_{u=u_i} = \frac{1}{3} \left. \frac{\partial^2 \mathbf{s}_i(u, u_i)}{\partial u_i^2} \right|_{u=u_i}.$$

Hence the osculating planes of the two families of curves coincide at the parameter values $u = u_i$. Moreover, the second derivatives do not depend on u_i , and using the notations

$$A := \frac{\partial^2 N_{i-3}^4}{\partial u_i^2}, \quad B := \frac{\partial^2 N_{i-1}^4}{\partial u_i^2}$$

they can be written in the form

$$\begin{aligned} \left. \frac{\partial^2 \mathbf{s}_i(u, u_i)}{\partial u_i^2} \right|_{u=u_i} &= A(\mathbf{d}_{i-3} - \mathbf{d}_{i-2}) + B(\mathbf{d}_{i-2} - \mathbf{d}_{i-1}) \\ \left. \frac{\partial^2 \mathbf{s}_i(u, u_i)}{\partial u^2} \right|_{u=u_i} &= \frac{1}{3} A(\mathbf{d}_{i-3} - \mathbf{d}_{i-2}) + \frac{1}{3} B(\mathbf{d}_{i-2} - \mathbf{d}_{i-1}). \end{aligned}$$

This means that these derivative vectors are in the plane of the control points $\mathbf{d}_{i-3}, \mathbf{d}_{i-2}, \mathbf{d}_{i-1}$ for every u_i . The same holds for the first derivative vectors since the envelope is a quadratic B-spline curve (a parabola) defined by these control points and it has common tangent lines with both of the families of the curves at $u = u_i$. This yields, that the osculating planes of the curves coincide with the plane of the three control points mentioned above for every u_i . \square

1.2.3 Extension of paths

Paths obtained by the modification of the knot u_i are relatively short arcs. In order to get more information about their characteristics we extend their domain, i.e., we let u_i be smaller than u_{i-1} and larger than u_{i+1} . For these extended paths the following holds.

Theorem 1.12. *Modifying the single multiplicity knot u_i of the B-spline curve $\mathbf{s}(u)$, points of the extended paths of the arcs $\mathbf{s}_{i-1}(u)$ and $\mathbf{s}_i(u)$ tend to the control points \mathbf{d}_i and \mathbf{d}_{i-k} as u_i tends to $-\infty$ and ∞ , respectively, i.e.,*

$$\lim_{u_i \rightarrow -\infty} \mathbf{s}(u, u_i) = \mathbf{d}_i, \lim_{u_i \rightarrow \infty} \mathbf{s}(u, u_i) = \mathbf{d}_{i-k}, \forall u \in [u_{i-1}, u_{i+1}].$$

Proof. We prove the statement for the arc $\mathbf{s}_i(u)$, for $\mathbf{s}_{i-1}(u)$ it can be proved analogously. Denote the original knot values by \bar{u}_j , ($j = 0, 1, \dots, n+k$). For the description of extended paths we will use the knot values $u_j = \bar{u}_j$, ($j = 0, 1, \dots, n+k$) which will differ from the original values only in u_i along the proof. Paths of the points $\mathbf{s}_i(u)$ can be written as

$$\mathbf{s}_i(u, u_i) = \sum_{l=i-k+1}^i \mathbf{d}_l N_l^k(u, u_i), u_i \in [\bar{u}_{i-1}, \bar{u}_{i+1}], u \in [\bar{u}_i, \bar{u}_{i+1}]. \quad (1.12)$$

Limits of this summation are modified when we extend these paths, since if $u_i > \bar{u}_{i+1}$ (i.e., $u_i \rightarrow \infty$) then $N_i^k(u) \equiv 0, u \in [\bar{u}_i, \bar{u}_{i+1}]$ and $N_{i-k}^k(u) \neq 0, u \in [\bar{u}_i, \bar{u}_{i+1}]$, thus these arcs of the extended paths become

$$\mathbf{s}_i(u, u_i) = \sum_{l=i-k}^{i-1} \mathbf{d}_l N_l^k(u, u_i), u_i > \bar{u}_{i+1}, u \in [\bar{u}_i, \bar{u}_{i+1}].$$

It can easily be seen that

$$N_{i-k}^k(u, u_i) = \frac{(u_i - u)^{k-1}}{\prod_{j=1}^{k-1} (u_i - u_{i-j})}$$

where both the numerator and the denominator are polynomials of degree $k-1$ in u_i , and the main coefficient in both polynomials is 1. This yields $\lim_{u_i \rightarrow \infty} N_{i-k}^k(u, u_i) = 1$. Now we prove by induction on k that $\lim_{u_i \rightarrow \infty} N_j^k(u, u_i) = 0, (j = i-k+1, \dots, i-1)$.

i) for $k = 3$

$$N_{i-1}^3(u, u_i) = \frac{(u - u_{i-1})^2}{(u_{i+1} - u_{i-1})(u_i - u_{i-1})},$$

$$N_{i-2}^3(u, u_i) = \frac{(u - u_{i-2})(u_i - u)}{(u_i - u_{i-2})(u_i - u_{i-1})} + \frac{(u_{i+1} - u)(u - u_{i-1})}{(u_{i+1} - u_{i-1})(u_i - u_{i-1})}$$

for which the statement holds.

ii) $k - 1 \rightarrow k$

By Definition 1.1

$$\begin{aligned} N_{i-1}^k(u, u_i) &= \frac{u - u_{i-1}}{u_{i+k-2} - u_{i-1}} N_{i-1}^{k-1}(u, u_i) + \frac{u_{i+k-1} - u}{u_{i+k-1} - u_i} N_i^{k-1}(u, u_i) \\ N_{i-2}^k(u, u_i) &= \frac{u - u_{i-2}}{u_{i+k-3} - u_{i-2}} N_{i-2}^{k-1}(u, u_i) + \frac{u_{i+k-2} - u}{u_{i+k-2} - u_{i-1}} N_{i-1}^{k-1}(u, u_i) \\ &\vdots \\ N_{i-k+1}^k(u, u_i) &= \frac{u - u_{i-k+1}}{u_i - u_{i-k+1}} N_{i-k+1}^{k-1}(u, u_i) + \frac{u_{i+1} - u}{u_{i+1} - u_{i-k+2}} N_{i-k+2}^{k-1}(u, u_i) \end{aligned}$$

Therefore, the k^{th} order functions are linear combinations of functions of order $k - 1$ where the numerator is independent of u_i and the denominator is linear at most in u_i , i.e., the order of the numerator can not be greater than that of the denominator. Thus, from the assumption for $k - 1$, the case of k results too.

If $u_i < \bar{u}_{i-1}$ ($u_i \rightarrow -\infty$) then the limits of the summation (1.12) are not modified. It is easy to show that in this case

$$N_i^k(u, u_i) = \frac{(u - u_i)^{k-1}}{\prod_{j=1}^{k-1} (u_{i+j} - u_i)}$$

which immediately yields $\lim_{u_i \rightarrow -\infty} N_i^k(u, u_i) = 1$.

Equalities $\lim_{u_i \rightarrow -\infty} N_{i-j}^k(u, u_i) = 0$, ($j = 1, \dots, k - 1$) can be proved by induction on k . \square

This property of the extended paths is illustrated in Fig. 1.3.

It is easy to show that, by altering a knot of higher multiplicity, Theorem 1.12 will be of the following form.

Theorem 1.13. *Altering the knot u_i of multiplicity m , points of the extended paths of the arcs $\mathbf{s}_{i-1}(u), \dots, \mathbf{s}_{i+m-1}(u)$ satisfy the equalities*

$$\begin{aligned} \lim_{u_i \rightarrow -\infty} \mathbf{s}_{i+j}(u, u_i) &= \mathbf{d}_{i+m-1}, \quad \lim_{u_i \rightarrow \infty} \mathbf{s}_{i+j}(u, u_i) = \mathbf{d}_{i-k}, \\ (j &= -1, 0, \dots, m - 1), \forall u \in [u_{i+j}, u_{i+j+1}). \end{aligned}$$

1.2.4 The case of higher order contact and higher multiplicity of knots

Theorem 1.9 can be generalized in two ways. At first we prove that the family of B-spline curves (1.2) and the envelope curve (1.3) have higher derivatives in common. Then we will consider the case when the modified knot is of multiplicity $m > 1$.

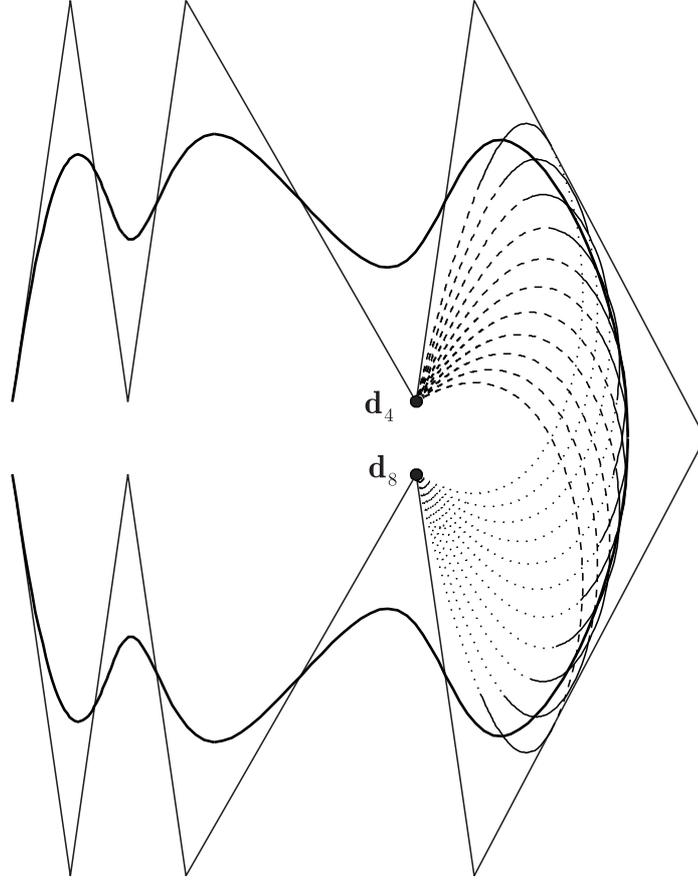


Figure 1.3. A cubic B-spline curve and its extended paths ($n = 12, k = 4, i = 8$).

Theorem 1.14. *Let us consider the one-parameter family of B-spline curves of order k*

$$\mathbf{s}(u, u_i) = \sum_{l=0}^n \mathbf{d}_l N_l^k(u, u_i), \quad u \in [u_{k-1}, u_{n+1}], u_i \in [u_{i-1}, u_{i+1}], k > 2$$

obtained by the modification of the knot u_i of single multiplicity between its neighboring knots. Also consider the B-spline curve

$$\mathbf{b}(v) = \sum_{l=i-k+1}^{i-1} \mathbf{d}_l N_l^{k-1}(v), \quad v \in [v_{i-1}, v_i]$$

of order $k-1$ defined by the same control points \mathbf{d}_l , and the knots $v_j = u_j$ if $j < i$ and $v_j = u_{j+1}$ otherwise, i.e., we leave out the knot u_i from the knot vector $\{u_j\}$. Then the relation between the derivatives of these two curves at $u = v = u_i$ is

$$\left. \frac{d^r}{dv^r} \mathbf{b}(v) \right|_{v=u_i} = \frac{k-1-r}{k-1} \left. \frac{d^r}{du^r} \mathbf{s}(u, u_i) \right|_{u=u_i}, \quad r \geq 0.$$

Proof. The proof follows the basic idea of the proof of Theorem 1.9 where we proved the statement for the case $r = 1$. To make the two curves compatible, we insert the knot u_i into the knot vector

$\{v_j\}$ with Boehm's insertion algorithm [8]. After the conversion of knots from $\{v_j\}$ to $\{u_j\}$, this yields the new representation

$$\mathbf{b}(u) = \sum_{l=i-k+1}^{i-1} \mathbf{d}_l \left(\frac{u_i - u_l}{u_{l+k-1} - u_l} N_l^{k-1}(u) + \frac{u_{l+k} - u_i}{u_{l+k} - u_{l+1}} N_{l+1}^{k-1}(u) \right), u \in [u_{i-1}, u_{i+1}] \quad (1.13)$$

of curve (2.6). It is easy to show that $\mathbf{b}(u_i) = \mathbf{s}(u_i, u_i)$, $\forall u_i \in [u_{i-1}, u_{i+1}]$.

For the $r > 0$ case we consider the r^{th} derivative of the curve (1.13)

$$\frac{d^r}{du^r} \mathbf{b}(u) = \sum_{l=i-k+1}^{i-1} \mathbf{d}_l \left(\frac{u_i - u_l}{u_{l+k-1} - u_l} \frac{d^r}{du^r} N_l^{k-1}(u) + \frac{u_{l+k} - u_i}{u_{l+k} - u_{l+1}} \frac{d^r}{du^r} N_{l+1}^{k-1}(u) \right). \quad (1.14)$$

The r^{th} derivative of a normalized B-spline basis functions of order k is

$$\frac{k-1-r}{k-1} \frac{d^r}{du^r} N_l^k(u) = \frac{u - u_l}{u_{l+k-1} - u_l} \frac{d^r}{du^r} N_l^{k-1}(u) + \frac{u_{l+k} - u}{u_{l+k} - u_{l+1}} \frac{d^r}{du^r} N_{l+1}^{k-1}(u)$$

$$k > 1, r \geq 0,$$

cf. [13]. Thus the r^{th} derivative of the arc $\mathbf{s}_i(u, u_i)$, ($u \in [u_{i-1}, u_{i+1}]$) with respect to u is

$$\frac{k-1-r}{k-1} \frac{d^r}{du^r} \mathbf{s}_i(u, u_i) =$$

$$\sum_{l=i-k+1}^i \mathbf{d}_l \left(\frac{u - u_l}{u_{l+k-1} - u_l} \frac{d^r}{du^r} N_l^{k-1}(u) + \frac{u_{l+k} - u}{u_{l+k} - u_{l+1}} \frac{d^r}{du^r} N_{l+1}^{k-1}(u) \right).$$

The evaluation of this and of equation (1.14) at $u = u_i$ completes the proof. \square

Theorem 1.9 can also be generalized to the case when the multiplicity of the modified knot is higher than 1.

Theorem 1.15. *In the case when the modified knot is of multiplicity $m > 1$, the curve (1.3) becomes*

$$\mathbf{b}(v) = \sum_{l=i-k+m}^{i-1} \mathbf{d}_l N_l^{k-m}(v), v \in [v_{i-1}, v_i]$$

on the knots $v_j = u_j$ if $j < i$ and $v_j = u_{j+m}$ otherwise, and the relation between the derivatives is

$$\frac{d^r}{dv^r} \mathbf{b}(v) \Big|_{v=u_i} = \frac{d^r}{du^r} \mathbf{s}(u, u_i) \Big|_{u=u_i} \prod_{j=1}^m \frac{k-j-r}{k-j}, r \geq 0.$$

Proof. For a proof of this statement we can show at first, by means of the considerations used in the previous proof, that

$$\frac{d^r}{du^r} \widehat{N}_l^{k-j}(u) \Big|_{u=u_i} = \frac{k-j-r}{k-j} \frac{d^r}{du^r} \widetilde{N}_l^{k-j+1}(u) \Big|_{u=u_i}, (j = 1, \dots, m) \quad (1.15)$$

where \widehat{N}_l^{k-j} is defined on the knots $\{\dots, u_{i-1}, u_i = u_{i+1} = \dots = u_{i+m-j-1}, \dots\}$ and \widetilde{N}_l^{k-j+1} is

on $\{\dots, u_{i-1}, u_i = u_{i+1} = \dots = u_{i+m-j}, \dots\}$. The repeated application of (1.15) completes the proof. \square

We mention two corollaries of these properties of B-spline curves and of Theorem 1.10 and Theorem 1.11.

Corollary 1.16. *The curve $\mathbf{b}(v)$ of order $k - m$ is an envelope of the family of curves $\mathbf{s}(u, u_i)$ of order k .*

Corollary 1.17. *For $k > 3$ spatial curves, $\mathbf{b}(v)$ and $\mathbf{s}(u, u_i)$ have also a common osculating plane at the point of contact. However, their curvatures are different, the relation between them is*

$$\kappa_b = \frac{(k-1)(k-m-2)}{(k-2)(k-m-1)} \kappa_s \quad (1.16)$$

thus $\mathbf{b}(v)$ is a singular curve of the surface $\mathbf{s}(u, u_i)$.

Until now only non-rational B-spline curves have been examined, but similar results hold for the rational case. A rational B-spline curve \mathbb{R}^d can always be considered as a central projection of a non-rational B-spline curve in \mathbb{R}^{d+1} . It is clear that G^1 continuity contact and the coincidence of the osculating planes remain valid, since these are preserved during central projection. The degree of a curve cannot increase by a central projection, thus Theorem 1.9 and its corollaries hold for paths of the points of a NURBS curve, except the parallel paths will be concurrent, which will be discussed in the next section. Similarly Theorem 1.10 holds for the rational case, but the envelope will also be a NURBS curve.

1.3 Shape control of B-spline and NURBS curves by knot modification

Constrained based shape control possibilities are discussed in this section, modifying knot values of a non-rational B-spline curve, while the effect of simultaneous modification of knots and weights is presented in the rational case. For the sake of simplicity in some cases we restrict our consideration for the case of cubic curve ($k = 4$). Some of the algorithms discussed below can be generalized for arbitrary k , while others use the specific properties of cubic curves. Results about cubic curve modification are based on [59].

1.3.1 B-spline curve passing through a point

Let a non-rational cubic B-spline curve $\mathbf{s}(u)$ with control points \mathbf{d}_i , ($i = 0, \dots, n$) and knot values u_k , ($k = 0, \dots, n + 4$) be given. Until now the only possibility for the modification of this curve has been the repositioning of its control points. Now we give an algorithm for changing this curve by modifying its knot values in such a way that the curve will pass through a given point \mathbf{p} at the given parameter value \tilde{u} . This point, of course cannot be anywhere: the algorithm works if this point is inside the region defined by the sides of the control polygon and the envelopes mentioned in Theorem 1.7, which are parabolic arcs in the cubic case.

Let point \mathbf{p} be in the region defined by the control points $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}, \mathbf{d}_j$. Let a parameter value $\tilde{u} \in [u_j, u_{j+2})$ be also given. Consider a quadratic B-spline curve $\mathbf{b}(v)$ with the same

dc_933_14

control points and knot values $v_0 = u_0, \dots, v_{j-1} = u_{j-1}, v_j = u_j, v_{j+1} = u_{j+2}, \dots, v_{n+3} = u_{n+4}$. Hence the given value $\tilde{u} \in [v_j, v_{j+1})$. Consider the j^{th} span of the quadratic curve

$$\mathbf{b}_j(v) = \sum_{l=j-2}^j N_l^3(v) \mathbf{d}_l, \quad v \in [v_j, v_{j+1}).$$

Using the monotonicity of the knot values one can write

$$\begin{aligned} v - v_{j-1} &= (v_{j+1} - v_{j-1}) - (v_{j+1} - v) \\ v_{j+2} - v &= (v_{j+2} - v_j) - (v - v_j). \end{aligned}$$

Substituting these formulae to the original equation we obtain the form

$$\begin{aligned} \mathbf{b}_j(v) &= \mathbf{d}_{j-1} + N_{j-2}^3(v) (\mathbf{d}_{j-2} - \mathbf{d}_{j-1}) \\ &\quad + N_j^3(v) (\mathbf{d}_j - \mathbf{d}_{j-1}). \end{aligned}$$

Now consider the affine coordinate system the origin of which is \mathbf{d}_{j-1} and the base vectors are $\mathbf{d}_{j-2} - \mathbf{d}_{j-1}$ and $\mathbf{d}_j - \mathbf{d}_{j-1}$. Let the coordinates of the given point \mathbf{p} in this coordinate system be x and y . This yields the following system of equations:

$$\begin{aligned} \frac{(v_{j+1} - v)(v_{j+1} - v)}{(v_{j+1} - v_{j-1})(v_{j+1} - v_j)} &= x \\ \frac{(v - v_j)(v - v_j)}{(v_{j+2} - v_j)(v_{j+1} - v_j)} &= y \end{aligned}$$

Hence x, y and $v = \tilde{u}$ are given, one can choose two unknowns from the knot values $(v_{j-1}, v_j, v_{j+1}, v_{j+2})$. The system can be solved for any two unknowns, but to avoid the unnecessary changes of farther spans it is better to chose two neighboring values. Solving the system e.g. for v_{j-1}, v_j and considering the quadratic curve $\bar{\mathbf{b}}(v)$ with these knot values $\bar{\mathbf{b}}(\tilde{u}) = \mathbf{p}$ holds. Hence, because of Theorem 1.7, the cubic curve $\bar{\mathbf{s}}(u)$ with the knot values $(\dots u_{j-1} = v_{j-1}, u_j = v_j, u_{j+1} = \tilde{u}, u_{j+2} = v_{j+1} \dots)$ also passes through the point \mathbf{p} at the parameter value \tilde{u} .

Since we have four free parameters v_{j-1}, v_j, v_{j+1} and v_{j+2} , some additional conditions can be assumed in advance. Such a condition can be the given tangent line \mathbf{t} in \mathbf{p} , which yields two additional equations for the derivatives. This system of 4 equations

$$\begin{aligned} \mathbf{b}(\tilde{u}) &= \mathbf{p} \\ \frac{\partial \mathbf{b}(u)}{\partial u}(\tilde{u}) &= \mathbf{t} \end{aligned}$$

can be uniquely solved for $(v_{j-1}, v_j, v_{j+1}, v_{j+2})$. Thus modifying 5 neighboring knot values the cubic curve will pass through a given point and have a given tangent at that point. This tangent line, however cannot be chosen arbitrarily, but between some limits, otherwise the monotonicity of knot values would be defected.

1.3.2 NURBS curve passing through a point

It is a well-known fact, that the modification of the weight w_j of a NURBS curve causes a perspective functional translation of points of the effected arcs, i.e. it pulls/pushes points of the curve toward/away from the control point \mathbf{d}_j . If a given point is on one of the line segments of the paths of this perspective change, one can easily compute the new weight value such a way, that the new curve will pass through the given point. This point can be *almost* anywhere in the convex hull, but for $k > 3$ these concurrent line segments starting from \mathbf{d}_j do not sweep the entire area of the triangle $\mathbf{d}_{j-1}, \mathbf{d}_j, \mathbf{d}_{j+1}$. If the given point is close to the side of the control polygon, the problem can be solved only for changing two neighboring weights. Now we give an algorithm solving this problem with the change of one weight and one knot value [39].

Let a cubic NURBS curve $\mathbf{s}(u)$ and a point \mathbf{p} in the convex hull be given. Let the point \mathbf{p} be in the triangle $\mathbf{d}_{j-1}, \mathbf{d}_j, \mathbf{d}_{j+1}$. Consider the quadratic envelope $\mathbf{b}(v)$ of this NURBS curve changing its knot value u_{j+1} . This parabolic arc intersects all the lines starting from \mathbf{d}_j in this triangle, hence suitably changing the weight w_j there will be a parameter value \tilde{v} , for which $\mathbf{b}(\tilde{v}) = \mathbf{p}$. Now if we modify the knot value u_{j+1} of the cubic curve for $u_{j+1} = \tilde{v}$, the cubic curve will also pass through the point \mathbf{p} . This type of shape modification is illustrated in Fig. 1.3.2.

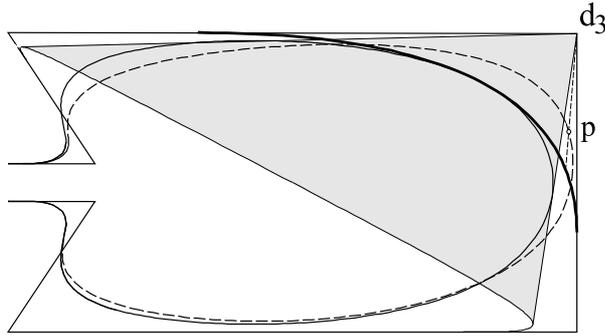


Figure 1.4. Modifying the weight w_3 and the knot u_4 the NURBS curve passes through a given point \mathbf{p} which is outside the area accessible by modifying w_3 only

In this subsection the quadratic envelope has been modified by a weight, where the points of the curve moves along straight lines towards a control point. Similar effect, however, can be achieved in terms of non-rational quadratic B-spline curves by appropriate simultaneous modification of two knot values. More precisely, from the definition of the B-spline functions and the Corollary of Theorem 1.4 one can easily prove the following property:

Theorem 1.18. *The points of the span \mathbf{s}_{i+1} of a non-rational quadratic B-spline curve move along concurrent straight lines with centre \mathbf{d}_i , if the knot values u_i and u_{i+3} are changed simultaneously toward (or away from) each other in such a way, that*

$$u_{i+1} - u_i = u_{i+3} - u_{i+2}$$

holds.

Proof. As we have seen above, the span \mathbf{s}_{i+1} can be written in the form

$$\mathbf{s}_{i+1}(u) = \mathbf{d}_i + N_{i-1}^3(\mathbf{d}_{i-1} - \mathbf{d}_i) + N_{i+1}^3(\mathbf{d}_{i+1} - \mathbf{d}_i).$$

Consider the path of the point $\mathbf{s}_{i+1}(\tilde{u})$. Applying the assumption of the theorem we obtain

$$\begin{aligned} \mathbf{s}_{i+1}(\tilde{u}, u_i, u_{i+3}) &= \mathbf{d}_i + \frac{1}{u_{i+2} - u_i} (C_1(\mathbf{d}_{i-1} - \mathbf{d}_i) + \\ &\quad + C_2(\mathbf{d}_{i+1} - \mathbf{d}_i)) \end{aligned}$$

where C_1 and C_2 are constants. This latter form is an equation of a straight line segment passing through \mathbf{d}_i . \square

The modification of these two knot values, of course, is not so effective, than that of a weight, because the feasible area is greater for the latter case while the number of changing spans is fewer (7 for the two knot values and 3 for the weight), but we have to emphasize, that this theorem allows us to modify non-rational B-spline curves similarly to NURBS curves.

1.3.3 Modification of two weights and a knot value of a NURBS curve

Modifying two neighboring weights w_j, w_{j+1} of a NURBS curve the points of the curve move along straight lines toward or away from the leg $\mathbf{d}_j, \mathbf{d}_{j+1}$ of the control polygon. This change is neither perspective nor parallel. This property can be made more intuitive geometrically by modifying a knot value in addition. Thus the points of a span of the curve will move along concurrent lines passing through any given point of the line $\mathbf{d}_j, \mathbf{d}_{j+1}$ except the inner point of the leg. As we have mentioned in the preceding section, modifying a knot value u_j of a cubic NURBS curve the points of the spans $\mathbf{s}_{j-3}, \mathbf{s}_{j+2}$ will move along two families of concurrent straight lines. Considering the span \mathbf{s}_{j-3} and assuming that $w_{j-4} \neq w_{j-3}$ the following result can be achieved: modifying the knot value u_j the points of this span move along concurrent lines the centre of which is on the line $\mathbf{d}_j, \mathbf{d}_{j+1}$ and its barycentric coordinates are

$$\left(\frac{w_{j-4}}{w_{j-4} - w_{j-3}}, 1 - \frac{w_{j-4}}{w_{j-4} - w_{j-3}} \right).$$

We can easily see, that one of its coordinates must be negative with the usual assumption $w_j \geq 0$ for $\forall j$. Hence this centre cannot be on the leg $\mathbf{d}_j, \mathbf{d}_{j+1}$ but on the rest of the line. Fig.1.3.3 shows a case of this type of modification.

1.3.4 Further constrained modification tools by knots

Extending and improving the basic idea described in [59], in this section we show a method how one can use the results above for constrained shape modification of cubic B-spline curves. These methods are based on the results published in [60].

Let the point \mathbf{p} be in the triangle defined by the control points $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}, \mathbf{d}_j$. Let a parameter value $\tilde{u} \in [u_j, u_{j+2})$ be also given. Consider a quadratic B-spline curve $\mathbf{b}(v)$ with the same control

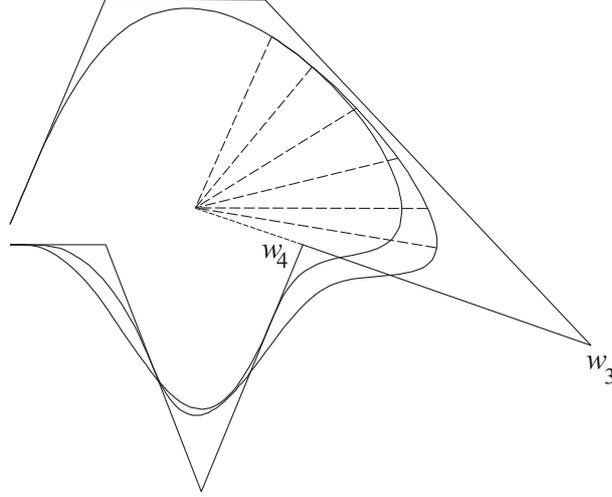


Figure 1.5. Modifying the knot value u_7 the points of the span \mathbf{s}_4 moves along concurrent straight lines the center of which depends on w_3 and w_4 and can be arbitrary chosen on the line of $\mathbf{d}_3\mathbf{d}_4$.

points, and knot values $v_0 = u_0, \dots, v_{j-1} = u_{j-1}, v_j = u_j, v_{j+1} = u_{j+2}, \dots, v_{n+3} = u_{n+4}$. Hence the given value $\tilde{u} \in [v_j, v_{j+1})$. Consider the j^{th} span of the quadratic curve

$$\mathbf{b}_j(v) = \sum_{l=j-2}^j N_l^3(v) \mathbf{d}_l, \quad v \in [v_j, v_{j+1}). \quad (1.17)$$

Utilizing that $N_{j-2}^3(v) + N_{j-1}^3(v) + N_j^3(v) = 1, \forall v \in [v_j, v_{j+1})$, equation (1.17) can be written in the form

$$\begin{aligned} \mathbf{b}_j(v) &= \mathbf{d}_{j-1} + N_{j-2}^3(v) (\mathbf{d}_{j-2} - \mathbf{d}_{j-1}) \\ &\quad + N_j^3(v) (\mathbf{d}_j - \mathbf{d}_{j-1}) \end{aligned}$$

where

$$\begin{aligned} N_{j-2}^3(v) &= \frac{(v_{j+1} - v)^2}{(v_{j+1} - v_{j-1})(v_{j+1} - v_j)} \\ N_j^3(v) &= \frac{(v - v_j)^2}{(v_{j+2} - v_j)(v_{j+1} - v_j)}. \end{aligned}$$

Now, consider the affine coordinate system the origin of which is \mathbf{d}_{j-1} and the base vectors are $\mathbf{e}_1 = \mathbf{d}_{j-2} - \mathbf{d}_{j-1}$ and $\mathbf{e}_2 = \mathbf{d}_j - \mathbf{d}_{j-1}$. Let the coordinates of the given point \mathbf{p} in this coordinate system be x and y . This yields system of equations

$$\begin{aligned} \frac{(v_{j+1} - v)^2}{(v_{j+1} - v_{j-1})(v_{j+1} - v_j)} &= x \\ \frac{(v - v_j)^2}{(v_{j+2} - v_j)(v_{j+1} - v_j)} &= y \end{aligned} \quad (1.18)$$

Hence x, y and $v = \tilde{u}$ are given, one can choose two unknowns from the four knot values $(v_{j-1}, v_j, v_{j+1}, v_{j+2})$. Solving the system, e.g., for v_{j-1}, v_j and considering the quadratic curve $\bar{\mathbf{b}}(v)$ with these knot values $\bar{\mathbf{b}}(\tilde{u}) = \mathbf{p}$ holds. Therefore, because of Theorem 1.7, the cubic curve $\bar{\mathbf{s}}(u)$ with the knot values $(\dots u_{j-1} = v_{j-1}, u_j = v_j, u_{j+1} = \tilde{u}, u_{j+2} = v_{j+1} \dots)$ also passes through the point \mathbf{p} at the parameter value \tilde{u} , and they share a common tangent line here. The point of this solution is that a degree three problem is reduced to a degree two one.

Now, we discuss the relation between the permissible values of the given parameter, the permissible positions of the given point and the choice of the two unknowns in the system of equations (1.18).

Permissible position of the point Since the quadratic B-spline curve has to pass through the point, it can be chosen inside the extreme positions of the curve. The extreme positions of arcs of the quadratic B-spline curve concerning the knot values subject to change are the following:

1. If $v_{j-1} = v_j < v_{j+1} = v_{j+2}$ holds, then $\mathbf{b}_j(v)$ is the quadratic Bézier curve of control points $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}$ and \mathbf{d}_j .
2. If $v_{j-1} = v_j = v_{j+1} < v_{j+2}$ holds, then the parabolic arc degenerates to the side $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}$ of the control polygon.
3. If $v_{j-1} < v_j = v_{j+1} = v_{j+2}$ holds, then the parabolic arc degenerates to the side $\mathbf{d}_{j-1}, \mathbf{d}_j$ of the control polygon.

These extreme positions form the boundary of a plane region (see the shaded area of Fig. 1.6) choosing a point within which one can obtain infinitely many parabolic arcs that pass through the point \mathbf{p} and satisfy the equation (1.17). These quadratic B-spline curves differ from each other only in the knot values v_{j-1}, v_j, v_{j+1} and v_{j+2} and can be chosen between two extreme positions. To obtain these extreme arcs, consider the following two situations: $\mathbf{b}(v_{j-1}) = \mathbf{b}(v_j) = \mathbf{d}_{j-2}$ and $\mathbf{b}(v_{j+1})$ is an inner point of the segment $\mathbf{d}_{j-1}, \mathbf{d}_j$, and the other $\mathbf{b}(v_{j+1}) = \mathbf{b}(v_{j+2}) = \mathbf{d}_j$ and $\mathbf{b}(v_j)$ is an inner point of the segment $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}$. These parabolic arcs can easily be calculated by considering the affine coordinate system $\mathbf{d}_{j-1}, \mathbf{e}_1, \mathbf{e}_2$ described in Section 1.3.4. In this coordinate system let the coordinates of the point \mathbf{p} be (x, y) . The control points of the first extreme arc are: $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}, \mathbf{d}_{j-1} + \mu\mathbf{e}_2, \mu < 1$, and it can be written in the parametric form

$$\mathbf{c}(v) = \mathbf{d}_{j-1} + (1 - v)^2 \mathbf{e}_1 + v^2 \mu \mathbf{e}_2.$$

One of its points will be the point \mathbf{p} at the parameter value $\hat{v} \in (0, 1)$ which is to be determined. For this point

$$\mathbf{p} = \mathbf{d}_{j-1} + x\mathbf{e}_1 + y\mathbf{e}_2$$

also holds. The vectors \mathbf{e}_1 and \mathbf{e}_2 are linearly independent, hence from the equation $\mathbf{p} = \mathbf{c}(\hat{v})$ we obtain the solutions $\hat{v} = 1 - \sqrt{x}, \mu = y / (1 - \sqrt{x})^2$. The other extreme arc can analogously be found. Fig. 1.6 shows the two extreme parabolic arcs passing through \mathbf{p} and their tangent lines.

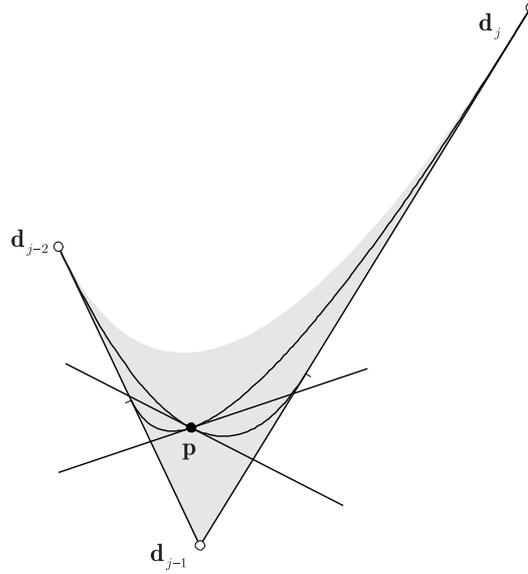


Figure 1.6. The permissible positions of the point \mathbf{p} (the shaded area) and the two extreme positions of the parabolic arcs with their tangent directions

Choices of the parameter and the unknowns In the previous subsection we clarified the permissible positions of the point the modified cubic B-spline curve has to pass through. On the other hand, one can also choose a parameter value \tilde{u} which has to be between the two knots v_j and v_{j+1} .

Once the parameter value \tilde{u} has been chosen and the value $v = \tilde{u}$ has been substituted to the system of equations (1.18), we have four free parameters v_{j-1}, v_j, v_{j+1} and v_{j+2} as possible unknowns. The system can be solved for any two of them, but the knot values has to fulfill the criteria of monotonicity $v_{j-1} \leq v_j \leq \tilde{u} \leq v_{j+1} \leq v_{j+2}$. Having in mind this criteria one can describe an area for every parameter value \tilde{u} and for every pair of knot values chosen to be unknown, inside which the point \mathbf{p} can be specified to find a proper solution. The union of these areas only slightly differs from the general permissible area of the point \mathbf{p} described in subsection 1.3.4 and after fixing the parameter value \tilde{u} the two unknowns are either unique or can be chosen optimally by the system. In most of the cases v_j and v_{j+1} are the best choice but if the point \mathbf{p} is close to the control points \mathbf{d}_{j-2} or \mathbf{d}_j , the pairs v_{j-1}, v_j or v_{j+1}, v_{j+2} can yield smoother curves, respectively.

Now we will describe some possible scenarios for interactive shape control of cubic B-spline curves. A common characteristics of these shape modification methods, that the user do not have to bother with knots, he/she has to specify geometric entities, such as points and lines.

Move a point of the curve to a specified location From the user's point of view, one of the most frequently applied shape control scenario is the following: the designer picks a point $\tilde{\mathbf{p}}$ on the curve and another point \mathbf{p} to which $\tilde{\mathbf{p}}$ is to be moved. This modification can be done by the repositioning of one or more control points, but now we can also apply the knot alteration to achieve the desired modification. Since picking the curve point $\tilde{\mathbf{p}} = \mathbf{s}(\tilde{u})$ yields the choice of the

parameter value \tilde{u} as well, thus one can apply the knot alteration method described in Section 1.3.4.

As we have mentioned above, this modification can also be done by control point repositioning. The advantages of our method are twofold. On the one hand, the modified curve will remain inside the *original* convex hull, while using control point repositioning, the convex hull will also be modified. On the other hand, in spite of the useful local control property of B-spline curves, sometimes the broader interval of change the better to avoid sharp changes in the curvature in comparison with the original one. With knot alteration the number of the modified spans will be larger, but this solution preserves the original shape of the curve much better than the one obtained by control point repositioning.

In Fig. 1.7 one can see a B-spline curve $\mathbf{s}(u)$ a part of which is close to be a straight line. The modification of this part by knot alteration results the curve $\mathbf{s}_k(u)$, and by control point repositioning results the curve $\mathbf{s}_p(u)$. The drawback of the latter method is obvious: the curve $\mathbf{s}_p(u)$ has inflection points, while the curve $\mathbf{s}_k(u)$ obtained by knot alteration is still inside the original convex hull. Fig. 1.8 shows the curvature plots (curvature vs arc length) of these curves.

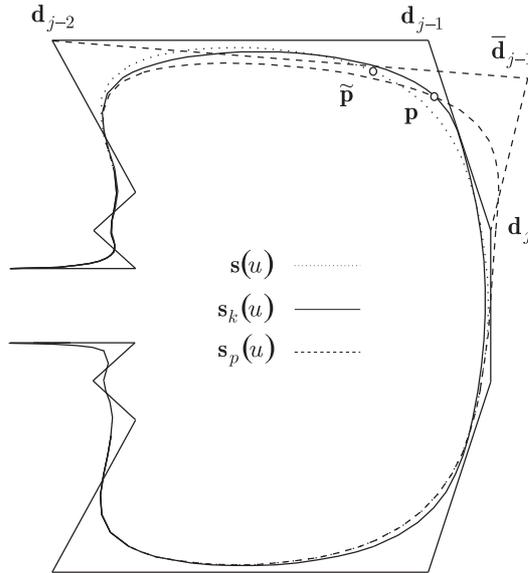


Figure 1.7. A cubic B-spline curve $\mathbf{s}(u)$ with its control polygon and its modifications, that move $\tilde{\mathbf{p}} = \mathbf{s}(0.39)$ to a given location \mathbf{p} . The curve $\mathbf{s}_p(u)$ obtained by repositioning \mathbf{d}_{j-1} has two undesired inflection points, while the curve $\mathbf{s}_k(u)$ obtained by knot alteration remains inside the original convex hull.

One can also be interested in the distance between the original and the modified curves. To measure this distance we introduce a semi-orthogonal distance between two curves measured by the function $d(u) = \text{dist}(\mathbf{s}(u), \mathbf{s}_k(\bar{u}))$. Here dist denotes the Euclidean distance between the points $\mathbf{s}(u)$ and $\mathbf{s}_k(\bar{u})$, where $\mathbf{s}_k(\bar{u})$ is the intersection point of the curve \mathbf{s}_k and the line which is perpendicular to the curve \mathbf{s} at its point $\mathbf{s}(u)$. The graph of this function for both shape control methods is shown in Fig. 1.9. It shows that the greatest distance is two times larger for the

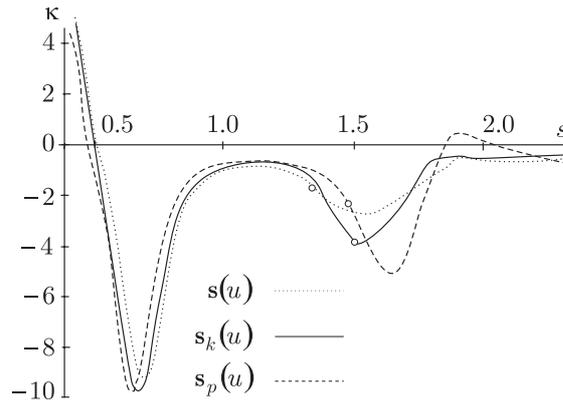


Figure 1.8. A part of the curvature plot of the curves in Fig. 1.7. One can observe the two undesired inflection points of the curve obtained by control point repositioning. Circles indicate the curvature at \mathbf{p} and $\tilde{\mathbf{p}}$.

curve $\mathbf{s}_p(u)$ obtained by control point repositioning, than for $\mathbf{s}_k(u)$ obtained by knot alteration. Further on, the change of this latter curve is more global but smaller in comparison with the rather sudden change of $\mathbf{s}_p(u)$. One can also observe that the greatest difference in the semi-orthogonal distance for $\mathbf{s}_k(u)$ is at \mathbf{p} which is not the case for $\mathbf{s}_p(u)$. This latter property is also typical for knot alteration, however does not always hold. A counter-example can be constructed, e.g., by picking a point of the given curve $\tilde{\mathbf{p}} = \mathbf{s}(\tilde{u})$ and another point of the curve $\mathbf{p} = \mathbf{s}(\hat{u})$. Moving the point $\tilde{\mathbf{p}}$ to \mathbf{p} , the distance function will be 0 at \mathbf{p} , however the two curves are not identical. Apart from these special cases, that can not be considered common shape modification objectives, the maximum of the distance function is typically close to the point picked by the user.

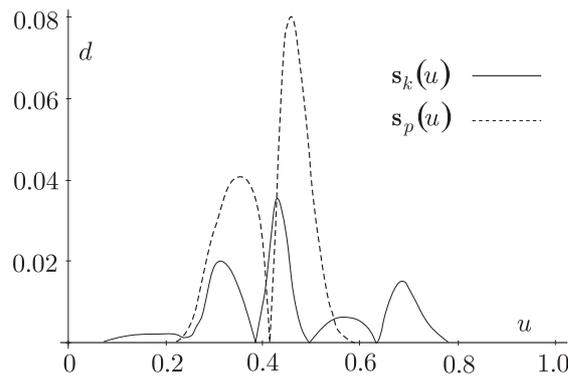


Figure 1.9. The semi-orthogonal distance functions of the curves in Fig. 1.7 obtained by knot alteration and by control point repositioning.

Pass through a point Back to the viewpoint of the user, further scenarios of interactive design can be described. One can pick a point \mathbf{p} inside the permissible area without choosing a

point on the curve to be modified, i.e., without fixing the parameter value \tilde{u} . In this case there are infinitely many curves passing through the point thus the user can find a solution satisfying some further constraint.

An evident option is to prescribe the tangent line at the specified point. After picking the point \mathbf{p} the user can choose a line between the two extreme cases that can be seen in Fig. 1.6. Once the tangent direction $\mathbf{t} = (t_x, t_y)$ in the affine coordinate system specified in Section 1.3.4 has been fixed, the system can compute the quadratic B-spline arc $\mathbf{b}(v)$ that satisfies the constraints, i.e., the system of equations

$$\begin{aligned} \frac{(v_{j+1} - \tilde{u})^2}{(v_{j+1} - v_{j-1})(v_{j+1} - v_j)} &= x \\ \frac{(v_{j+2} - v_j)(v_{j+1} - v_j)}{(\tilde{u} - v_j)^2} &= y \\ \frac{\tilde{u} - v_j}{v_{j+2} - v_j} t_x + \frac{v_{j+1} - \tilde{u}}{v_{j+1} - v_{j-1}} t_y &= 0 \end{aligned} \tag{1.19}$$

has to be solved for \tilde{u}, v_j, v_{j+1} . The existence of a unique solution is guaranteed by the preliminary constraints imposed on the position of \mathbf{p} and \mathbf{t} . The resulted arc $\mathbf{b}(v)$ has the properties $\mathbf{b}(\tilde{u}) = \mathbf{p}$ and $\dot{\mathbf{b}}(\tilde{u}) \parallel \mathbf{t}$, therefore the cubic B-spline curve $\mathbf{s}(u)$ with the knots

$$u_l = \begin{cases} v_l & \text{if } l < j + 1 \\ \tilde{u} & \text{if } l = j + 1 \\ v_{l+1} & \text{if } l > j + 1 \end{cases} \tag{1.20}$$

shares the same properties due to Theorem 1.7.

Another option for the choice from the solutions is to preserve the original parametrization as much as we can. The parametrization is determined by the knot values, three of which are altered in this situation. We consider this change optimal, if the standard deviation, i.e., the square root of the sum of the squared differences, is minimal. In Fig. 1.10 there are three different curves passing through the same point at different parameter values, the one drawn in thick solid line is the optimal. Fig. 1.11 shows the graph of the standard deviation function, the minimum of which has to be determined in order to find the optimal parametrization for the shape modification of Fig. 1.10.

Touch a line The modification of the curve $\mathbf{s}(u)$ can not only be determined by the specification of a location through which the curve has to pass through (as we did in Subsection 1.3.4), but by the specification of a line the curve has to touch as well. These two constraints can be considered dual of each other.

For solvability, the specified tangent line has to intersect the legs $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}$ and $\mathbf{d}_{j-1}, \mathbf{d}_j$ moreover, the straight line segment determined by the intersected points \mathbf{m}_1 and \mathbf{m}_2 has to be within the region bounded by the segments $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}$ and $\mathbf{d}_{j-1}, \mathbf{d}_j$, and the quadratic Bézier curve of control points $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}, \mathbf{d}_j$, see the shaded area of Fig. 1.12. This region coincides with the area of permissible positions of \mathbf{p} , cf. Subsection 1.3.4.

In the affine coordinate system $\mathbf{d}_{j-1}, \mathbf{e}_1, \mathbf{e}_2$ described in Section 1.3.4, the endpoints of the

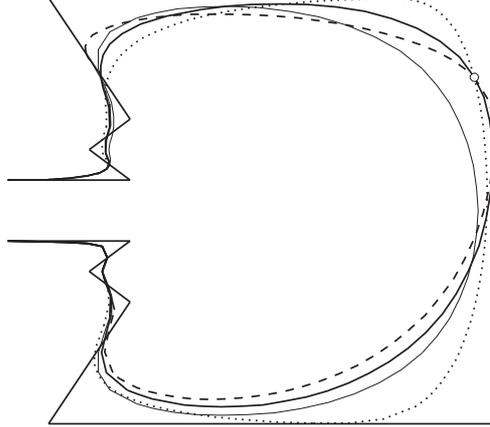


Figure 1.10. Three different curves passing through a specified point, obtained by the modification of three consecutive knots. The original curve (thin solid line) has a uniform knot vector $\{\dots, 0.2, 0.3, 0.4, 0.5, 0.6, \dots\}$, the optimal parametrization for the modified curve is: $\tilde{u} = 0.413, \{\dots, 0.2, 0.329, 0.413, 0.45, 0.6, \dots\}$ (thick solid line), while the other two non-optimal parametrizations are $\tilde{u} = 0.3, \{\dots, 0.2, 0.20007, 0.3, 0.338, 0.6, \dots\}$ (dashed line) and $\tilde{u} = 0.55, \{\dots, 0.2, 0.51, 0.55, 0.58, 0.6, \dots\}$ (dotted line).

tangential segment has the form

$$\mathbf{m}_1 = \lambda_1 \mathbf{e}_1, \quad \mathbf{m}_2 = \lambda_2 \mathbf{e}_2$$

Points of this segment are inside the permissible region if and only if

$$\lambda_1 + \lambda_2 \leq 1.$$

If $\lambda_1 + \lambda_2 = 1$ then there is a unique solution, namely the quadratic Bézier curve of control points $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}, \mathbf{d}_j$, i.e., the bounding parabolic arc of the permissible region. Otherwise, there is an infinite number of such quadratic Bézier curves that touch the segment $\mathbf{m}_1, \mathbf{m}_2$ and for the control points of which

$$\mathbf{b}_0 = \mathbf{d}_{j-1} + (\lambda_1 + \alpha) \mathbf{e}_1, \quad \mathbf{b}_1 = \mathbf{d}_{j-1}, \quad \mathbf{b}_2 = \mathbf{d}_{j-1} + \left(\lambda_2 + \frac{\lambda_1 \lambda_2}{\alpha} \right) \mathbf{e}_2$$

is fulfilled. For \mathbf{b}_0 and \mathbf{b}_2 to be on the segments $\mathbf{d}_{j-2}, \mathbf{d}_{j-1}$ and $\mathbf{d}_{j-1}, \mathbf{d}_j$, respectively, the inequality

$$\frac{\lambda_1 \lambda_2}{1 - \lambda_2} \leq \alpha \leq 1 - \lambda_1$$

has to be satisfied. Points of contact of these parabolas with the line $\mathbf{m}_1, \mathbf{m}_2$ form a segment the endpoints of which can be obtained with the substitutions $\alpha = \lambda_1 \lambda_2 / (1 - \lambda_2)$ and $\alpha = 1 - \lambda_1$. The free parameter α can be fixed by the specification of the point of contact \mathbf{p} . Another option is to specify the curvature of the modified cubic B-spline curve at the point of contact. It is

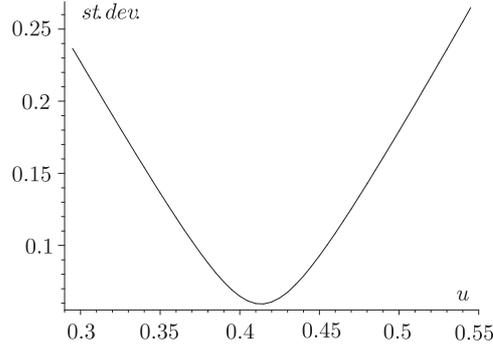


Figure 1.11. The graph of the standard deviation function the minimum of which provides the optimal parametrization for the shape modification in Fig. 1.10.

feasible, since the curvature of the parabola is

$$\kappa_p(\alpha) = \frac{(\alpha + \lambda_1)^2}{2\alpha\lambda_1} \frac{|\mathbf{m}_1 \times \mathbf{m}_2|}{|\mathbf{m}_1 - \mathbf{m}_2|^3}$$

thus with the application of Corollary 1.17, the curvature of the parabola is

$$\kappa_c(\alpha) = \frac{(\alpha + \lambda_1)^2}{3\alpha\lambda_1} \frac{|\mathbf{m}_1 \times \mathbf{m}_2|}{|\mathbf{m}_1 - \mathbf{m}_2|^3}$$

at the point of contact. Thus the shape modification process is as follows:

- the user specifies the points \mathbf{m}_1 and \mathbf{m}_2 ,
- the system responds the segment of possible points of contact,
- the user locates the point of contact \mathbf{p} , or specifies the curvature (or the more intuitive radius of curvature),
- the system solves the system of equations (1.19) for \tilde{u}, v_j, v_{j+1} with the substitution $\mathbf{t} = (\mathbf{m}_2 - \mathbf{m}_1)$.

Thus with the settings of (1.20) the segment $\mathbf{m}_1, \mathbf{m}_2$ will touch the curve $\mathbf{s}(u)$ at the point \mathbf{p} .

Some geometric aspects of knot modification of B-spline curves were presented in this section. We proved the existence of an envelope of the one-parameter family of B-spline curves, obtained by the modification of a knot value of single or higher multiplicity. Constrained local shape control methods for cubic B-spline curves has also been presented, that are based on the alteration of knot values, and utilize this envelope. A definite advantage of knot modification over the repositioning of control points is that the modified curve always remains within the convex hull of the original curve. Several scenarios were described for shape modification of cubic B-spline curves, including:

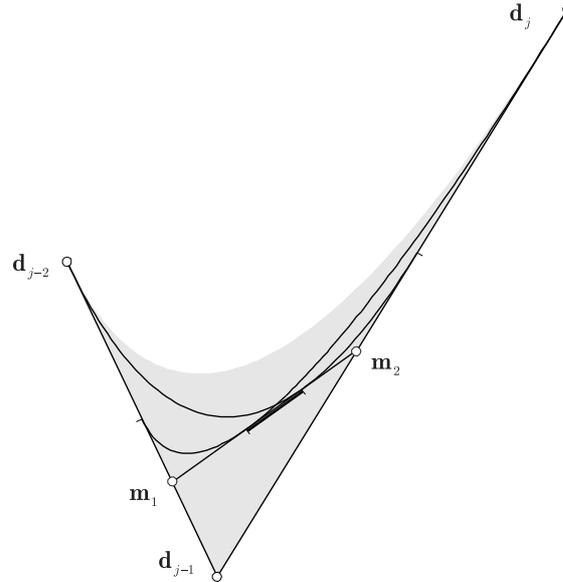


Figure 1.12. Permissible position of tangent segments (shaded region), the range of points of contact and the two extreme positions of the parabolas.

- pass through a point at a prescribed parameter value,
- pass through a point with a prescribed tangent direction,
- pass through a point with the minimal change of parametrization,
- touch a line with a prescribed point of contact.

Using these methods, users do not have to deal with knots, they just have to specify geometric constraints.

Shape modifications methods, we have described for cubic B-spline curves, can be generalized for B-spline curves of arbitrary degree, using Theorem 1.14. Thus, if we have a B-spline curve $\mathbf{s}(u)$ of order $k > 3$, and we want to perform a shape modification of the type described in Section 1.3.4 with the modification of the knot u_{j+1} , we have to insert u_{j+1} repeatedly with Böhm's knot insertion algorithm (cf. [8]) until its multiplicity becomes $k - 3$. Thus the envelope of the family of curves $\mathbf{s}(u, u_{j+1})$ will be a quadratic B-spline for which considerations described in the previous sections are valid. Nevertheless, one has to take into account that knot insertion results a new control polygon the concerned part of which is closer to the curve than the original one, consequently, the region of change (permissible position of \mathbf{p} , cf. Subsection 1.3.4) becomes smaller.

1.4 Extension to surfaces

In the previous section we have described the effect of knot modifications of B-spline and NURBS curves. These results are generalized to surfaces in [35] and [40]. Altering one or two knot values of a B-spline surface, we proved that the family of B-spline surfaces obtained by knot alteration

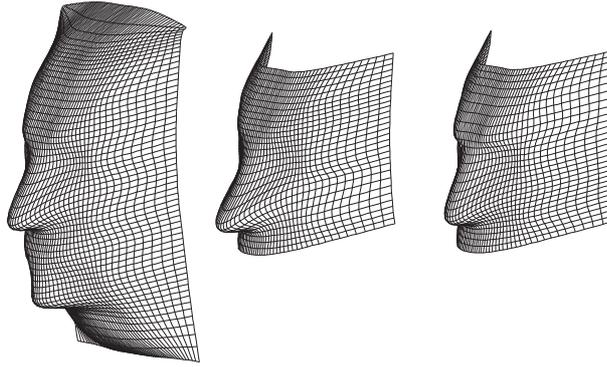


Figure 1.13. A (5,4) order B-spline surface and its modifications by altering a knot in the u (vertical) direction. Left: the original surface; middle: the modified strip of the surface if $u_i = u_{i-1}$ and right: if $u_i = u_{i+1}$.

possesses an envelope which is a lower order B-spline surface. We also proved that simultaneously modifying the knot values u_p and v_q a topologically quadrilateral part of the surface is effected containing at most $4(k-1)(l-1)$ number of patches around the patch $\mathbf{s}_{p,q}(u, v)$. The points of these patches move on rational surfaces the degree of which decrease in a central symmetrical way as we consider farther patches in both parameter directions. Along the sides of this array of patches path-surfaces are ruled surfaces, while at the four corners one can find bilinear path-surfaces. We also proved that the envelope of the family of B-spline surfaces is also an envelope of these paths.

2 New curve types in geometric modeling

In the last decade several new types of spline curves and surfaces have been introduced to CAGD. In one hand, additional parameters are frequently incorporated into the functions in order to control the shape of the curve. The NURBS curve itself associates weights with the control points, which can also be considered as shape parameters. This curve however, suffers from several drawbacks due to the relative complexity of rational basis functions (c.f. [23], [89]). The first attempt to describe B-spline curve with shape parameters, while preserving polynomial basis functions was β -spline curve [5], [6], but several other approaches can be found in recent papers as well [29], [106], [115]. Among them, one of the most promising curve types is the quartic curve of Han [30], which preserves all the nice properties of B-spline curve, having local shape parameters and quartic polynomial basis functions.

On the other hand, instead of polynomial basis, that was a reasonable choice in the epoch of computers with modest computational capabilities, trigonometric functions are incorporated into the base functions. The theoretical fundamentals for this kind of curves have been laid in [92]. C-Bézier and uniform CB-spline curves are defined by means of the basis $\{\sin t, \cos t, t, 1\}$, that was generalized to $\{\sin t, \cos t, t^{k-3}, t^{k-4}, \dots, t, 1\}$ (cf. [15], [117], [118], [119]). Wang et al. introduced NUAT B-spline curves [111] that are the non-uniform generalizations of CB-spline curves. The other basic type is the HB-spline curve, the basis of which is $\{\sinh t, \cosh t, t, 1\}$, and $\{\sinh t, \cosh t, t^{k-3}, t^{k-4}, \dots, t, 1\}$ in higher order [74], [92]. Li and Wang developed its non-uniform generalization [71].

The reason for the introduction of these new spline curves and surfaces is that they are capable of the exact description of such curves and surfaces that are of great importance in applications. Such curves and surfaces are circle and circular cylinder [117], ellipse [119], surfaces of revolution [82], cycloid [78], helix [92], hyperbola and catenary [74]. Although, some of these object can be described by the traditional NURBS technique, the evaluation of CB-spline and HB-spline are more stable [79], [78].

These splines of uniform parametrization have been unified in two similar papers by Zhang and Krause [120], and Zhang et al. [121] with the name FB-spline. FB-spline curves that include uniform HB-spline, CB-spline curves and the uniform B-spline curve itself inherit most advantageous properties of B-spline curves.

Properties of these curves have been clarified mainly from theoretical point of view ([79], [78], [82], [93], [114]). However, in order to be applied, it is essential to develop tools that can be used for constrained shape modification. The objective of this section is to examine the effect of the modification of control points and shape parameters on the shape of these curves, based on the results published in [41, 42, 43, 45, 46, 61].

2.1 Linear blending of curves - the quartic curve of Han

The purpose of this subsection is to provide a general framework of certain types of curves with shape parameters by linear blending, and study geometric properties of these curve, especially the effect of shape parameters on the shape of the curve, based on [61]. We describe the linear

blending method in Section 2.1.1 and 2.1.2. Based on this approach general properties of shape parameters are discussed in Section 2.1.3, while practical computational methods for constrained shape modification are presented in Section 2.1.4.

2.1.1 Definition and linear blending description

In [30] a quartic polynomial curve with shape parameters is defined in the following way:

Definition 2.1. Given a sequence of control points \mathbf{p}_i , ($i = 0, \dots, 3$) the arc is defined by

$$\mathbf{c}(\lambda_1, \lambda_2, t) = \sum_{i=0}^3 B_i(\lambda_1, \lambda_2, t) \mathbf{p}_i, \quad \lambda_1, \lambda_2 \in [-8, 1], \quad t \in [0, 1], \quad (2.1)$$

where the basis functions are

$$\begin{aligned} B_0(\lambda_1, \lambda_2, t) &= \frac{1}{24} \left((4 - \lambda_1) (1 - t)^4 + 4(1 - \lambda_1) (1 - t)^3 t \right), \\ B_1(\lambda_1, \lambda_2, t) &= \frac{1}{24} \left(2(8 + \lambda_1) (1 - t)^4 + 8(8 + \lambda_1) (1 - t)^3 t + 72(1 - t)^2 t^2 + \right. \\ &\quad \left. 4(7 - \lambda_2) (1 - t) t^3 + (4 - \lambda_2) t^4 \right), \\ B_2(\lambda_1, \lambda_2, t) &= \frac{1}{24} \left((4 - \lambda_1) (1 - t)^4 + 4(7 - \lambda_1) (1 - t)^3 t + 72(1 - t)^2 t^2 + \right. \\ &\quad \left. 8(8 + \lambda_2) (1 - t) t^3 + 2(8 + \lambda_2) t^4 \right), \\ B_3(\lambda_1, \lambda_2, t) &= \frac{1}{24} \left(4(1 - \lambda_2) (1 - t) t^3 + (4 - \lambda_2) t^4 \right). \end{aligned}$$

Remark 2.2. For the sake of simplicity, in the definition and throughout the section we deal with a curve arc defined by four control points. A curve with arbitrary number of control points \mathbf{p}_i , ($i = 0, \dots, n$) can naturally be defined by consecutive arcs

$$\mathbf{c}_j(\lambda_j, \lambda_{j+1}, t) = \sum_{i=0}^3 B_i(\lambda_j, \lambda_{j+1}, t) \mathbf{p}_{i+j-1}, \quad j = 1, \dots, n - 2.$$

Remark 2.3. The curve can also be extended to the non-uniform case in the usual way, i.e. by intersecting knots $0 = u_1 < u_2 < \dots < u_{n-1} = 1$ into the domain of definition $[0, 1]$ and substituting the parameter t in the j^{th} arc with

$$t = \frac{u - u_j}{u_{j+1} - u_j}.$$

All results of this section can easily be generalized to non-uniform curves composed of multiple arcs.

After some calculations one can observe that for uniform shape parameter ($\lambda_1 = \lambda_2 = \lambda$) the curve (2.1) can also be described by linearly blending the classical uniform cubic B-spline curve $\mathbf{b}(t)$ and a quartic polynomial curve $\mathbf{l}(t) = \sum_{i=0}^3 Q_i(t) \mathbf{p}_i$ by

$$\mathbf{c}(\lambda, t) = \lambda \mathbf{l}(t) + (1 - \lambda) \mathbf{b}(t), \quad (2.2)$$

where the basis functions of $\mathbf{l}(t)$ are of the form

$$\begin{aligned} Q_0(t) &= \frac{1}{8} - \frac{1}{2}t + \frac{3}{4}t^2 - \frac{1}{2}t^3 + \frac{1}{8}t^4, \\ Q_1(t) &= \frac{3}{4} - \frac{3}{2}t^2 + t^3 - \frac{1}{8}t^4, \\ Q_2(t) &= \frac{1}{8} + \frac{1}{2}t + \frac{3}{4}t^2 - \frac{1}{2}t^3 - \frac{1}{8}t^4, \\ Q_3(t) &= \frac{1}{8}t^4. \end{aligned} \tag{2.3}$$

2.1.2 Linear blending on a common basis

Here we provide a more general framework for the linear blending approach. A common characteristic of shape modification methods is that they modify the shape of the curve

$$\mathbf{b}(t) = \sum_{i=0}^3 B_i(t) \mathbf{p}_i$$

by pulling it towards (or pushing it away from) a target curve

$$\mathbf{l}(t) = \sum_{j=0}^m G_j(t) \mathbf{g}_j$$

by means of a convex combination of the two curves. Thus, the modified curve is of the form

$$\mathbf{c}(\lambda, t) = q(\lambda)\mathbf{l}(t) + (1 - q(\lambda)) \mathbf{b}(t).$$

If, e.g., $\mathbf{g}_i = \mathbf{p}_i$, $B_i(t)$ are the cubic uniform normalized B-spline basis functions, $n = m = 3$, $q(\lambda) = \lambda$ and $G_i(t) = Q_i(t)$ defined by (2.3) we obtain the quartic curve of Han with uniform shape parameters, but other curves with shape parameter, like α B-spline curve [73], [106], GB-spline curve [29] or SPB-spline curve [115] can also be described by this framework.

An obvious reparametrization of the target curve would be the $G_i(t) = B_i(t)$ choice, i.e., when the target curve is described in the basis of the curve to be modified. In our case the original curve is a cubic B-spline curve, while the modified curve is a quartic one, thus our aim is now to describe these curves in a common basis, which will turn to be the quartic Bernstein basis. As we will see, this reparametrization allows us to describe the effects of shape parameter alteration by simple control point repositioning.

Due to [30] the curve (2.1) can be written in the following form:

$$\begin{aligned} \mathbf{c}(\lambda_1, \lambda_2, t) &= (1-t)^4 \mathbf{c}(\lambda_1, \lambda_2, 0) + 2(1-t)^3 t \mathbf{a}_1 + \\ & 2((1-t)^3 t + 3(1-t)^2 t^2 + (1-t)t^3) \mathbf{m} + \\ & 2(1-t)t^3 \mathbf{a}_2 + t^4 \mathbf{c}(\lambda_1, \lambda_2, 1), \end{aligned}$$

where

$$\mathbf{a}_i = \frac{1}{12} ((1 - \lambda_i)\mathbf{p}_{i-1} + 2(5 + \lambda_i)\mathbf{p}_i + (1 - \lambda_i)\mathbf{p}_{i+1}) \quad i = 1, 2$$

$$\mathbf{m} = \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2).$$

Now, we want to rewrite this quartic curve into Bézier form

$$\mathbf{c}((\lambda_1, \lambda_2, t) = \sum_{i=0}^4 N_i^4(t) \mathbf{g}_i(\lambda_1, \lambda_2), \quad (2.4)$$

where $N_i^4(t)$ are the well-known Bernstein basis functions

$$N_i^4(t) = \binom{4}{k} (1-t)^k t^{(4-k)}, \quad k = 0, \dots, 4.$$

After some calculation we obtain the control points \mathbf{g}_i of the Bézier curve

$$\begin{aligned} \mathbf{g}_0(\lambda_1, \lambda_2) &= \mathbf{c}(\lambda_1, \lambda_2, 0) = \frac{1}{24} ((4 - \lambda_1)\mathbf{p}_0 + 2(8 + \lambda_1)\mathbf{p}_1 + (4 - \lambda_1)\mathbf{p}_2), \\ \mathbf{g}_1(\lambda_1, \lambda_2) &= \frac{1}{2}(\mathbf{a}_1 + \mathbf{m}) = \frac{1}{24} ((1 - \lambda_1)\mathbf{p}_0 + 2(8 + \lambda_1)\mathbf{p}_1 + (7 - \lambda_1)\mathbf{p}_2), \\ \mathbf{g}_2(\lambda_1, \lambda_2) &= \mathbf{m} = \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2), \\ \mathbf{g}_3(\lambda_1, \lambda_2) &= \frac{1}{2}(\mathbf{a}_2 + \mathbf{m}) = \frac{1}{24} ((7 - \lambda_2)\mathbf{p}_1 + 2(8 + \lambda_2)\mathbf{p}_2 + (1 - \lambda_2)\mathbf{p}_3), \\ \mathbf{g}_4(\lambda_1, \lambda_2) &= \mathbf{c}(\lambda_1, \lambda_2, 1) = \frac{1}{24} ((4 - \lambda_2)\mathbf{p}_1 + 2(8 + \lambda_2)\mathbf{p}_2 + (4 - \lambda_2)\mathbf{p}_3). \end{aligned}$$

Description (2.1) of the curve is useful from user interface point of view, while description (2.4) is advantageous if we want to integrate the curve into nowadays CAD systems, i.e. when we have to convert the curve into B-spline or NURBS representation.

In accordance with this form, for $\lambda_1 = \lambda_2 = 0$ curve (2.1) is the cubic B-spline curve, which can also be written in quartic Bézier form:

$$\mathbf{c}(0, 0, t) = \sum_{i=0}^3 B_i(t)\mathbf{p}_i = \sum_{j=0}^4 N_j^4(t)\mathbf{g}_j(0, 0),$$

where, using the notations $\mathbf{b}_j = \mathbf{g}_j(0, 0)$,

$$\begin{aligned} \mathbf{b}_0 &= \frac{1}{6}(\mathbf{p}_0 + 4\mathbf{p}_1 + \mathbf{p}_2), \\ \mathbf{b}_1 &= \frac{1}{24}(\mathbf{p}_0 + 16\mathbf{p}_1 + 7\mathbf{p}_2), \\ \mathbf{b}_2 &= \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2), \\ \mathbf{b}_3 &= \frac{1}{24}(7\mathbf{p}_1 + 16\mathbf{p}_2 + \mathbf{p}_3), \\ \mathbf{b}_4 &= \frac{1}{6}(\mathbf{p}_1 + 4\mathbf{p}_2 + \mathbf{p}_3). \end{aligned}$$

Note, that the first two control points \mathbf{g}_0 and \mathbf{g}_1 of the Bézier curve depend only on λ_1 , \mathbf{g}_2 is fixed, while \mathbf{g}_3 and \mathbf{g}_4 depend exclusively on λ_2 .

2.1.3 Shape parameter alteration

Now, we describe some results altering the shape parameter $\lambda_1 \in [-8, 1]$ with some fixed value of λ_2 (analogous results can be achieved for $\lambda_2 \in [-8, 1]$ with fixed λ_1). The extreme positions of the Bézier curve (2.4) are $\mathbf{c}(-8, \lambda_2, t)$ and $\mathbf{c}(1, \lambda_2, t)$. As we have seen, the original definition of Han provides fixed control points and altering shape parameter, but now this shape parameter alteration can also be described by the repositioning of the first two new control points of the Bézier representation, without using the shape parameter (see Fig. 2.1).

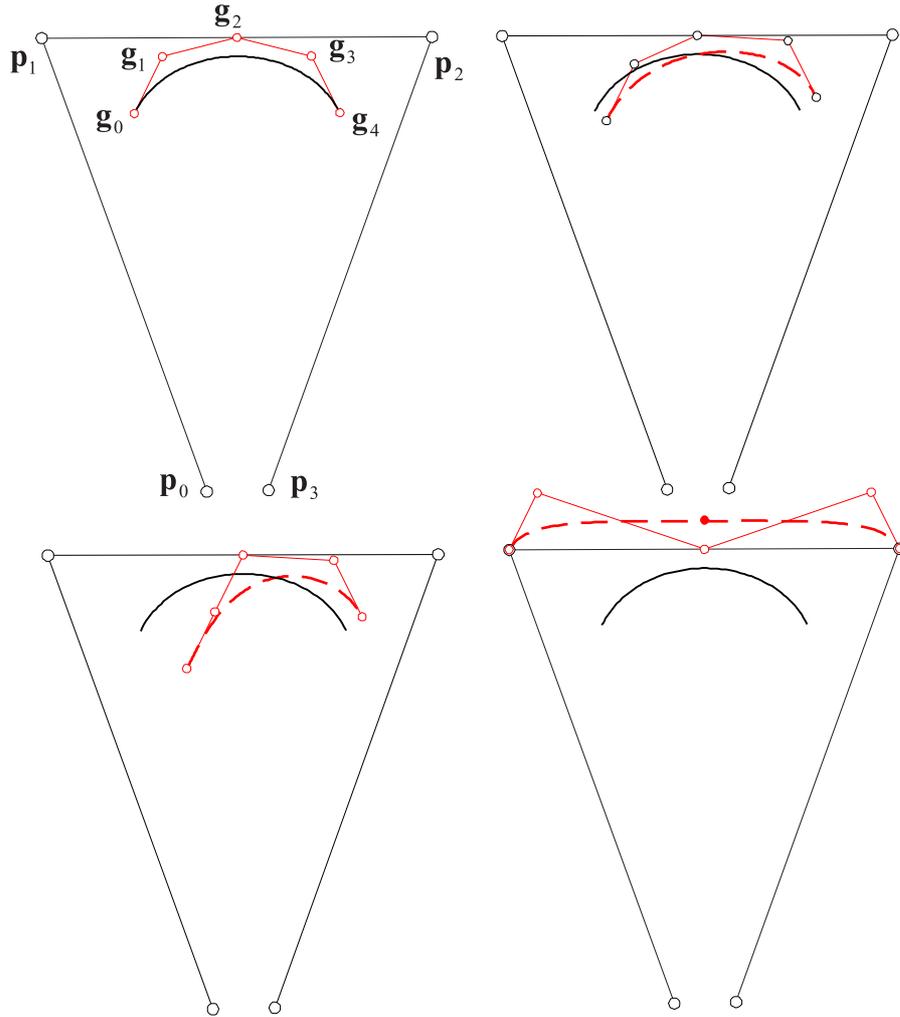


Figure 2.1. The original B-spline curve with its Bézier control polygon (upper left), the curve of Han with its Bézier polygon for $\lambda_1 = -0.5, \lambda_2 = 0.7$ (upper right), for $\lambda_1 = -2, \lambda_2 = 0.7$ (lower left) and for $\lambda_1 = \lambda_2 = 4$ (lower right) along with the original curve

As one can observe in Equation (2.2), the curve $\mathbf{c}(t)$ can be described as the linear blending of two extreme curves, the cubic B-spline curve and a quartic curve. Since these curves have all been described in the same basis by Equation (2.4), and \mathbf{g}_0 and \mathbf{g}_1 are linear functions of λ_1 ,

after some calculation we get the linear blending form

$$\mathbf{c}(\lambda_1, \lambda_2, t) = \sum_{i=0}^4 (q(\lambda_1)\mathbf{g}_i(1, \lambda_2) + (1 - q(\lambda_1))\mathbf{g}_i(-8, \lambda_2)) N_i^4(t).$$

If $\lambda_1 \in [0, 1]$, i.e. we let the curve be modified only between the B-spline curve and the upper limit curve, as is usual in other shape parameter forms, then the linear blending function is $q(\lambda_1) = \lambda_1$ (while $\mathbf{g}_i(-8, \lambda_2)$ has to be substituted by $\mathbf{g}_i(0, \lambda_2)$, of course). In this case and for $\lambda_2 = 0$ this equation is of the special form

$$\mathbf{c}(\lambda_1, 0, t) = \sum_{i=0}^4 (\lambda_1\mathbf{g}_i(1, 0) + (1 - \lambda_1)\mathbf{b}_i) N_i^4(t).$$

If we let λ_1 to be changed in the whole domain, i.e. $\lambda_1 \in [-8, 1]$, then the blending function is

$$q(\lambda_1) = \frac{8}{9} + \frac{1}{9}\lambda_1.$$

More generally, linear blending can handle any range of the shape parameter, that is if we would let $\lambda_1 \in [a, b]$ then the form remains valid (naturally substituting $\mathbf{g}_i(1, \lambda_2)$ by $\mathbf{g}_i(b, \lambda_2)$ and $\mathbf{g}_i(-8, \lambda_2)$ by $\mathbf{g}_i(a, \lambda_2)$) with blending function

$$q(\lambda_1) = \frac{\lambda_1 - a}{b - a}.$$

Here a and b are not necessarily in the range $[-8, 1]$, although exceeding this range the curve will lose important features like the convex hull property. For $\lambda_1 = 4$, however the curve interpolates \mathbf{p}_1 , as one can immediately observe from the equation of the control point \mathbf{g}_0 of the Bézier curve (2.4).

We also have to note, that altering the shape parameters one can naturally expect similar curvature plots and monotonicity properties than that of the original curve. But for $\lambda_1 < -2$ or $\lambda_2 < -2$ the curve can have undesired inflexion points. This is a consequence of the fact, that if $\lambda_1 = -2$ then

$$\mathbf{g}_0(-2, \lambda_2) = \frac{1}{4}(\mathbf{p}_0 + 2\mathbf{p}_1 + \mathbf{p}_2) = \mathbf{a}_1,$$

but, as we have seen,

$$\mathbf{g}_1(\lambda_1, \lambda_2) = \frac{1}{2}(\mathbf{a}_1 + \mathbf{m})$$

$$\mathbf{g}_2(\lambda_1, \lambda_2) = \mathbf{m},$$

which immediately yields, that the control points $\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2$ of the Bézier representation of the curve are collinear for $\lambda_1 = -2$ and \mathbf{g}_2 bisects the segment $\mathbf{g}_0\mathbf{g}_1$. The curvature at \mathbf{g}_0 (at $t = 0$) vanishes, since it is proportional to the area of the triangle with vertices $\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2$. (This can also be seen by Theorem 2 in [30].) In case of plane curves, if $\lambda_1 < -2$ the sign of the curvature of the modified curve will defer from that of the original curve. Similar results can be derived

for $\lambda_2 = -2$, i.e. for the collinearity of the control points $\mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4$ (c.f. Fig. 2.2).

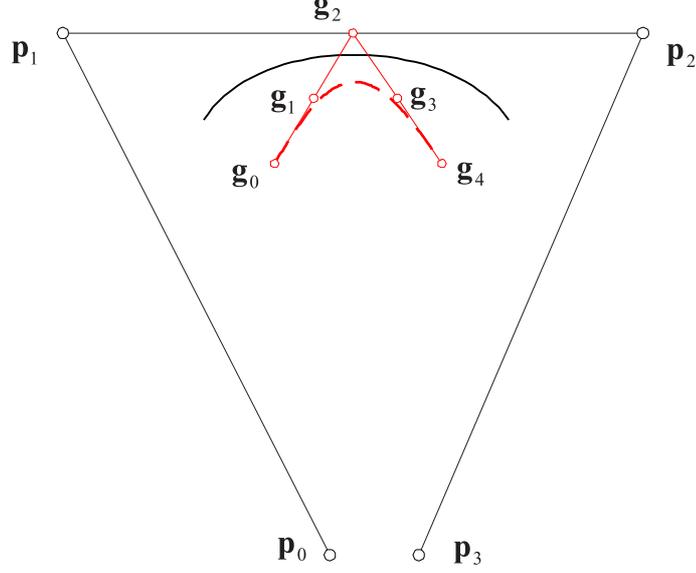


Figure 2.2. The original B-spline curve and the curve of Han for $\lambda_1 = \lambda_2 = -2$ with two sets of collinear control points $\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2$ and $\mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4$, consequently with zero curvature at \mathbf{g}_0 and \mathbf{g}_4

The linear blending description instantly yields, that altering the shape parameter any fixed point $\mathbf{c}(\lambda_1, \lambda_2, t_0)$ of the curve will move along a line segment, with endpoints $\mathbf{c}(a, \lambda_2, t_0)$ and $\mathbf{c}(b, \lambda_2, t_0)$ independently of the range of $[a, b]$ in which λ_1 varies.

If $\lambda_1 = \lambda_2 = 4$ (c.f. lower right in Fig. 2.1), i.e. if the curve interpolates both \mathbf{p}_1 and \mathbf{p}_2 , the curvature of the curve vanishes at $t = 0.5$. In order to prove this statement we consider the Bézier form (2.4) of the curve and its discriminant curve that corresponds to the control point \mathbf{g}_0 (c.f. [56]). As is shown in [56], this discriminant is of the form

$$\mathbf{s}_0(t) = \mathbf{g}_1 + \sum_{j=1}^3 \binom{3}{j} \left(\frac{t}{1-t}\right)^j (\mathbf{g}_{j+1} - \mathbf{g}_j) \quad (2.5)$$

We have to show that the tangent line of this discriminant at its point $t = 0.5$ passes through the control point $\mathbf{g}_0(4, 4) = \mathbf{p}_1$. It is enough to prove that vectors $\mathbf{s}_0(0.5) - \mathbf{p}_1$ and $\dot{\mathbf{s}}_0(0.5) = \frac{d}{dt}\mathbf{s}_0(t)|_{t=0.5}$ are parallel.

After substitution and some rearrangement we obtain

$$\begin{aligned} \mathbf{s}_0(0.5) - \mathbf{p}_1 &= \frac{11}{4}(\mathbf{p}_2 - \mathbf{p}_1) + \frac{1}{4}(\mathbf{p}_0 - \mathbf{p}_3) \\ \dot{\mathbf{s}}_0(0.5) &= \frac{33}{2}(\mathbf{p}_2 - \mathbf{p}_1) + \frac{3}{2}(\mathbf{p}_0 - \mathbf{p}_3) \end{aligned}$$

from which it is obvious that

$$(\mathbf{s}_0(0.5) - \mathbf{p}_1) \times \dot{\mathbf{s}}_0(0.5) = \mathbf{0}$$

that completes the proof.

2.1.4 Constrained shape modification

Control points \mathbf{g}_i , ($i = 0, \dots, 4$) can be written in the form

$$\begin{aligned}\mathbf{g}_0 &= \mathbf{b}_0 + \lambda_1 \mathbf{d}_1, \\ \mathbf{g}_1 &= \mathbf{b}_1 + \lambda_1 \mathbf{d}_1, \\ \mathbf{g}_2 &= \mathbf{b}_2, \\ \mathbf{g}_3 &= \mathbf{b}_3 + \lambda_2 \mathbf{d}_2, \\ \mathbf{g}_4 &= \mathbf{b}_4 + \lambda_2 \mathbf{d}_2,\end{aligned}$$

with directions

$$\begin{aligned}\mathbf{d}_1 &= \frac{1}{24} ((\mathbf{p}_1 - \mathbf{p}_0) - (\mathbf{p}_2 - \mathbf{p}_1)), \\ \mathbf{d}_2 &= \frac{1}{24} ((\mathbf{p}_2 - \mathbf{p}_1) - (\mathbf{p}_3 - \mathbf{p}_2)).\end{aligned}$$

Thus, curve (4) has the form

$$\mathbf{c}(\lambda_1, \lambda_2, t) = \sum_{i=0}^4 \mathbf{b}_i N_i^4(t) + \lambda_1 \mathbf{d}_1 (N_0^4(t) + N_1^4(t)) + \lambda_2 \mathbf{d}_2 (N_3^4(t) + N_4^4(t)). \quad (2.6)$$

From this we can see that

- if λ_1 is altered points of the curve move along straight lines that are parallel to \mathbf{d}_1 ;
- if λ_2 is altered points of the curve move along straight lines that are parallel to \mathbf{d}_2 ;
- if λ_1 and λ_2 are simultaneously altered, points of the curve move on a plane that is parallel to the directions \mathbf{d}_1 and \mathbf{d}_2 , provided $\mathbf{d}_1 \not\parallel \mathbf{d}_2$. If $\mathbf{d}_1 \parallel \mathbf{d}_2$ points of the curve move parallel to this common direction.

For the parallelism of directions \mathbf{d}_1 and \mathbf{d}_2 the coplanarity of control points \mathbf{p}_i , ($i = 0, \dots, 3$) is necessary, moreover the locus of control point \mathbf{p}_3 is the straight line indicated in Fig. 2.3.

The knowledge of path (curves or surfaces along which points of the modified curve move when shape parameters are altered) enables us to perform constrained shape modifications. Further practical computational techniques to modify the curve in a way that it will pass through a given point are described in [61].

2.2 C-curves

C-curves are extensions of the widely used cubic spline curves and are introduced by [117] applying the basis $\{\sin t, \cos t, t, 1\}$. In the case of C-B-splines this extension coincides with the helix splines defined by [93]. These tools provide exact representations of several important curves and

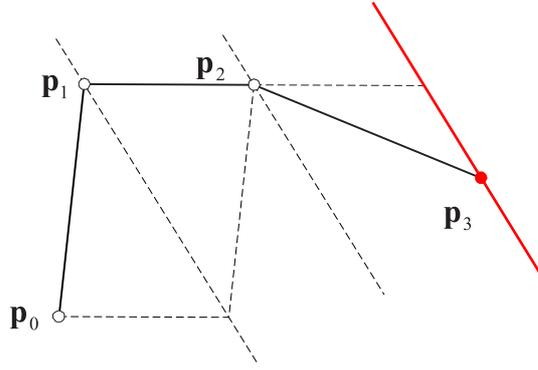


Figure 2.3. The locus of \mathbf{p}_3 for the parallelism of directions \mathbf{d}_1 and \mathbf{d}_2 (the three parallel lines are equally spaced)

surfaces such as the circle and the cylinder [117], the ellipse [119], the sphere [82], the cycloid and the helix [78]. Further properties of C-curves have been studied by [79] and by [114].

C-curves are all defined on the interval $t \in [0, \alpha]$, where $\alpha \in (0, \pi]$ is a given real number. Since α appears in all the basis functions, it heavily affects the shape of the curve. While it is already proved [117], that the limiting case $\alpha \rightarrow 0$ is a cubic polynomial curve, the effects of the modification of α have not been described yet. The aim of this subsection is to give a geometric interpretation of the change of α for C-Bézier and C-B-spline curves, based on [41].

Modifying one or more data of a given spline curve, the points of the curve will move on certain curves called paths, as we have seen in the case of B-spline curves in the preceding chapter. If the parameter α of a C-curve is altered, the points of the curve obviously change their positions as well. In this subsection these paths of C-Bézier and C-B-spline curves will be discussed. These paths can closely be approximated by lines and have some nice geometric properties which may yield to a better understanding of the role of α in terms of the shape of these curves.

2.2.1 Paths of C-Bézier curves and their extensions

Consider the C-Bézier curve (c.f. [117]):

$$\mathbf{b}(t, \alpha) = \sum_{i=0}^3 Z_i(t, \alpha) \mathbf{p}_i, \quad t \in [0, \alpha], \alpha \in (0, \pi]$$

where the basis functions are defined as:

$$\begin{aligned} M &= \begin{cases} 1 & \text{if } \alpha = \pi, \\ \frac{\sin(\alpha)}{\alpha - 2 \frac{\alpha - \sin(\alpha)}{1 - \cos(\alpha)}} & \text{otherwise} \end{cases} \\ Z_0(t, \alpha) &= \frac{(\alpha - t) - \sin(\alpha - t)}{\alpha - \sin(\alpha)} \\ Z_1(t, \alpha) &= M \left(\frac{1 - \cos(\alpha - t)}{1 - \cos(\alpha)} - \frac{(\alpha - t) - \sin(\alpha - t)}{\alpha - \sin(\alpha)} \right) \end{aligned} \quad (2.7)$$

$$Z_2(t, \alpha) = M \left(\frac{1 - \cos(t)}{1 - \cos(\alpha)} - \frac{t - \sin(t)}{\alpha - \sin(\alpha)} \right)$$

$$Z_3(t, \alpha) = \frac{t - \sin(t)}{\alpha - \sin(\alpha)}.$$

We would like to describe the movement of a single point of the curve as the parameter α changes. Altering this parameter we receive a family of C-Bézier curves with family parameter α . Due to the changing domain of definition there is not much sense to examine a point of these curves with fixed parameter t . Instead we consider the point at each curve associated to the parameter $(\alpha/ratio)$, where $ratio \in [1, \infty)$ is a fixed value. This parameter changes from curve to curve but if the domain of definition $[0, \alpha]$ would be normalized to $[0, 1]$ for each α , then the specified parameter $(\alpha/ratio)$ would have been transformed to the constant value $(1/ratio)$. This way we can define the *relative α -paths* of the family of C-Bézier curves:

$$\mathbf{s}(\alpha, ratio) = \sum_{i=0}^3 Z_i(\alpha/ratio) \mathbf{p}_i, \quad \alpha \in (0, \pi]; ratio \in [1, \infty)$$

where α is the running parameter along the path, while $ratio$ is the parameter of the path among the family of paths (see Fig.2.4).

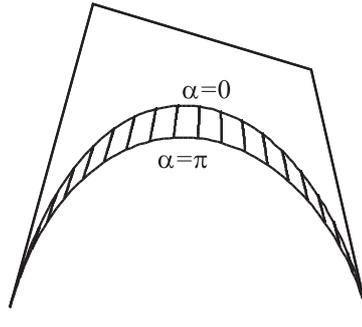


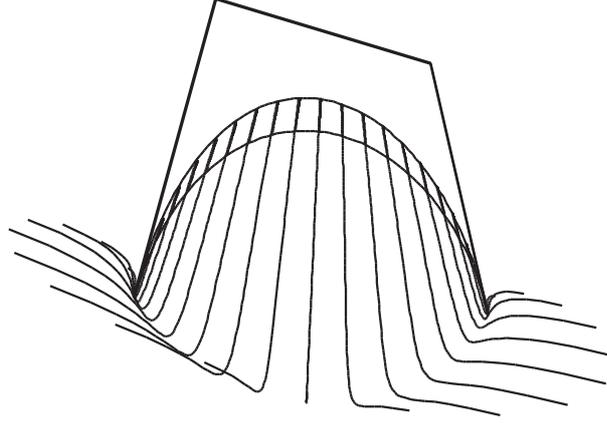
Figure 2.4. Two C-Bézier curves defined by the same control polygon and their relative α -paths

Note, that the basis functions of the original C-Bézier curve are symmetric in t for the parameter $t = \alpha/2$, thus the relative α -paths also have a symmetric property in $ratio$ for the parameter $ratio = 2$. The relative α -path associated to $ratio = 2$ can be described by the functions

$$Z_0(\alpha, 2) = Z_3(\alpha, 2) = \frac{(\alpha/2) - \sin(\alpha/2)}{\alpha - \sin(\alpha)}$$

$$Z_1(\alpha, 2) = Z_2(\alpha, 2) = M \left(\frac{1 - \cos(\alpha/2)}{1 - \cos(\alpha)} - \frac{(\alpha/2) - \sin(\alpha/2)}{\alpha - \sin(\alpha)} \right)$$

which obviously yields that this path is a part of the line connected the midpoints of $\mathbf{p}_0\mathbf{p}_3$ and $\mathbf{p}_1\mathbf{p}_2$. Paths associated to $\alpha \neq 2$ are not lines as one can easily observe by the mathematical extension of the paths (see Fig 2.5.). This extension is defined by the points

Figure 2.5. Extension of the paths for $\alpha \geq \pi$

$$\mathbf{s}(\alpha, \text{ratio}) = \sum_{i=0}^3 Z_i(\alpha/\text{ratio}) \mathbf{p}_i, \quad \text{ratio} \in [1, \infty)$$

for $\alpha \geq \pi$. We have to emphasize that these points do not belong to any C-Bézier curves and the substitution of these values of α is merely a mathematical extension. Similar extension have been successfully used for paths of B-spline curves by Hoffmann and Juhász in [37].

The paths, as we have seen are not lines, but in the original interval $\alpha \in (0, \pi]$ they can closely be approximated by lines. The approximate line of the path $\mathbf{s}(\alpha, \text{ratio})$ can be defined by the joint segment of the point $\mathbf{s}(\pi, \text{ratio})$ and $\mathbf{s}(0, \text{ratio})$ (more precisely, since α cannot be equal to 0, we consider the point obtained by $\alpha \rightarrow 0$ in this latter case).

2.2.2 Paths of C-B-spline curves and their approximate lines

C-B-spline curves are also introduced by [117] who also provided the following formula of this curve in [119] (for the sake of simplicity here we consider only four control points with a single C-B-spline arc):

$$\mathbf{b}(t, \alpha) = \sum_{i=0}^3 B_i(t, \alpha) \mathbf{p}_i, \quad t \in [0, \alpha], \alpha \in (0, \pi]$$

where the basis functions are defined as:

$$\begin{aligned} B_0(t, \alpha) &= \frac{(\alpha - t) - \sin(\alpha - t)}{2\alpha(1 - \cos \alpha)} \\ B_3(t, \alpha) &= \frac{t - \sin t}{2\alpha(1 - \cos \alpha)} \\ B_1(t, \alpha) &= B_3(t, \alpha) - 2B_0(t, \alpha) + \frac{2(\alpha - t)(1 - \cos \alpha)}{2\alpha(1 - \cos \alpha)} \\ B_2(t, \alpha) &= B_0(t, \alpha) - 2B_3(t, \alpha) + \frac{2t(1 - \cos \alpha)}{2\alpha(1 - \cos \alpha)}. \end{aligned} \tag{2.8}$$

Relative α -paths $\mathbf{s}(\alpha, \text{ratio})$ of C-B-spline curves can analogously be defined to the case of C-Bézier curves. Mathematical extension of these paths for $\alpha \geq \pi$ is also similar to that one we

have seen in the previous section (see Fig. 2.6). The path associated to $ratio = 2$ is a line again, due to the equalities

$$\begin{aligned} \mathbf{B}_0 &= \mathbf{B}_3 = \frac{2 \sin(\alpha/2) - \alpha}{4\alpha(\cos \alpha - 1)} \\ \mathbf{B}_1 &= \mathbf{B}_2 = \frac{-2 \sin(\alpha/2) - \alpha + 2\alpha \cos \alpha}{4\alpha(\cos \alpha - 1)}. \end{aligned}$$

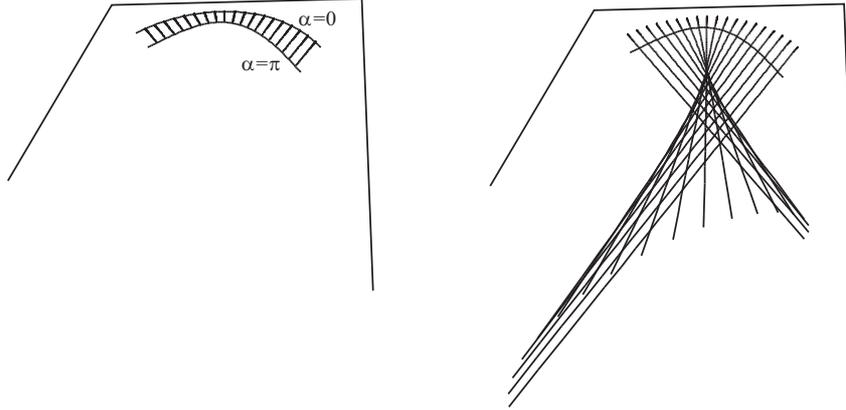


Figure 2.6. Relative α -paths of a C-B-spline arc and their extensions

Just as for C-Bézier curves, apart from the case $ratio = 2$ these paths are not lines but can be approximated by lines. The approximate line of the path $\mathbf{s}(\alpha, ratio)$ can be defined by the joint segment of the point $\mathbf{s}(\pi, ratio)$ and $\mathbf{s}(0, ratio)$.

If $\alpha = \pi$ and $t = \pi/ratio$, then we obtain:

$$\begin{aligned} \mathbf{B}_0(\pi/ratio, \pi) &= \frac{ratio \sin(\pi/ratio) + \pi - \pi ratio}{-4\pi ratio} \\ \mathbf{B}_1(\pi/ratio, \pi) &= \frac{-ratio \sin(\pi/ratio) + \pi - 2\pi ratio}{-4\pi ratio} \\ \mathbf{B}_2(\pi/ratio, \pi) &= \frac{-ratio \sin(\pi/ratio) - \pi - \pi ratio}{-4\pi ratio} \\ \mathbf{B}_3(\pi/ratio, \pi) &= \frac{ratio \sin(\pi/ratio) - \pi}{-4\pi ratio}, \end{aligned} \tag{2.9}$$

while applying the limit $\alpha \rightarrow 0$ for equations (2.8):

$$\begin{aligned} \mathbf{B}_{0lim} &= \frac{ratio^3 - 3ratio^2 + 3ratio - 1}{6ratio^3} \\ \mathbf{B}_{1lim} &= \frac{4ratio^2 - 6ratio + 3}{6ratio^3} \\ \mathbf{B}_{2lim} &= \frac{ratio^3 + 3ratio^2 + 3ratio - 3}{6ratio^3} \\ \mathbf{B}_{3lim} &= \frac{1}{6ratio^3}. \end{aligned} \tag{2.10}$$

Connecting the points $\sum_{i=0}^3 B_i(\pi/ratio, \pi)\mathbf{p}_i$ and $\sum_{i=0}^3 B_{i \lim}(ratio)\mathbf{p}_i$ the result is a family of lines with family parameter *ratio*. The intersection curve of the symmetric lines has the same property as in the C-Bézier case (here we suppose that the control points are coplanar so the intersection curve exists). For the proof of the following theorems and further details of approximate lines, see [41].

Theorem 2.4. *The intersection curve of the symmetric lines is a straight line segment if $\mathbf{p}_0\mathbf{p}_3$ is parallel to $\mathbf{p}_1\mathbf{p}_2$. Furthermore, the line segment is just on the line connecting the midpoints of $\mathbf{p}_0\mathbf{p}_3$ and $\mathbf{p}_1\mathbf{p}_2$.*

The approximate lines of the relative α -paths of C-B-spline curves have a property which has no analogue in the C-Bézier case: for a certain position of control points all the lines are parallel (see Fig. 2.7).

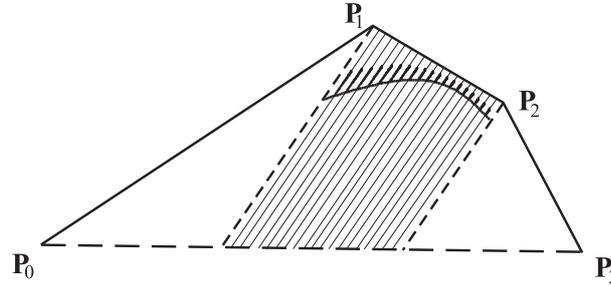


Figure 2.7. In a special case paths can be replaced by parallel lines

Theorem 2.5. *Dividing the line $\mathbf{p}_0\mathbf{p}_3$ into three equal parts by points $\mathbf{q}_1, \mathbf{q}_2$, the approximate lines are parallel if the line $\mathbf{p}_1\mathbf{q}_1$ is parallel to the line $\mathbf{p}_2\mathbf{q}_2$.*

2.3 FB-spline curves

The objective of this section is to examine the effect of the modification of control points and shape parameters on the shape of FB-spline curves, and to provide shape modification methods based on them, based on the results published in [42]. These methods are indispensable for the application of FB-spline curves in design. After basic definitions, we have collected control point based methods, then we study the influence of shape parameters, and endpoint interpolation. Throughout the section, we use the definition of FB-spline curves specified in [121].

Definition 2.6. Given control points $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_{n+1}$, ($n \geq 2$) and parameters C_1, C_2, \dots, C_n , ($C_i \in [0, \infty)$). The curve that consists of arcs

$$\mathbf{p}_i(\tau) = N_{i,0}(\tau)\mathbf{b}_{i-1} + N_{i,1}(\tau)\mathbf{b}_i + N_{i,2}(\tau)\mathbf{b}_{i+1} + N_{i,3}(\tau)\mathbf{b}_{i+2} \quad (2.11)$$

$$\tau \in [0, 1], (i = 1, 2, \dots, n - 1)$$

is called FB-spline curve, where, using the abbreviations

$$spr(x) = \frac{x - \sin(x)}{x^3}, \quad cpr(x) = \frac{1 - \cos(x)}{x^2},$$

dc_933_14

$$sph(x) = \frac{\sinh(x)-x}{x^3}, \quad cph(x) = \frac{\cosh(x)-1}{x^2}$$

the basis functions are

$$N_{i,0}(\tau) = \begin{cases} \frac{spr(2 \arccos(C_i)(1-\tau))}{2cpr(2 \arccos(C_i))} (1-\tau)^3 & \text{if } C_i \leq 1 \\ \frac{sph(2 \operatorname{arccosh}(C_i)(1-\tau))}{2cph(2 \operatorname{arccosh}(C_i))} (1-\tau)^3 & \text{if } C_i > 1 \end{cases}$$

$$N_{i,3}(\tau) = \begin{cases} \frac{spr(2 \arccos(C_{i+1})\tau)}{2cpr(2 \arccos(C_{i+1}))} \tau^3 & \text{if } C_{i+1} \leq 1 \\ \frac{sph(2 \operatorname{arccosh}(C_{i+1})\tau)}{2cph(2 \operatorname{arccosh}(C_{i+1}))} \tau^3 & \text{if } C_{i+1} > 1 \end{cases}$$

$$N_{i,1}(\tau) = N_{i,3}(\tau) - 2N_{i,0}(\tau) + (1-\tau)$$

$$N_{i,2}(\tau) = N_{i,0}(\tau) - 2N_{i,3}(\tau) + \tau.$$

Scalars C_i are called shape parameters. Practically if these shape parameters are all greater than 1, then we get a curve between the classical B-spline curve and its control polygon, which is identical to the CB-spline curve. If the shape parameters are all less than 1, then we get a curve "below" the classical B-spline curve which is identical to the HB-spline curve. The definition described above allow us to get curves which somehow mix these two possibilities. The classical B-spline curve can be obtained as a limit case (all the shape parameters $C_i = 1$) but here we have to approximate the applied trigonometric functions.

If $x = 0$ or $x \approx 0$, i.e. $C_i = 1$ or $C_i \approx 1$, we use expansions

$$spr(x) = \frac{1}{3!} - \frac{x^2}{5!} + \frac{x^4}{7!} - \dots + (-1)^{n-1} \frac{x^{2n-2}}{(2n+1)!} + \dots$$

$$cpr(x) = \frac{1}{2!} - \frac{x^2}{4!} + \frac{x^4}{6!} - \dots + (-1)^{n-1} \frac{x^{2n-2}}{(2n)!} + \dots$$

$$sph(x) = \frac{1}{3!} + \frac{x^2}{5!} + \frac{x^4}{7!} + \dots + \frac{x^{2n-2}}{(2n+1)!} + \dots$$

$$cph(x) = \frac{1}{2!} + \frac{x^2}{4!} + \frac{x^4}{6!} + \dots + \frac{x^{2n-2}}{(2n)!} + \dots$$

The arc $\mathbf{p}_i(\tau)$ can also be described in the form

$$\begin{aligned} \mathbf{p}_i(\tau) &= N_{i,0}(\tau) (\mathbf{b}_{i-1} - 2\mathbf{b}_i + \mathbf{b}_{i+1}) + N_{i,3}(\tau) (\mathbf{b}_i - 2\mathbf{b}_{i+1} + \mathbf{b}_{i+2}) \\ &\quad + (1-\tau) \mathbf{b}_i + \tau \mathbf{b}_{i+1}. \end{aligned} \quad (2.12)$$

Varying one of the defining data (control point or shape parameter) of the curve, its points move along curves that we also call *path*.

2.3.1 Control point based methods

Move a curve point to a specified location Any control point of curve (2.11) affects at most four consecutive arcs. When a control point of an FB-spline curve is translated, paths of curve points are straight line segments that are parallel to the translation vector. This is the property of all curves that are combinations of control points and basis functions, therefore we discuss it briefly only for the sake of completeness.

If the control point \mathbf{b}_i is translated by the vector \mathbf{d} , the shape of the arcs $\mathbf{p}_j(\tau)$, ($j = i + k$, $k = -2, -1, 0, 1$) are modified in the form

$$\tilde{\mathbf{p}}_j(\tau) = \mathbf{p}_j(\tau) + N_{j,1-k}(\tau) \mathbf{d}.$$

If we want to move the curve point $\mathbf{p}_j(\tau)$ with prescribed parameter value τ to an arbitrarily chosen point \mathbf{q} (there is no restriction to the location of \mathbf{q} at all) by the translation of control point \mathbf{b}_i , the translation vector is

$$\mathbf{d} = \frac{1}{N_{j,1-k}(\tau)} (\mathbf{q} - \mathbf{p}_j(\tau)).$$

Multiple control points We examine the effect of coinciding consecutive control points on the shape of the curve.

In case of double control point we assume that $\mathbf{b}_i = \mathbf{b}_{i+1}$. The i th arc becomes

$$\mathbf{p}_i(\tau) = \mathbf{b}_i + N_{i,0}(\tau) (\mathbf{b}_{i-1} - \mathbf{b}_i) + N_{i,3}(\tau) (\mathbf{b}_{i+2} - \mathbf{b}_i).$$

$\tau = 0$ implies $N_{i,3}(0) = 0$, thus

$$\mathbf{p}_i(0) = \mathbf{b}_i + N_{i,0}(0) (\mathbf{b}_{i-1} - \mathbf{b}_i),$$

i.e., the beginning point of the arc is on the segment $\mathbf{b}_{i-1}\mathbf{b}_i$ which segment is the tangent at this point.

$\tau = 1$ implies $N_{i,0}(1) = 0$ and

$$\mathbf{p}_i(1) = \mathbf{b}_i + N_{i,3}(1) (\mathbf{b}_{i+2} - \mathbf{b}_i),$$

therefore the line $\mathbf{b}_{i+2}\mathbf{b}_i$ touches the arc at its endpoint which point is on the segment.

The $(i - 1)$ th arc is

$$\begin{aligned} \mathbf{p}_{i-1}(\tau) &= N_{i-1,0}(\tau) (\mathbf{b}_{i-2} - 2\mathbf{b}_{i-1} + \mathbf{b}_i) + N_{i-1,3}(\tau) (\mathbf{b}_{i-1} - \mathbf{b}_i) + \\ &\quad (1 - \tau) \mathbf{b}_{i-1} + \tau \mathbf{b}_i. \end{aligned}$$

At $\tau = 1$

$$\mathbf{p}_{i-1}(1) = \mathbf{b}_i + N_{i-1,3}(1) (\mathbf{b}_{i-1} - \mathbf{b}_i). \quad (2.13)$$

Utilizing that $N_{i,0}(0) = N_{i-1,3}(1)$ we can see that line $\mathbf{b}_{i-1}\mathbf{b}_i$ is the tangent at the point of

joint.

The $(i + 1)$ th arc is

$$\mathbf{p}_{i+1}(\tau) = N_{i+1,0}(\tau)(\mathbf{b}_{i+2} - \mathbf{b}_i) + N_{i+1,3}(\tau)(\mathbf{b}_i - 2\mathbf{b}_{i+2}) + \mathbf{b}_{i+3} \\ (1 - \tau)\mathbf{b}_i + \tau\mathbf{b}_{i+2},$$

at $\tau = 0$

$$\mathbf{p}_{i+1}(0) = \mathbf{b}_i + N_{i+1,0}(0)(\mathbf{b}_{i+2} - \mathbf{b}_i),$$

moreover $N_{i+1,0}(0) = N_{i,3}(1)$, i.e. at this point of joint the tangent is the control polygon side $\mathbf{b}_{i+2}\mathbf{b}_i$ (cf. Fig. 2.8).

This property enables us to specify shape parameters C_i and C_{i+1} intuitively by the direct specification of the point of contact $\mathbf{p}_i(0)$ and $\mathbf{p}_i(1)$, respectively.

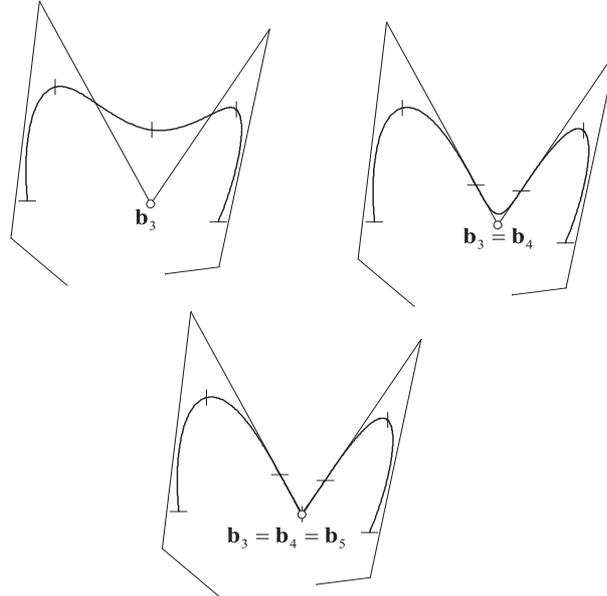


Figure 2.8. FB-splines with multiplicity 1, 2 and 3 of control point \mathbf{b}_3

In case of triple control point we assume that $\mathbf{b}_i = \mathbf{b}_{i+1} = \mathbf{b}_{i+2}$ (cf. Fig. 2.8) which implies

$$\mathbf{p}_i(\tau) = \mathbf{b}_i + N_{i,0}(\tau)(\mathbf{b}_{i-1} - \mathbf{b}_i).$$

This means that the i th arc is a segment with endpoint \mathbf{b}_i of the control polygon side $\mathbf{b}_{i-1}\mathbf{b}_i$.

The endpoint of the $(i - 1)$ th arc is (2.13), since in the evaluation of this arc the considered control point is of multiplicity two. Therefore, arcs $\mathbf{p}_{i-1}(\tau)$ and $\mathbf{p}_i(\tau)$ form a C^2 continuous straight line segment and curved arc. Similar results can be derived for arcs $\mathbf{p}_{i+1}(\tau)$ and $\mathbf{p}_{i+2}(\tau)$.

By means of this property one can describe C^2 continuously joining straight line segments and curved arcs with an FB-spline curve.

Finally, in case of quadruple control point, the assumption $\mathbf{b}_i = \mathbf{b}_{i+1} = \mathbf{b}_{i+2} = \mathbf{b}_{i+3}$ implies

$$\mathbf{p}_i(\tau) = \mathbf{b}_i + N_{i,0}(\tau)(\mathbf{b}_{i-1} - \mathbf{b}_i)$$

$$\mathbf{p}_{i+1}(\tau) = \mathbf{b}_i$$

$$\mathbf{p}_{i+2}(\tau) = \mathbf{b}_i + N_{i+2,3}(\tau)(\mathbf{b}_{i+4} - \mathbf{b}_i)$$

that is arc $\mathbf{p}_{i+1}(\tau)$ degenerates to a single point, arcs $\mathbf{p}_i(\tau)$ and $\mathbf{p}_{i+2}(\tau)$ are FB-spline arcs of a triple control point. At point \mathbf{b}_i the FB-spline curve is of C^0 continuity.

2.3.2 Shape parameter based methods

Modifying a single shape parameter Parameter C_i affects only arcs $\mathbf{p}_{i-1}(\tau)$ and $\mathbf{p}_i(\tau)$. We fix τ and C_{i+1} and let C_i vary in the range $[0, \infty)$. In expression (2.12) only $N_{i,0}(\tau)$ depends on C_i , thus paths are straight line segments that are parallel to the vector $(\mathbf{b}_{i-1} - \mathbf{b}_i) + (\mathbf{b}_{i+1} - \mathbf{b}_i)$. Therefore, path of points of the affected arcs form a cylinder with base curve $\mathbf{p}_{i-1}(\tau)$, $\mathbf{p}_i(\tau)$ and generator direction $(\mathbf{b}_{i-1} - 2\mathbf{b}_i + \mathbf{b}_{i+1})$ (Fig. 2.9).

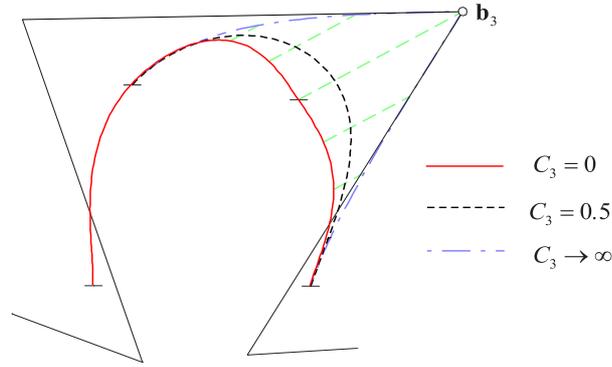


Figure 2.9. Paths of an FB-spline curve obtained by the alteration of shape parameter C_3

Limiting positions of the affected arcs are at values $C_i = 0$ and $C_i \rightarrow \infty$. In case of the arc $\mathbf{p}_{i-1}(\tau)$

$$\lim_{C_i \rightarrow \infty} N_{i-1,3}(\tau) = 0,$$

therefore

$$\lim_{C_i \rightarrow \infty} N_{i-1,1}(\tau) = (1 - \tau) - 2N_{i-1,0}(\tau), \quad \lim_{C_i \rightarrow \infty} N_{i-1,2}(\tau) = N_{i-1,0}(\tau) + \tau$$

from which the limiting position of the arc is

$$\mathbf{p}_{i-1}^{C_i \infty}(\tau) = N_{i-1,0}(\tau)(\mathbf{b}_{i-2} - 2\mathbf{b}_{i-1} + \mathbf{b}_i) + (1 - \tau)\mathbf{b}_{i-1} + \tau\mathbf{b}_i.$$

Its derivative with respect to τ is

$$\dot{\mathbf{p}}_{i-1}^{C_i \infty}(\tau) = \dot{N}_{i-1,0}(\tau)(\mathbf{b}_{i-2} - 2\mathbf{b}_{i-1} + \mathbf{b}_i) + \mathbf{b}_i - \mathbf{b}_{i-1}.$$

In case of $\tau = 1$

$$\mathbf{p}_{i-1}^{C_i^\infty}(1) = \mathbf{b}_i, \quad \dot{\mathbf{p}}_{i-1}^{C_i^\infty}(1) = \mathbf{b}_i - \mathbf{b}_{i-1},$$

thus the endpoint of the arc $\mathbf{p}_{i-1}^{C_i^\infty}(\tau)$ is the control point \mathbf{b}_i where the tangent is the side $\mathbf{b}_{i-1}\mathbf{b}_i$ of the control polygon.

By analogous considerations we obtain that the beginning point of the arc $\mathbf{p}_i^{C_i^\infty}(\tau)$ is \mathbf{b}_i , where the tangent is the control polygon side $\mathbf{b}_i\mathbf{b}_{i+1}$.

The cylinder of paths generated by the alteration of C_i , always passes through the control point \mathbf{b}_i and the tangent plane along its incident generator is spanned by control points \mathbf{b}_{i-1} , \mathbf{b}_i and \mathbf{b}_{i+1} .

Shape control by modifying a single shape parameter In practical CAGD systems constrained modification of a curve is essential, e.g. moving a curve point to a specified location. Based on the previous observations, by the alteration of a shape parameter we can modify an FB-spline curve in such a way that a selected point of the modified curve will pass through a specified point, but using purely shape parameter alteration, the target point must be on a well-defined line segment (see Fig.2.10). Steps of the procedure in an implementation are as follows:

- select the point \mathbf{r} to be moved on the arc, i.e. fix the parameter τ , ($\mathbf{r} = \mathbf{p}(\tau)$);
- chose the shape parameter to be modified (there are two options, in the rest we assume that shape parameter C_i has been chosen);
- the system displays the path of the selected point, i.e. the straight line segment bounded by points \mathbf{o} and $\mathbf{o} + M_0(\tau)\mathbf{e}$, where

$$\begin{aligned} \mathbf{o} &= (1 - \tau)\mathbf{b}_i + \tau\mathbf{b}_{i+1} + N_{i,3}(\tau)(\mathbf{b}_i - 2\mathbf{b}_{i+1} + \mathbf{b}_{i+2}) \\ \mathbf{e} &= \mathbf{b}_{i-1} - 2\mathbf{b}_i + \mathbf{b}_{i+1} \\ M_0(\tau) &= \frac{\pi(1 - \tau) - \sin(\pi(1 - \tau))}{4\pi} \end{aligned}$$

- specify the new position \mathbf{q} of the selected point on the path.

The new position can be written in the form

$$\mathbf{q} = \mathbf{o} + \lambda\mathbf{e}, \quad \lambda \in [0, M_0(\tau)].$$

The $N_{i,0}(\tau) = \lambda$ trigonometric equation has to be solved for the unknown shape parameter C_i . There will always be a unique solution due to the geometric constraints. The solution is an HB-spline if $\lambda \in [0, \lim_{C_i \rightarrow \infty} N_{i,0}(\tau))$, and a CB-spline if $\lambda \in [\lim_{C_i \rightarrow \infty} N_{i,0}(\tau), M_0(\tau))$. C_i will be in the ranges $(1, \infty)$ and $[0, 1]$, respectively. The high accuracy computation of the root is essential for the satisfactory geometric result. In our experience, the false position (regula falsi) root finding method is fast and accurate enough.

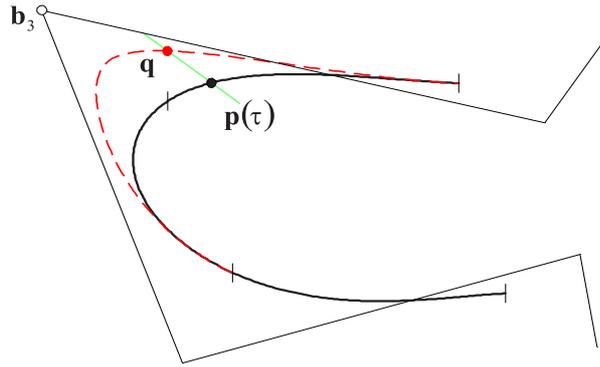


Figure 2.10. A selected point $\mathbf{p}(\tau)$ of the FB-spline curve is moved to the given position \mathbf{q} by modifying the shape parameter C_3 ; the green line is the permissible positions of \mathbf{q}

Fig.2.10 illustrates shape modification subject to positional constraints by means of a single shape parameter.

Certainly, such a shape modification objective can also be obtained by control point repositioning. However, the shape of resulted curves of different methods are not the same, as we can see in Fig. 2.11. The advantage of shape parameter alteration is twofold. The alteration affects only two arcs (not four, like in case of control point repositioning), and the modified curve is always within the convex hull of the original control points.

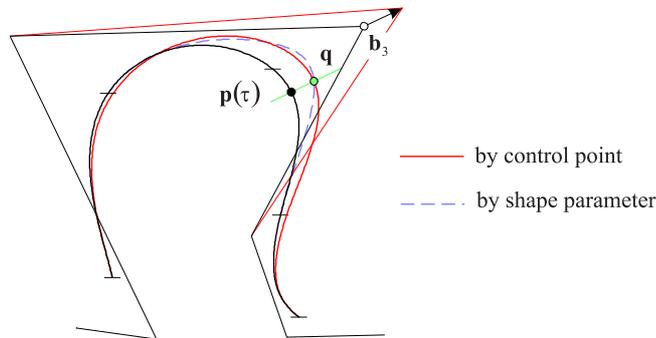


Figure 2.11. Point $\mathbf{p}(\tau)$ of the FB-spline curve is moved to \mathbf{q} by the reposition of control point \mathbf{b}_3 (red) and by the shape parameter C_3 (dashed blue)

As another practical method one can modify a curve by passing through a point, without specifying the corresponding parameter value. In this case the point through which we want the modified curve to pass can be specified on the cylinder of paths. (In case of plane curves this cylinder degenerates to a plane region.) Using the generator that passes through the specified point, we can determine the corresponding parameter τ of the curve. Then, we can proceed according to the previous Subsection.

Simultaneous modification of two shape parameters We assume that shape parameters C_i and C_{i+1} are modified simultaneously. Both parameters affect only the arc $\mathbf{p}_i(\tau)$. It is obvious from expression (2.12) that any point $\mathbf{p}_i(\tau)$ of the arc moves within a parallelogram. Sides of this parallelogram are parallel to the directions $\mathbf{b}_{i-1} - 2\mathbf{b}_i + \mathbf{b}_{i+1}$ and $\mathbf{b}_i - 2\mathbf{b}_{i+1} + \mathbf{b}_{i+2}$, and the endpoints of one of its diagonals are $(1 - \tau)\mathbf{b}_i + \tau\mathbf{b}_{i+1}$ and $\mathbf{p}_i(\tau)$ with $C_i = C_{i+1} = 0$. Based on these observations, we can develop shape modification methods simultaneously altering two shape parameters. Shape control by modifying two shape parameters and endpoint interpolation are also discussed in detail in [42].

2.4 A trigonometric curve with exponential shape parameters

In [31] a new trigonometric curve, which can be considered as a kind of generalisation of the well-known quartic Bézier curve, has been introduced using five new trigonometric blending functions with two exponential shape parameters. In this section we will study this curve, its generalizations and geometric properties, based on [46]. The definition of the curve [31] is as follows.

Definition 2.7. Let control points \mathbf{p}_i are given. Then the trigonometric Bézier curve with shape parameters $\alpha, \beta \in [2, \infty)$ is defined as

$$\mathbf{c}(t, \alpha, \beta) = \sum_{i=0}^4 T_i(t, \alpha, \beta) \mathbf{p}_i \quad \text{for } t \in [0, \pi/2],$$

where

$$\begin{aligned} T_0(t, \alpha, \beta) &= (1 - \sin t)^\alpha \\ T_1(t, \alpha, \beta) &= \alpha \sin t (1 - \sin t)^{\alpha-1} \\ T_3(t, \alpha, \beta) &= \beta \cos t (1 - \cos t)^{\beta-1} \\ T_4(t, \alpha, \beta) &= (1 - \cos t)^\beta \\ T_2(t, \alpha, \beta) &= 1 - \sum_{i \neq 2} T_i(t, \alpha, \beta). \end{aligned}$$

The authors prove in [31] that these blending functions form a basis. Nonnegativity, partition of unity and some properties of the curve itself are also shown. Unfortunately, the number of control points of the curve is restricted to be five. However, it is a well-known fact, that the concatenation of short Bézier arcs is not ideal for the design of complex shapes due to the positional restrictions.

The aim of this section is to extend this definition into two directions. On the one hand, the generalized curve is defined for an arbitrary number of control points, preserving the same properties and having the same shape parameters as the original curve. To this aim, the function basis $T_i(t, \alpha, \beta)$, $i = 0, \dots, 4$ is generalized for arbitrary dimension. On the other hand, while the original blending functions are based on the function pair $\sin t, \cos t$ satisfying $\sin^2 t + \cos^2 t = 1$, in

this section it is substituted by a more general function pair $\varphi(t), \psi(t)$ with the property $\varphi^{n_1}(t) + \psi^{n_2}(t) = 1$, where the positive integers n_1, n_2 replace the exponents 2. In the next subsections we define the new basis functions, prove their important properties, such as nonnegativity, linear independence and partition of unity, moreover we study their total positivity. Based on these blending functions, the curve is defined and some of its properties are demonstrated.

2.4.1 New basis functions

Given positive integers n_1, n_2 , let $\varphi, \psi : [a, b] \rightarrow [0, 1]$ be increasing resp. decreasing bijective functions satisfying the relation

$$\varphi^{n_1}(t) + \psi^{n_2}(t) = 1 \quad (2.14)$$

for every $t \in [a, b]$. In particular,

$$\varphi(a) = \psi(b) = 0, \quad \varphi(b) = \psi(a) = 1, \quad (2.15)$$

both functions are continuous, and ψ is expressed in terms of φ as

$$\psi(t) = (1 - \varphi^{n_1}(t))^{\frac{1}{n_2}}.$$

According to Newton's generalized binomial theorem, assuming $\varphi(t) < 1/2$,

$$1 = (\varphi(t) + (1 - \varphi(t)))^\alpha = \sum_{k=0}^{\infty} \binom{\alpha}{k} \varphi^k(t) (1 - \varphi(t))^{\alpha-k} \quad (2.16)$$

holds for every real number α with

$$\binom{\alpha}{k} = \frac{1}{k!} \prod_{m=0}^{k-1} (\alpha - m) = \frac{(\alpha - k + 1)_k}{k!},$$

where we use the Pochhammer symbol $(x)_k = x(x+1) \dots (x+k-1)$ to represent rising factorials.

Fix $\alpha \in [n_1, \infty)$ and $\beta \in [n_2, \infty)$. By means of the terms in the above series expansion, we define functions

$$L_i(t, \alpha) = \binom{\alpha}{i} \varphi^i(t) (1 - \varphi(t))^{\alpha-i}$$

and

$$R_{n_1+n_2-j}(t, \beta) = \binom{\beta}{j} \psi^j(t) (1 - \psi(t))^{\beta-j}$$

for $i = 0, 1, \dots, n_1 - 1$ and $j = 0, 1, \dots, n_2 - 1$.

Definition 2.8. Using the functions introduced above, let

$$\mathcal{T} = \mathcal{T}(a, b; n_1, n_2; \varphi, \psi; \alpha, \beta) := \{T_j(t, \alpha, \beta)\}_{j=0}^{n_1+n_2}$$

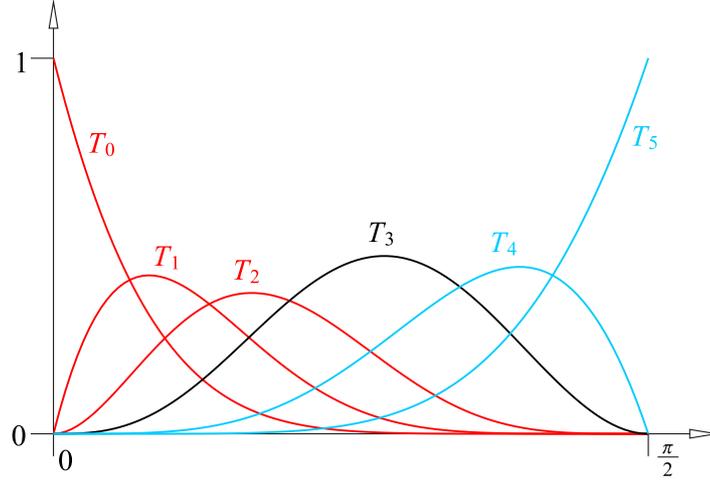


Figure 2.12. Basis functions $T_j(t, \alpha, \beta)$ with settings $n_1 = 3, n_2 = 2, \alpha = 4, \beta = 3$ and $\varphi(t) = \sin(t)$

denote the system of $n_1 + n_2 + 1$ functions given in the form

$$T_j(t, \alpha, \beta) = \begin{cases} L_j(t, \alpha) & \text{for } 0 \leq j \leq n_1 - 1, \\ R_j(t, \beta) & \text{for } n_1 + 1 \leq j \leq n_1 + n_2, \\ 1 - \sum_{i \neq n_1} T_i(t, \alpha, \beta) & \text{for } j = n_1. \end{cases}$$

It is immediate that the elements of \mathcal{T} form a partition of unity on the interval $[a, b]$. The particular choice of $a = 0, b = \pi/2, n_1 = n_2 = 2, \varphi(t) = \sin t, \psi(t) = \cos t$ results in the function system introduced by Han and Zhu (cf. Definition 2.7). Fig 2.12 illustrates basis functions $T_j(t, \alpha, \beta)$ for settings $n_1 = 3, n_2 = 2, \alpha = 4, \beta = 3$ and $\varphi(t) = \sin(t)$.

2.4.2 Nonnegativity

Using the properties of the functions φ, ψ one readily infers that

$$T_i(a, \alpha, \beta) = \begin{cases} 1 & \text{for } i = 0 \text{ and} \\ 0 & \text{otherwise,} \end{cases} \quad (2.17)$$

$$T_i(b, \alpha, \beta) = \begin{cases} 1 & \text{for } i = n_1 + n_2 \text{ and} \\ 0 & \text{otherwise,} \end{cases} \quad (2.18)$$

and that the functions T_i are nonnegative for $i \neq n_1$. Our proof that T_{n_1} is also nonnegative depends on the following inequality, which may be of independent interest.

Lemma 2.9. *For any positive integer n and real numbers $\alpha \in [n, \infty), x \in [0, 1]$ one has*

$$K_\alpha(x) := 1 - x^n - \sum_{k=0}^{n-1} \binom{\alpha}{k} x^k (1-x)^{\alpha-k} \geq 0.$$

Moreover, $K_\alpha(x) > 0$ holds for $x \in (0, 1)$ when $\alpha \in (n, \infty)$.

Proof. Fix the positive integer n . When $\alpha = n$, the left hand side is identically zero. It is also zero for arbitrary α , when $x = 0$ or $x = 1$. Therefore it is enough to show that for any fixed $x \in (0, 1)$, the function

$$f(\alpha, x) = \sum_{k=0}^{n-1} \binom{\alpha}{k} x^k (1-x)^{\alpha-k}$$

as a continuous function of α is strictly decreasing on the interval $[n, \infty)$. Using the polynomial

$$p(\alpha, x) = \sum_{k=0}^{n-1} \binom{\alpha}{k} x^k (1-x)^{n-1-k}$$

we can write $f(\alpha, x) = (1-x)^{\alpha+1-n} p(\alpha, x)$ and thus it is enough to prove that

$$\frac{d}{d\alpha} f(\alpha, x) = (1-x)^{\alpha+1-n} \left(\log(1-x) p(\alpha, x) + \frac{d}{d\alpha} p(\alpha, x) \right) < 0$$

holds for any $x \in (0, 1)$ and $\alpha \in (n, \infty)$. Given that in this domain $p(\alpha, x) > 0$ and using the power series expansion of $\log(1-x)$ valid on the interval $(-1, 1)$ this amounts to checking the inequality

$$\left(\sum_{k=1}^{\infty} \frac{x^k}{k} \right) p(\alpha, x) > \frac{d}{d\alpha} p(\alpha, x) \tag{2.19}$$

for $x \in (0, 1)$, $\alpha \in (n, \infty)$. Write

$$p(\alpha, x) = \sum_{k=0}^{n-1} c_k(\alpha) x^k, \quad \left(\sum_{k=1}^{\infty} \frac{x^k}{k} \right) p(\alpha, x) = \sum_{k=1}^{\infty} d_k(\alpha) x^k.$$

Inequality (2.19) thus follows immediately from the following two facts:

- (i) $d_k(\alpha) > 0$ for every $\alpha \geq n$ and positive integer k ;
- (ii) $d_k(\alpha) = c'_k(\alpha)$ for every $\alpha \geq n$ and positive integer $k \leq n-1$.

The key to both points is the Chu–Vandermonde convolution formula [12], which implies that

$$c_k(\alpha) = \sum_{i=0}^k \binom{\alpha}{i} (-1)^{k-i} \binom{n-1-i}{k-i} = \sum_{i=0}^k \binom{\alpha}{i} \binom{k-n}{k-i} = \binom{\alpha-n+k}{k}$$

Thus, $c_k(\alpha) > 0$ holds for every $\alpha \geq n$, $k = 0, 1, \dots, n-1$. It follows that $d_k(\alpha) > 0$ for every $\alpha \geq n$ and positive integer k , proving (i). Turning to (ii), notice that

$$d_k(\alpha) = \sum_{i=0}^{k-1} \frac{c_i(\alpha)}{k-i}.$$

Therefore (ii) amounts to proving

$$\sum_{i=0}^{k-1} \frac{1}{k-i} \binom{\alpha-n+i}{i} = \sum_{i=1}^k \frac{1}{\alpha-n+i} \binom{\alpha-n+k}{k}$$

for every $\alpha \geq n$ and positive integer $k \leq n-1$. Thus it will be enough to prove the identity

$$\sum_{i=0}^{k-1} \frac{1}{k-i} \binom{i-\beta}{i} = \sum_{i=1}^k \frac{1}{i-\beta} \binom{k-\beta}{k} \quad (2.20)$$

for positive integers k and real numbers β . This is all standard and can be done by manipulating the Chu–Vandermonde identity, but we prefer here to present a more direct approach. Notice that for any positive integer k , each side of the equation represents a polynomial in β whose degree is $k-1$. Therefore it is enough to check that with any fixed positive integer k , (2.20) holds for k different values of β . The case $\beta = 0$ being obvious, assume that $\beta = j$ where j is a positive integer not exceeding $k-1$. Then (2.20) reads as

$$\sum_{i=0}^{k-1} \frac{1}{k-i} \binom{i-j}{i} = \sum_{i=1}^k \frac{(1-j)_k}{(i-j)k!}.$$

Since in the given range $\binom{i-j}{i} = 0$ for $i \geq j$ and $\frac{(1-j)_k}{(i-j)k!} = 0$ for $i \neq j$, this equation simplifies to

$$\sum_{i=0}^{j-1} \frac{1}{k-i} \binom{i-j}{i} = \frac{(k-j)!(-1)^{j-1}(j-1)!}{k!},$$

or equivalently, to

$$\sum_{i=0}^{j-1} (-1)^i \binom{j-1}{i} \frac{(k+1-j)_j}{k-i} = (-1)^{j-1}(j-1)!.$$

Thus it will be enough to prove the identity

$$\sum_{i=0}^{j-1} (-1)^i \binom{j-1}{i} \frac{(\gamma+1-j)_j}{\gamma-i} = (-1)^{j-1}(j-1)! \quad (2.21)$$

for positive integers j and real numbers γ . For any fixed positive integer j , the left hand side represents a polynomial in γ whose degree is at most $j-1$, whereas the right hand side is independent of γ . Therefore it is enough to check that with any fixed positive integer j , (2.21) holds for j different values of γ . Let $\gamma \in \{0, 1, \dots, j-1\}$, then $\frac{(\gamma+1-j)_j}{\gamma-i} = 0$ holds for $i \neq \gamma$, and indeed

$$\begin{aligned} \sum_{i=0}^{j-1} (-1)^i \binom{j-1}{i} \frac{(\gamma+1-j)_j}{\gamma-i} \\ = (-1)^\gamma \binom{j-1}{\gamma} \gamma! (-1)^{j-1-\gamma} (j-1-\gamma)! = (-1)^{j-1} (j-1)!. \end{aligned}$$

The proof is thus complete. □ □

Proposition 2.10. *Recall that $\alpha \in [n_1, \infty), \beta \in [n_2, \infty)$. If $\alpha = n_1$ and $\beta = n_2$, then $T_{n_1}(t, \alpha, \beta)$ is identically zero on the interval $[a, b]$. Otherwise we have*

$$T_{n_1}(t, \alpha, \beta) \geq 0$$

for every $t \in [a, b]$. Apart from the endpoints $t = a$ and $t = b$, the inequality is strict.

Proof. In view of Definition 2.8 and Eq. (2.14) we can write

$$\begin{aligned} T_{n_1}(t, \alpha, \beta) &= 1 - \sum_{i=0}^{n_1-1} L_i(t, \alpha) - \sum_{i=0}^{n_2-1} R_{n_1+n_2-i}(t, \beta) \\ &= 1 - \sum_{i=0}^{n_1-1} \binom{\alpha}{i} \varphi^i(t)(1 - \varphi(t))^{\alpha-i} - \sum_{i=0}^{n_2-1} \binom{\beta}{i} \psi^i(t)(1 - \psi(t))^{\beta-i} \\ &= K_\alpha(\varphi(t)) + K_\beta(\psi(t)). \end{aligned}$$

If $\alpha = n_1$ and $\beta = n_2$, then both summands are identically zero because of the binomial theorem. The rest of the statement follows directly from Lemma 2.9. □ □

2.4.3 Linear independence

In order to prove the linear independence of the function system \mathcal{T} , initially we assume that the functions φ, ψ are sufficiently smooth and satisfy certain boundary conditions. These assumptions will be dropped later. Recall that a real function is smooth on an interval $[a, b]$ if it is continuously differentiable on $[a, b]$ up to any order. Of course at the endpoints a and b only right resp. left continuity/differentiability is assumed, but for simplicity this subtlety will be suppressed both in language and in notation. In addition we say that it is analytic around an endpoint if has an extension which is analytic on some open interval that contains the given endpoint.

For a smooth real function $f = f(t)$ the following conventions will be employed: $f^{(0)} = f$, $f^{(1)} = f'$ and in general $f^{(r)} = \frac{d^r}{dt^r} f$ denote higher order derivatives of f . These notations will be used interchangeably throughout the section.

Definition 2.11. Let φ, ψ be as in the definition of \mathcal{T} . The pair (φ, ψ) is *admissible* if they satisfy the following additional conditions:

- the function φ is smooth on $[a, b]$ and $\varphi'(a) \neq 0$;
- the function ψ is smooth on $[a, b]$ and $\psi'(b) \neq 0$;
- the function φ is analytic around b and $\varphi^{(j)}(b) = 0$ for $j = 1, \dots, n_2 - 1$;
- the function ψ is analytic around a and $\psi^{(j)}(a) = 0$ for $j = 1, \dots, n_1 - 1$.

For example, the pair $(\sin t, \cos t)$ which occurs in Definition 2.7 is admissible on the interval $[0, \pi/2]$ with $n_1 = n_2 = 2$. Our construction of admissible pairs for arbitrary values of n_1, n_2 depends on the following observation.

Let m, n denote arbitrary positive integers. Since $\mathbb{R}[t]$ is a principal ideal domain in which the polynomials $(1-t)^m$ and t^n coprime, there exists a unique pair of polynomials $f = f_{m,n}, g = g_{m,n} \in \mathbb{R}[t]$ with $\deg f \leq n-1$ and $\deg g \leq m-1$ such that

$$(1-t)^m f_{m,n}(t) + t^n g_{m,n}(t) \equiv 1. \quad (2.22)$$

Since the resultant of the polynomials $(1-t)^m$ and t^n is obviously 1, it follows from [62, Theorem 1.1] that in fact $f_{m,n}, g_{m,n} \in \mathbb{Z}[t]$.

Lemma 2.12. *For arbitrary positive integers m, n we have*

$$f_{m,n}(t) = \sum_{j=0}^{n-1} \binom{m-1+j}{j} t^j$$

and $g_{m,n}(t) = f_{n,m}(1-t)$. In particular, $f_{m,n}(0) = g_{m,n}(1) = 1$ and

$$f_{m,n}(1) = \binom{m+n-1}{m}, \quad g_{m,n}(0) = \binom{m+n-1}{n}$$

are different from zero.

Proof. Write

$$f_{m,n}^*(t) = \sum_{j=0}^{n-1} \binom{m-1+j}{j} t^j.$$

Based on the recursive formula for the binomial coefficients it is easy to show that

$$(1-t)f_{m,n}^*(t) = f_{m-1,n}^*(t) - \binom{m+n-2}{n-1} t^n.$$

By induction on i we obtain for $i = 1, \dots, m-1$ the identity

$$(1-t)^i f_{m,n}^*(t) = f_{m-i,n}^*(t) - t^n g_{m,n,i}(t)$$

with some auxiliary polynomial $g_{m,n,i} \in \mathbb{Z}[t]$ whose degree is $i-1$, and finally also

$$(1-t)^m f_{m,n}^*(t) = (1-t)(f_{1,n}^*(t) - t^n g_{m,n,m-1}(t)) = 1 - t^n g_{m,n}^*(t)$$

with a polynomial $g_{m,n}^* \in \mathbb{Z}[t]$ of degree $m-1$. By the uniqueness of the polynomials satisfying (2.22) under the given degree conditions we conclude that $f_{m,n} = f_{m,n}^*$ and $g_{m,n} = g_{m,n}^*$ for every m, n . Substituting $1-t$ for t in (2.22) gives

$$t^m f_{m,n}(1-t) + (1-t)^n g_{m,n}(1-t) \equiv 1$$

for every m, n . Therefore $f_{m,n}(1-t) = g_{n,m}(t)$ and $g_{m,n}(1-t) = f_{n,m}(t)$. \square \square

Proposition 2.13. *Let m, n be arbitrary positive integers. Then the pair of functions*

$$\varphi(t) = t \sqrt[n]{g_{m,n}(t)} = t \sqrt[n]{f_{n,m}(1-t)}, \quad \psi(t) = (1-t) \sqrt[m]{f_{m,n}(t)}$$

is admissible on the interval $[a, b] = [0, 1]$ with $n_1 = n, n_2 = m$.

Proof. The functions φ, ψ obviously satisfy (2.14) and (2.15). To see that they are increasing resp. decreasing bijective functions it is enough to check that

$$F(t) = (1-t)^m f_{m,n}(t)$$

is a strictly decreasing continuous function on the interval $[0, 1]$, which follows from

$$\begin{aligned} F'(t) &= (1-t)^{m-1} ((1-t)f'_{m,n}(t) - mf_{m,n}(t)) \\ &= (1-t)^{m-1} \left((1-t) \sum_{j=1}^{n-1} j \binom{m-1+j}{j} t^{j-1} - m \sum_{j=0}^{n-1} \binom{m-1+j}{j} t^j \right) \\ &= -(1-t)^{m-1} t^{n-1} (m+n-1) \binom{m+n-2}{n-1} \\ &< 0 \end{aligned}$$

for $0 < t < 1$.

Since φ and ψ are positive on the interval $(0, 1)$, so are the functions $f_{m,n}, g_{m,n}$. In view of the boundary values given in Lemma 2.12 and by the continuity of the functions there exist $\varepsilon, \delta > 0$ such that

$$f_{m,n}(t), g_{m,n}(t) > \delta \text{ for } t \in (-\varepsilon, 1 + \varepsilon).$$

It follows that the functions $\sqrt[n]{f_{m,n}(t)}$ and $\sqrt[m]{g_{m,n}(t)}$ are analytic on the interval $(-\varepsilon, 1 + \varepsilon)$, hence so are φ and ψ . Thus the pair φ, ψ satisfy all the smoothness requirements.

As for the boundary conditions,

$$\varphi'(0) = \sqrt[n]{g_{m,n}(0)} = \binom{m+n-1}{n}^{1/n} \neq 0.$$

In view of (2.22) the function φ can be represented as

$$\varphi(t) = \sqrt[n]{1 - (1-t)^m f_{m,n}(t)} = \sqrt[n]{1 - F(t)}$$

on the interval $(0, 1 + \varepsilon)$. Accordingly we have

$$\varphi'(t) = -\frac{1}{n} F'(t) (1 - F(t))^{\frac{1}{n}-1} = (1-t)^{m-1} G(t)$$

where

$$G(t) = -\frac{1}{n} ((1-t)f'_{m,n}(t) - mf_{m,n}(t)) (1 - F(t))^{\frac{1}{n}-1}$$

is a smooth function on $(0, 1 + \varepsilon)$. Thus it is immediate from the general Leibniz rule that $\varphi'(1) = \dots = \varphi^{(m-1)}(1) = 0$. Similarly,

$$\psi'(1) = \binom{m+n-1}{m}^{1/m} \neq 0$$

and $\psi'(0) = \dots = \psi^{(n-1)}(0) = 0$. □ □

Next we establish analogues of (2.17) and (2.18) for the derivatives of the functions T_i . Note that the functions $L_i(t, \alpha)$ can be defined on the interval $[a, b)$ for arbitrary nonnegative integers i .

Lemma 2.14. *Assume that the pair (φ, ψ) is admissible and let $r \in \{1, 2, \dots, n_1 - 1\}$, $i \geq r$. Then*

$$\frac{d^r}{dt^r} L_i(t, \alpha) \Big|_{t=a} = \begin{cases} 0 & \text{for } i > r, \\ (\varphi'(a))^r (\alpha - r + 1)_r \neq 0 & \text{for } i = r. \end{cases}$$

Proof. By the general Leibniz rule, the r th order derivative of the function L_i can be written in the form

$$\frac{d^r}{dt^r} L_i(t, \alpha) = \binom{\alpha}{i} \sum_{k=0}^r \binom{r}{k} \frac{d^{r-k}}{dt^{r-k}} (\varphi(t))^i \frac{d^k}{dt^k} (1 - \varphi(t))^{\alpha-i}.$$

Using the Leibniz rule again (see also [54, Lemma on p. 225]) we obtain the formula

$$\frac{d^{r-k}}{dt^{r-k}} (\varphi(t))^i = \sum_{j_1 + j_2 + \dots + j_i = r-k} \binom{r-k}{j_1, j_2, \dots, j_i} \varphi^{(j_1)}(t) \varphi^{(j_2)}(t) \dots \varphi^{(j_i)}(t).$$

Here the j_l are nonnegative integers. Since $\varphi(a) = 0$, we have

$$\varphi^{(j_1)}(a) \varphi^{(j_2)}(a) \dots \varphi^{(j_i)}(a) = 0$$

unless $j_l \geq 1$ for every l . Therefore

$$\frac{d^{r-k}}{dt^{r-k}} (\varphi(t))^i \Big|_{t=a} = 0$$

when $i > r - k$. Consequently,

$$\frac{d^r}{dt^r} L_i(t, \alpha) \Big|_{t=a} = 0$$

for $i > r$. Furthermore,

$$\begin{aligned} \frac{d^r}{dt^r} L_i(t, \alpha) \Big|_{t=a} &= \binom{\alpha}{r} \binom{r}{0} \frac{d^r}{dt^r} (\varphi(t))^r \Big|_{t=a} (1 - \varphi(a))^{\alpha-r} \\ &= \binom{\alpha}{r} \binom{r}{1, 1, \dots, 1} (\varphi'(a))^r \\ &= (\varphi'(a))^r (\alpha - r + 1)_r \end{aligned}$$

is nonzero due to $\alpha \geq n_1 > r$ and the boundary condition $\varphi'(a) \neq 0$. □ □

Lemma 2.15. *Assume that the pair (φ, ψ) is admissible and let $r \in \{1, 2, \dots, n_1 - 1\}$. Then*

$$\left. \frac{d^r}{dt^r} R_{n_1+n_2-i}(t, \beta) \right|_{t=a} = 0$$

holds for $i \in \{0, 1, \dots, n_2 - 1\}$. Moreover,

$$\left. \frac{d^r}{dt^r} T_{n_1}(t, \alpha, \beta) \right|_{t=a} = 0.$$

Proof. According to the general Leibniz rule,

$$\frac{d^r}{dt^r} R_{n_1+n_2-i}(t, \beta) = \binom{\beta}{i} \sum_{m=0}^r \binom{r}{m} \frac{d^{r-m}}{dt^{r-m}} (\psi(t))^i \frac{d^m}{dt^m} (1 - \psi(t))^{\beta-i}.$$

For the calculation of the derivatives $\frac{d^m}{dt^m} (1 - \psi(t))^{\beta-i}$ with $m \leq r \leq n_1 - 1$ we apply Faà di Bruno's formula (see e.g. [54]), which yields

$$\frac{d^m}{dt^m} (1 - \psi(t))^{\beta-i} = \sum \binom{m}{b_1, b_2, \dots, b_m} \binom{\beta-i}{k} (-1)^k (1 - \psi(t))^{\beta-i-k} \Psi_b(t).$$

Here $k = b_1 + b_2 + \dots + b_m$,

$$\Psi_b(t) = \left(\frac{\psi^{(1)}(t)}{1!} \right)^{b_1} \left(\frac{\psi^{(2)}(t)}{2!} \right)^{b_2} \dots \left(\frac{\psi^{(m)}(t)}{m!} \right)^{b_m}$$

and the summation is taken over all nonnegative solutions $b = (b_1, b_2, \dots, b_m)$ of the Diophantine equation $b_1 + 2b_2 + \dots + mb_m = m$.

Since ψ is analytic around a and $\psi^{(j)}(a) = 0$ for $j = 1, \dots, n_1 - 1$, (an extension of) ψ has a convergent Taylor series expansion on an interval $(a - \varepsilon, a + \varepsilon)$ in the form

$$\psi(t) = 1 + c_u(t - a)^u + \dots,$$

where $u \geq n_1$ and $c_u \neq 0$. Then $(1 - \psi(t))^{n_2-i-k} \Psi_b(t)$ can be expanded into Taylor series on the interval $[a, a + \varepsilon)$ as

$$(1 - \psi(t))^{n_2-i-k} \Psi_b(t) = c_{u^*}^* (t - a)^{u^*} + \dots,$$

where

$$c_{u^*}^* = (-1)^{n_2-i-k} c_u^{n_2-i} \prod_{j=1}^m ((u - j + 1)_j)^{b_j} \neq 0$$

and

$$u^* = u(n_2 - i - k) + \sum_{j=1}^m b_j(u - j) = u(n_2 - i) - \sum_{j=1}^m j b_j \geq u - m \geq 1.$$

Therefore

$$\lim_{t \rightarrow a^+} (1 - \psi(t))^{\beta-i-k} \Psi_b(t) = (1 - \psi(a))^{\beta-n_2} \cdot \lim_{t \rightarrow a^+} (1 - \psi(t))^{n_2-i-k} \Psi_b(t) = 0.$$

This implies that

$$\frac{d^m}{dt^m} (1 - \psi(t))^{\beta-i} \Big|_{t=a} = 0$$

for $m \leq r$, and finally

$$\frac{d^r}{dt^r} R_{n_1+n_2-i}(t, \beta) \Big|_{t=a} = 0$$

for $i = 0, 1, \dots, n_2 - 1$.

To prove the other equality, first we take the derivative of (2.16), valid on the interval $[a, c)$, where $\varphi(c) = 1/2$. This can be rephrased as

$$\frac{d^r}{dt^r} \sum_{i=0}^{\infty} L_i(t, \alpha) = 0.$$

Since $r \leq n_1 - 1$, it follows from Lemma 2.14 that

$$\sum_{i=0}^{n_1-1} \frac{d^r}{dt^r} L_i(t, \alpha) = 0$$

is valid on the interval $[a, c)$. Therefore

$$\frac{d^r}{dt^r} T_{n_1}(t, \alpha, \beta) \Big|_{t=a} = - \sum_{i=0}^{n_1-1} \frac{d^r}{dt^r} L_i(t, \alpha) \Big|_{t=a} - \sum_{i=0}^{n_2-1} \frac{d^r}{dt^r} R_{n_1+n_2-i}(t, \beta) \Big|_{t=a} = 0,$$

which completes the proof. □ □

Due to the symmetry of the construction, in particular the symmetry in Eq. (2.14), switching the role of the parameters according to $a \leftrightarrow b$, $n_1 \leftrightarrow n_2$, $\varphi \leftrightarrow \psi$, $\alpha \leftrightarrow \beta$ we arrive at the following counterpart of Lemmas 2.14 and 2.15.

Lemma 2.16. *Assume that the pair (φ, ψ) is admissible and let $r \in \{1, 2, \dots, n_2 - 1\}$, $i \geq r$. Then*

$$\frac{d^r}{dt^r} R_{n_1+n_2-i}(t, \beta) \Big|_{t=b} = \begin{cases} 0 & \text{for } i > r, \\ (\psi'(b))^r (\beta - r + 1)_r \neq 0 & \text{for } i = r. \end{cases}$$

In addition

$$\frac{d^r}{dt^r} L_i(t, \alpha) \Big|_{t=b} = 0$$

holds for $i \in \{0, 1, \dots, n_1 - 1\}$, and

$$\frac{d^r}{dt^r} T_{n_1}(t, \alpha, \beta) \Big|_{t=b} = 0.$$

Proposition 2.17. *If $(\alpha, \beta) \neq (n_1, n_2)$, then the function system \mathcal{T} is linearly independent.*

Proof. Throughout the proof we may assume that the pair (φ, ψ) is admissible. Indeed, consider an arbitrary system

$$\mathcal{T}^* = \mathcal{T}(a^*, b^*; n_1, n_2; \varphi^*, \psi^*; \alpha, \beta)$$

based on an admissible pair (φ^*, ψ^*) . The existence of such a system is guaranteed by Proposition 2.13. The monotone increasing bijective function

$$h = (\varphi^*)^{-1}\varphi : [a, b] \rightarrow [a^*, b^*]$$

satisfies $\varphi(t) = \varphi^*(h(t))$ for every $t \in [a, b]$. In view of Eq. (2.14) $\psi(t) = \psi^*(h(t))$ also holds for every $t \in [a, b]$. By the construction of the systems \mathcal{T} and \mathcal{T}^* it is clear that \mathcal{T} is linearly independent if and only if \mathcal{T}^* is so.

Consider a relation

$$\sum_{i=0}^{n_1+n_2} \lambda_i T_i(t, \alpha, \beta) = 0$$

and its derivatives

$$\sum_{i=0}^{n_1+n_2} \lambda_i \frac{d^r}{dt^r} T_i(t, \alpha, \beta) = 0.$$

In order to prove that all the coefficients λ_i are zero, we first evaluate the derivatives at $t = a$. If $r = 0$, then in view of (2.17) we obtain

$$0 = \sum_{i=0}^{n_1+n_2} \lambda_i T_i(a, \alpha, \beta) = \lambda_0 T_0(a, \alpha, \beta)$$

with $T_0(a, \alpha, \beta) \neq 0$, yielding

$$\lambda_0 = 0.$$

When $r = 1$, we can use Lemmas 2.14 and 2.15 to infer that

$$\sum_{i=0}^{n_1+n_2} \lambda_i \frac{d}{dt} T_i(t, \alpha, \beta) \Big|_{t=a} = \lambda_0 \frac{d}{dt} T_0(t, \alpha, \beta) \Big|_{t=a} + \lambda_1 \frac{d}{dt} T_1(t, \alpha, \beta) \Big|_{t=a} = 0.$$

Since $\lambda_0 = 0$, Lemma 2.14 applied for $i = r = 1$ implies

$$\lambda_1 = 0.$$

This process can be continued for $r = 2, \dots, n_1 - 1$. In the r th step, taking into account that we have already established $\lambda_j = 0$ for $j = 0, 1, \dots, r - 1$, we obtain

$$0 = \sum_{i=0}^{n_1+n_2} \lambda_i \frac{d^r}{dt^r} T_i(t, \alpha, \beta) \Big|_{t=a} = \sum_{i=0}^r \lambda_i \frac{d^r}{dt^r} T_i(t, \alpha, \beta) \Big|_{t=a} = \lambda_r \frac{d^r}{dt^r} T_r(t, \alpha, \beta) \Big|_{t=a}$$

which results in $\lambda_r = 0$ according to Lemma 2.14. This way

$$\lambda_0 = \lambda_1 = \dots = \lambda_{n_1-1} = 0$$

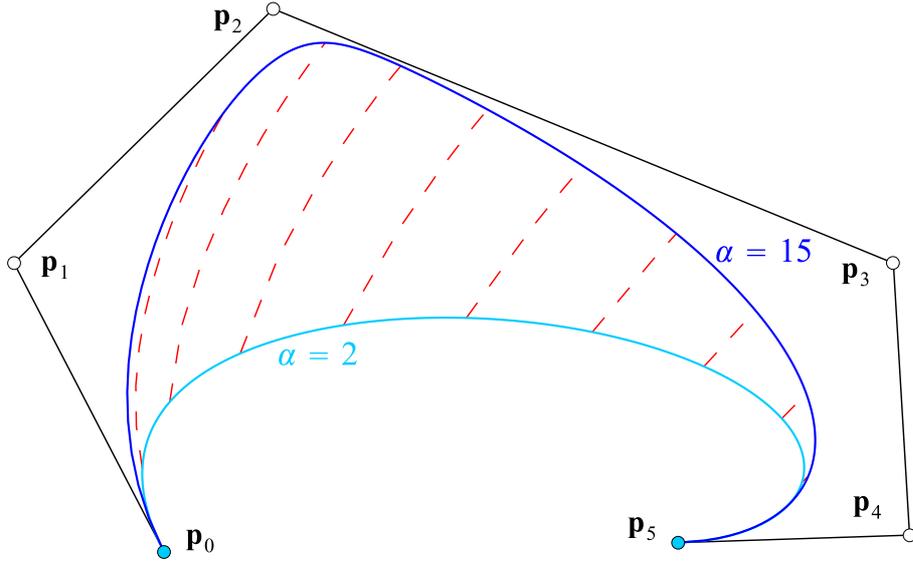


Figure 2.13. α -paths (red dashed lines) of a curve with settings $n_1 = 2, n_2 = 3, \beta = 3, \alpha \in [2, 15]$

is established.

Evaluating derivatives at $t = b$, we obtain $\lambda_j = 0$ for $j = n_1 + n_2, \dots, n_1 + 1$ as well by the step by step process shown above. Finally, λ_{n_1} also has to be zero, since the rest of the coefficients are zero and $T_{n_1}(t, \alpha, \beta)$ is not the zero function according to Proposition 2.10. \square \square

We conclude this section with the following summary of the advantageous properties of \mathcal{T} .

Proposition 2.18. *With fixed parameters $\alpha \in [n_1, \infty), \beta \in [n_2, \infty)$, the function system \mathcal{T} has the following properties:*

- nonnegativity,
- partition of unity,
- linear independence if $(\alpha, \beta) \neq (n_1, n_2)$.

2.4.4 The curve

Definition 2.19. Given control points $\mathbf{p}_i \in \mathbb{R}^d$ ($d \geq 2$), by means of the basis functions T_i the curve is defined in the form

$$\mathbf{c}(t, \alpha, \beta) = \sum_{i=0}^{n_1+n_2} T_i(t, \alpha, \beta) \mathbf{p}_i \quad \text{for } t \in [a, b] \quad (2.23)$$

where $\alpha \in [n_1, \infty)$ and $\beta \in [n_2, \infty)$ are global shape parameters.

The properties of the blending functions $T_i(t, \alpha, \beta)$ described in Proposition 2.18 involve the following characteristics of the curve (2.23), similar to those of the original Bézier curve and analogous to the properties of the curve defined in [31]:

- Affine invariance: since the function system \mathcal{T} provides a partition of unity, the curve is closed for affine transformations of its control points.
- Convex hull property: due to nonnegativity and partition of unity, the curve lies within the convex hull of its control points.
- Endpoint interpolation: due to Eqs. (2.17) and (2.18), the curve passes through the first and last control points.
- Global control: if $\alpha = n_1$ and $\beta = n_2$, then the control point \mathbf{p}_{n_1} has no influence on the shape of the curve, otherwise the curve is globally controlled, that is, each control point affects the shape of the whole curve.
- Endpoint tangency: tangent lines at the endpoints are parallel to the first and last sides of the of the control polygon, since the derivative of the curve at its first and last point is $\alpha\varphi'(a)(\mathbf{p}_1 - \mathbf{p}_0)$ and $\beta\psi'(b)(\mathbf{p}_{n_1+n_2} - \mathbf{p}_{n_1+n_2-1})$, respectively.

Remark 2.20. Recall that for the linear independence of \mathcal{T} one has to impose the constraint $(\alpha, \beta) \neq (n_1, n_2)$. Linear independence is required for the generation of interpolating curves of type (2.23), i.e. for the solution to the following problem. Given a sequence of data points $\{\mathbf{q}_i\}_{i=0}^{n_1+n_2}$ along with associated fixed parameter values $t_i < t_{i+1}$ in the range $[a, b]$ and shape parameters α, β , find suitable control points $\{\mathbf{p}_i\}_{i=0}^{n_1+n_2}$ for the curve (2.23) so that

$$\mathbf{c}(t_i, \alpha, \beta) = \mathbf{q}_i \quad (i = 0, 1, \dots, n_1 + n_2).$$

In order to describe the impact of the shape parameters of the curve, the path of a fixed curve point is considered. Let t_0 and β_0 be fixed values and consider the path $\mathbf{a}(\alpha) = \mathbf{c}(t_0, \alpha, \beta_0)$ along which the point of the curve associated with t_0 moves. These paths are called α -paths of the curve points. It is obvious from the definition of the basis functions, that each α -path is part of an exponential curve (cf. Fig. 2.13). Similarly, β -paths $\mathbf{b}(\beta) = \mathbf{c}(t_0, \alpha_0, \beta)$ can be computed by fixing the values t_0 and α_0 . When increasing any of the shape parameters, the curve is pulled towards the control point \mathbf{p}_{n_1} . This phenomenon can be observed in Fig. 2.13 for the shape parameter α .

Changing the function pair φ, ψ means only the reparametrization of the curve (2.23), that is, for any permissible function φ we obtain the same shape. The simplest choice would be $\varphi(t) = t$, $t \in [0, 1]$, however this parametrization is quite poor concerning the distribution of the points on the curve corresponding to uniformly specified parameter values in the domain. The function $\varphi(t) = \sin(t)$, $t \in [0, \pi/2]$ or its rational counterpart $\varphi(t) = 2t/(1+t^2)$, $t \in [0, 1]$ is a much better choice from this point of view. In Fig. 2.14 it can be observed that in the latter case the points lie closer to each other where the curvature is higher. In our experience the function $\varphi(t) = -2t^3 + 3t^2$, $t \in [0, 1]$ also provides a reasonable parameterization.

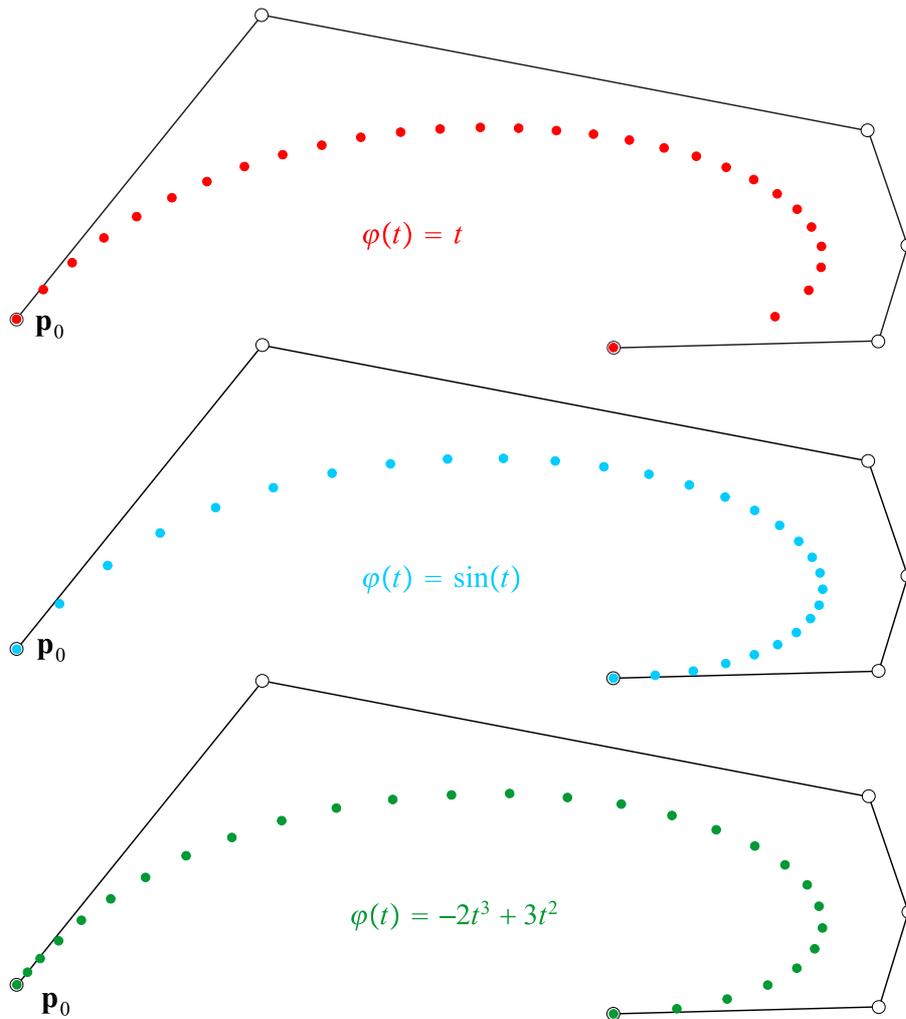


Figure 2.14. Comparison of different parametrizations, settings are $n_1 = 3, n_2 = 2, \alpha = 3, \beta = 2.5$ and domains are divided into 30 equal parts

3 Non-control-point-based methods

In the previous sections all curves and surfaces are defined by control points, which method has been applied since the beginning of computer aided modeling of geometric objects. There are, however, problems where this classical approach cannot be applied, or a set of preprocessing steps is required in order to use the standard methods. In this section two of these problems will be discussed, based on the publications [2, 32, 33, 36, 38, 69, 110].

3.1 Modeling unorganized points by artificial neural networks

Surface reconstruction from a set of unorganized spatial points is one of the central problems in computer aided design. In many applications, such as ship and car design, creating a surface from scattered data is a frequently applied technique. One can find numerous methods for approximation or interpolation of scattered data or updating existing surfaces by scattered points using space warping, NURBS, subdivision or algebraic surfaces (see e.g. [109] and references therein for a general overview of the problem).

Throughout this section we will apply B-spline surface as final surface. For the sake of simplicity generally bicubic surfaces are used. As we have previously seen, a B-spline surface is uniquely given by its degree, knot values and control points, which latter ones form a topologically quadrilateral mesh. In surface reconstruction problems the input is a set of unorganized points, thus the order, the knots and the control points are all unknowns. The overall aim of reconstruction methods is to determine these values where the basic strategy is the following (see [113] for overview):

1. Fix the order (k, l) , the number of control points (n, m) and the knot values u_i, v_j .
2. Assign a pair of parameters (u_r, v_r) to each scattered point \mathbf{P}_r .
3. Solve the system $\mathbf{s}(u_r, v_r) = \mathbf{P}_r$ or minimize $\sum_r \|\mathbf{s}(u_r, v_r) - \mathbf{P}_r\|^2$.

In terms of B-spline surfaces the crucial point of this strategy is step 2, which is frequently referred as the parametrization of the given data. Parametrization is the way how to assign parameter values to each point, where normally several restrictions and assumptions are introduced. One can try to consider the assigned values as unknown parameters in an optimization problem, but for large amount of data this approach leads to a complex non-linear system with several unknowns (see also [109]). At the recently developed base surface method data points are projected onto a predefined parametric surface to find the corresponding parameter value (c.f. [77], [91]). This technique can work well for certain type of data, but there are several conditions in terms of creating the base surface and the projection has to be a function, i.e. no overlapping allowed. Sometimes it is quite difficult to find a base surface which satisfies all the conditions.

This section is devoted to the neural network approach of scattered data fitting. The earliest approaches of surface reconstruction by Kohonen self-organizing neural network can be found in the author's previous works ([32], [33], [110]) and Yu's paper [116]. Later on similar methods in different contexts have been developed in the recent papers of Barhak *et. al.* ([3], [4], [66]),

Echevarría *et. al.* ([20]), Ivriissimtzis *et. al.* ([51], [52], [53]), and Knopf and Sangole ([63]). In this method we create different types of B-spline surfaces by Kohonen neural network in an iterative way. The quality of approximation can be controlled by the number of iteration steps and by other numerical parameters. The obtained surface can be used as a coarse approximant of the scattered data set or as a base surface for further reconstruction process. This way one can create base surfaces for more general types of data sets than by earlier methods. In this section the method is also applied for creating 3D ruled surfaces which are of great importance in computer aided manufactory.

In the next section we give a brief introduction to Kohonen neural network and the network will be applied to create base surface for scattered points.

3.1.1 The Kohonen neural network and its application

Artificial neural networks are widely used in computer science and its applications thus there are several types of networks for specific problems. A neural network consists of numerous computational elements (neurons or nodes), highly interconnected to each other. A weight is associated to every connection. Normally nodes are arranged into layers. During a training procedure input vectors are presented to the input layer with or without specifying the desired output. According to this difference neural networks can be classified as supervised or unsupervised (self-organizing) neural nets. Networks can also be classified according to the input values (binary or continuous). The learning procedure itself contains three main steps, the presentation of the input sample, the calculation of the output and the modification of the weights by specified training rules. These steps are repeated several times, until the network is said to be trained. For details and survey of artificial neural networks see e.g. [27],[96].

The Kohonen network, also known as self-organizing map, is a two-layer unsupervised continuous valued neural network [64]. The network has a strong self-organizing ability, which practically used for dragging a predefined structure - a polygon for curve modelling and a quadrilateral grid for surface modelling - towards the given points. After the so-called training procedure this predefined grid will follow the structure and distribution of the given points.

The outline of the training steps of a Kohonen network containing n input and m output node are the following (see Fig.3.1.):

1. Present new input values $x_i, i = 1, \dots, n$.
2. Input nodes send them to each output node.
3. Output nodes compute the output values d_j by

$$d_j = \sum_{i=1}^n (x_i - w_{ij})^2 \quad j = 1, \dots, m$$

where w_{ij} is the weight associated to the connection from the i^{th} input node to the j^{th} output node.

4. The node with the minimum output $d_{\min} = \min \{d_j\}$ is the winning node.

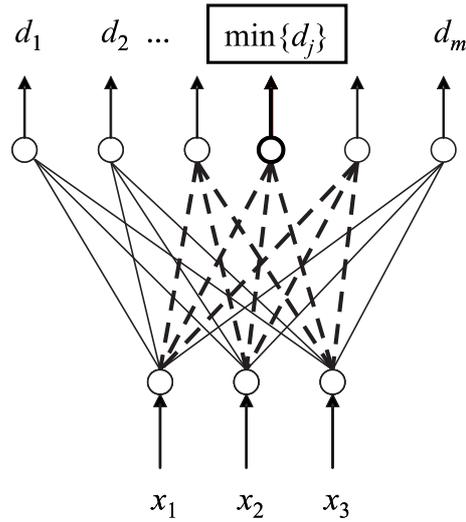


Figure 3.1. The training procedure of the Kohonen neural network

5. The weights of the connections to the winning node as well as their neighbors are updated in a well-defined way.
6. Go to step 1 until the network is trained.

How can we apply this tool for surface fitting? Let the number of input nodes $n = 3$, thus each output node has three connections. The three weights associated to these connections can be considered as spatial coordinates of a point $\mathbf{Q}_j(w_{1j}, w_{2j}, w_{3j})$ in 3D (see Fig.3.2).

One can fix the neighborhood relations of the network in advance, which yields a topologically fixed grid in 3D. If the weights of the network are changed during the training steps, some points of the spatial grid will change their spatial positions but the topology remains the same as before.

Now if the input values are spatial coordinates of one of the scattered points, then the training procedure will move this grid slightly towards the input point. This is because the winning node is associated to that point of the grid which is closest to the input point. The next iteration comes with another scattered point and the grid will move a little towards that point etc., thus after several iteration the grid will spread out and follow the overall shape of the scattered points. The movement can be controlled by numerical parameters as we will discuss it in the next section, based on [38].

The precise algorithm of the training procedure is as follows [32], [38]:

Input: scattered points \mathbf{P}_r (the number of the points are irrelevant)

Output: a grid with predefined topology which follows the overall shape of the scattered point set

1. Fix the topology of the grid and the number of output nodes m . Let the number of input nodes $n = 3$. Let $t = 1$.
2. Initialize the weights w_{ij} ($i = 1, 2, 3; j = 1, \dots, m$) of the network as small random numbers

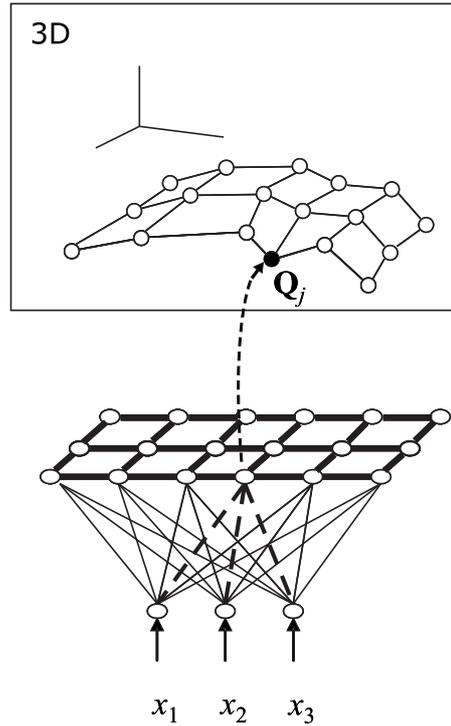


Figure 3.2. The application of the Kohonen network for surface fitting

around the centroid of the point set or according to additional data (e.g. border of the point cloud).

3. Present an input – three coordinates of a randomly selected spatial point $\mathbf{P}_i(x_1, x_2, x_3)$.
4. Compute the output values and find the winning node by

$$d_{\min} = \min \left\{ d_j = \sum_{i=1}^3 (x_i - w_{ij})^2, \quad j = 1, \dots, m \right\}$$

i.e. the node which is associated to the closest point \mathbf{Q}_{\min} of the grid to the output point in 3D.

5. Find the neighbors of the winning node by the neighborhood function $N(t)$ and update the weights of these nodes by

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)(x_i - w_{ij}(t)) \quad (3.1)$$

where $\eta(t)$ is a real-valued function called gain term.

6. Let $t = t + 1$ and decrease $\eta(t)$ and $N(t)$.
7. Go to step (3) and start next iteration until the network is trained. The network is said to be trained if the movement of the grid (i.e. the value of the gain term) falls under a predefined limit (normally $\eta(t) = 0.001$).

Some steps of the algorithm may need some further explanation. The topology of the grid in Step 1 means the structure of connections and the overall topology of the grid as well. Here the points of the grid are connected in quadrilateral way since the B-spline surface originally defined on this kind of grid. Other types of connectivity, however, can also be defined: triangular topology may be better suited for other applications (c.f. [116], [52]). The overall topology of the grid is a much harder problem, since normally we have no information of the topological properties of the input data. It has to be defined in advance and here we assume that the desired surface is of genus 0. The change of the overall topology of the grid during the training session would be the solution of the problem. Recent results in [52] and [66] show the problems of this direction of research which is, generally speaking, one of the main problems of surface reconstruction techniques today.

The number of output nodes m has to be fixed in Step1 of the algorithm. This works well in several cases but sometimes the refinement of the grid would yield a better surface. This problem can be solved by the dynamic version of the Kohonen neural network also known as growing cell structure [28]. The basic idea of the refinement is that further nodes are incorporated to the grid around the most frequent winners. In this case each node has a counter which is increased by 1 if this node is the winner. If one of the counters will be equal to a predefined limit a row or a column will be inserted next to that neuron. For the detailed description of this modified algorithm see [110]. If we want to preserve the quadrilateral topology of connectivity we have to insert rows or columns anyway. In the case of triangular connectivity vertex split and edge collapse can be executed to refine the grid (c.f. [52]). This dynamic version of neural network certainly has some advantages: the number of training iterations decreases dramatically, the algorithm is generally faster even with the additional counters. The main drawback of this version is that the convergence of the dynamic version has not been established theoretically as yet.

Step 2 of the algorithm includes the initialization of the weights. This process determines the initial position of the grid. If we have no additional information about the point cloud, this initial situation can be a shranked grid somewhere "close" to the cloud. That's why we applied the following technique: some points are randomly chosen from the set and the centroid of them is computed. The grid is positioned initially around this center adding small random values to the coordinates of the center to obtain the vertices meanwhile the quadrilateral connections are fixed. This yields the shranked grid which will spread out during the iterations. Actually this initial position is not extremely important in terms of the final surface - the first few iterations drag the grid towards the center of the point cloud anyway. Most of the techniques not even mention the initial position ([51]) or using a very similar technique ([63]). The only important thing is that if the desired surface has a sphere-like shape and topology, like in our example (Fig. 3.3), the initial position should be somewhere inside the point cloud.

If we have additional information of the data, we can use it to determine an even better initial position. If for example we know the plane where the surface has a boundary, we can apply the "SOM boundary first" technique (c.f. [3]) to determine a planar grid with correct boundary which does not move during the training (see our example Fig. 3.3).

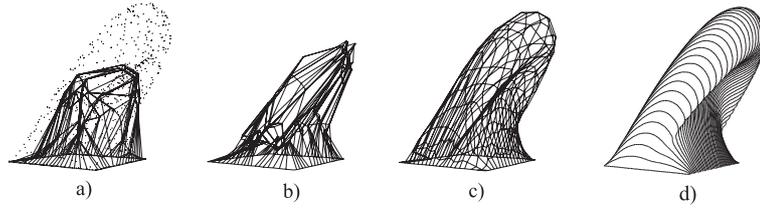


Figure 3.3. Movement of the grid during the training session: a) the scattered data and one of the first iterations b)-c) further iteration steps of the training d) the surface computed from the trained network

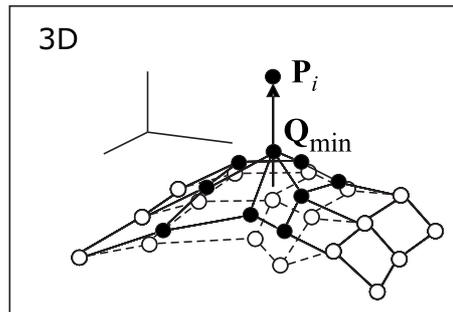


Figure 3.4. The gain term and the neighborhood function are for control of the movement of the grid: the gain term influences the measure of the movement of Q_{min} , the neighborhood function influences the moving part (filled circles)

Fig. 3.3. shows some iterations of the training procedure as the grid is spreading out towards the input points. Finally the output of the algorithm is a topologically predefined grid which can be used as a control mesh for a B-spline surface. This B-spline surface is a rough approximant of the point cloud but can also be used for further process as a base surface for surface reconstruction. The reconstructed surface can be seen in Fig.3.3.d).

3.1.2 Numerical control of the training: the gain term and the neighborhood function

The algorithm given above contains two "hidden" functions: the gain term and the neighborhood function. These functions are for numerical control of the speed and accuracy of the approximation, i.e. the quality of the final surface.

The gain term, as one can see from equation (3.1) at step (5) of the algorithm, is for control of the measure of movement of the grid. The less the value of the gain term the smaller the movement of the point Q_{min} and its neighbors towards the input point P_i (see Fig.3.4.). The value of the gain term has to be in $[0, 1)$. If it would be equal to 1, the point Q_{min} would reach P_i .

The movement of the grid is desired to be large at the first iterations of the training session, when the overall shape of the scattered data should be found. The final iterations however require

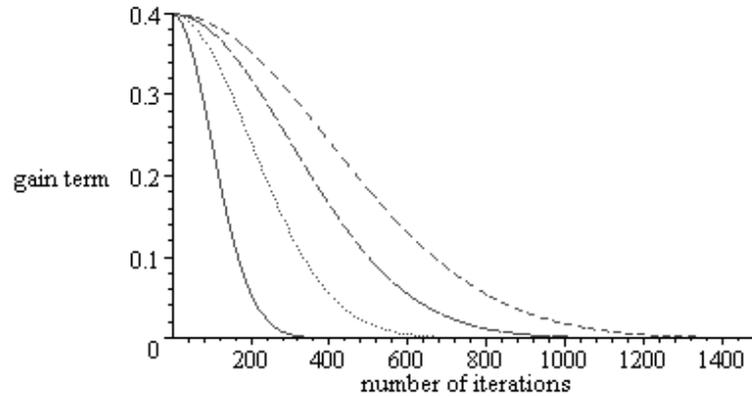


Figure 3.5. Graph of the gain term with parameter values $q=100$ (leftmost), 200, 300 and 400 (rightmost)

smaller movements when the final tuning of the grid is executed. Thus the gain term should be a Gaussian function. Notice that the gain term tends to 0 which guarantees the convergence of the algorithm as it has already been declared in [64]. Using the original suggestion of Kohonen here we applied the following function

$$\eta(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} \frac{t}{q}^2},$$

where t is the actual number of iterations and q is a parameter which has to be fixed in advance. It is very useful to incorporate the parameter q to the function by which one can control the steepness of the gain term : if the overall structure of the scattered data seems to be quite simple, then the decrease of the gain term can be very fast. If the shape is more complicated then slower change of the gain term is required to achieve good approximation (see Fig. 3.5). Naturally this latter case requires more iteration steps for the neural network to be trained.

The other function for the control of the training session is the neighborhood function $N(t)$. From step (5) of the algorithm it is clear that $N(t)$ is for the control of the moving part of the grid: the less the value of the neighborhood function the smaller the part of the grid which moves. During the training session movement of large parts at the first steps and small parts at the final tune are desired. Thus similarly to the gain term the neighborhood function is also a Gaussian:

$$N(t) = \text{INT} \left(\frac{m}{2} e^{-\frac{1}{2} \left(\frac{t}{s}\right)^2} \right),$$

where t is the actual number of iterations, m is the number of output neurons and s is a parameter which has to be fixed in advance. The role of s is similar to that of q . For simple shape the neighborhood function can be decreased faster than for more complicated structure to fasten the training session as well. If we have no information about the input data, q and s can be settled for 200.

Finally we have to mention that unfortunately these parameters cannot be settled for ever since different scattered data sets require different values. There is not much hope to determine

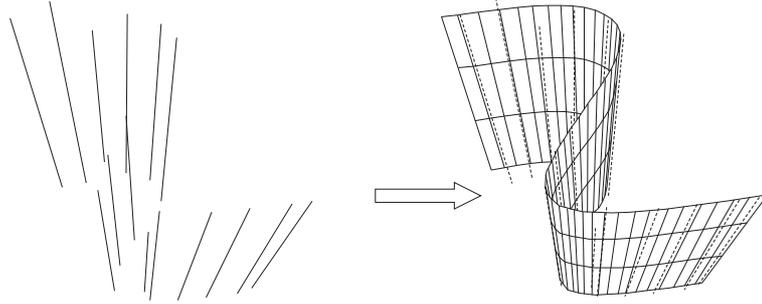


Figure 3.6. Construction of ruled surface from a set of unordered rulings

these values in an automatic way as well, since in that case the overall structure of the input point set should have been described automatically on a point based method *before* the modeling process. If we fix these parameters q and s as 200, the learning rate and accuracy will be a kind of average, i.e. probably better result can be achieved by modifying the parameters in advance. Other approaches, like [52], [53] also use some parameters to control the learning procedure and these also should be adjusted in a data dependent way.

3.1.3 Ruled surfaces from a set of rulings

Ruled surfaces are surfaces which contain a line through each point of the surface. They are of great importance in CAD/CAM. A further application of the described neural network technique is the construction of ruled surfaces from an unordered set of line segments, called rulings (c.f. Fig.3.6.).

To apply the method described above first we shift the problem from \mathbb{E}^3 to the five dimensional projective space \mathbb{P}^5 with the help of Plücker-coordinates (c.f.[47]). Thus a point in \mathbb{P}^5 will be associated to each line in \mathbb{E}^3 and the original algorithm will be executed in \mathbb{P}^5 . The output will be transformed back to \mathbb{E}^3 then.

Pick up two arbitrary points $\mathbf{A}(x_1, x_2, x_3)$ and $\mathbf{B}(y_1, y_2, y_3)$ of a line e and use the homogeneous coordinates $(x_1, x_2, x_3, x_4 = 1)$ and $(y_1, y_2, y_3, y_4 = 1)$, respectively. Compute the six Plücker-coordinates (l_1, l_2, \dots, l_6) of the line e by the following equations

$$\begin{aligned} l_{ij} &= x_i y_j - x_j y_i, & i, j &= 1, \dots, 4 \\ (l_1, l_2, \dots, l_6) &= (l_{41}, l_{42}, l_{43}, l_{23}, l_{31}, l_{12}). \end{aligned}$$

The Plücker-coordinates are independent to the selected points and are unique, thus a point $\mathbf{L}_e(l_1, l_2, l_3, l_4, l_5, l_6) \in \mathbb{P}^5$ is assigned to the line $e \in \mathbb{E}^3$.

Now the algorithm to create ruled surface from a set of rulings is as follows:

Input: a set of unordered rulings e_r in \mathbb{E}^3 .

Output: a topologically quadrilateral grid which contains only line segments in one direction

1. A point \mathbf{L}_r in \mathbb{P}^5 is assigned to each given ruling e_r by the Plücker-coordinates. Thus an unordered set of points is obtained.

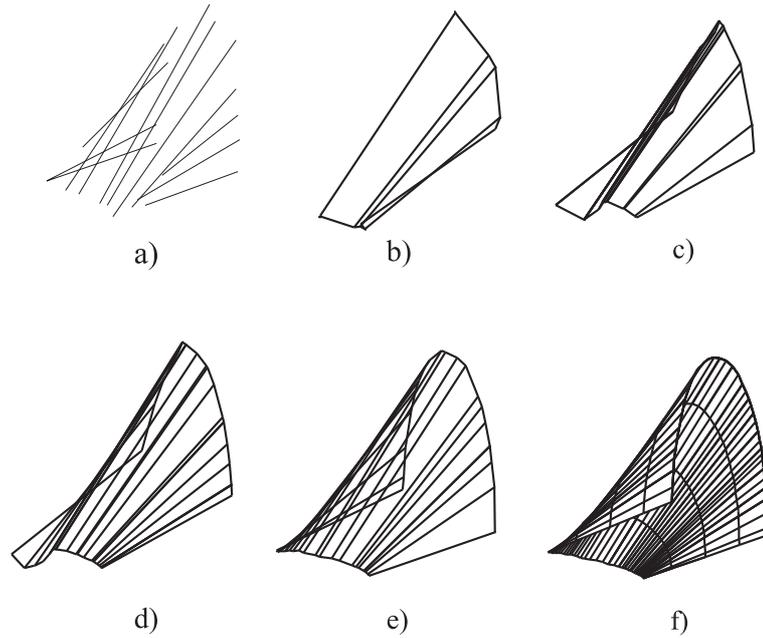


Figure 3.7. Ruled surface from a set of rulings: a) the input lines, b)-e) the training steps, f) the surface

2. A Kohonen network is defined with 6 input nodes and the topology of the output nodes is fixed in a linear way. Let the points assigned to the nodes are $\mathbf{Q}_j \in \mathbb{P}^5$. These points form a polygon which moves during the training session.
3. Applying the Kohonen network algorithm we obtain a polygon \mathbf{Q}_j in \mathbb{P}^5 which follows the overall shape of the point set \mathbf{L}_r .
4. Using the Plücker-coordinates backwards ordered lines $q_j \in \mathbb{E}^3$ are assigned to the vertices \mathbf{Q}_j of the polygon in \mathbb{P}^5 .
5. Connecting the lines q_j and cut them by two planes a topologically quadrilateral grid is obtained with line segments in one direction.

The resulted grid can be applied as a control mesh for a B-spline surface which is a good approximant of the given rulings. An example can be seen in Fig.3.7. (the training steps are executed in \mathbb{P}^5 but shown in \mathbb{E}^3 for better visibility).

3.2 Sphere-based modeling

Interpolation of geometric data sets is of central importance in Computer Aided Geometric Design. If geometric data consist of points, then we have several, now standard methods to interpolate them [24, 90, 49].

Creating various shapes is always the first step of many computer graphics related problems. Nowadays there is a growing demand to extend the well-known, point based surface methods (all the spline and subdivision surfaces are based on a predefined set of points) to new algorithms

which rely on other types of geometric primitives such as spheres. Animation and creation of characters are of central importance in computer graphics, where skinning methods are also applied [102].

If, however, the data set consists of other types of objects (e.g. circles), interpolation is transferred to *skinning*, that is construction of a curve or surface which touches each of the objects and somehow bounds the given data. Since this is a largely ill-posed problem, constraints have to be defined for the data set as well as for the desired solution.

As a new tool, surface modeling by spheres has already appeared in computer graphics in the last couple of years. Several papers have been published in this topic, where theory of sphere based surface modeling has been introduced, and applied in several fields, including medical and biological applications, character animation or covering problems, see e.g. [102], [103], [104], [97], [98]. Moreover, commercial software tools have also been developed in this field, such as ZSpheres[®] [124] by Pixologic, or Spore[™] [105] by Electronic Arts.

In this subsection we introduce a novel approach of sphere based shape modeling. Based on the theoretical approach of blending of spheres developed in [69], the method introduced in [2] provides a parametric surface, skinning of a predefined, ordered set of spheres. The surface can be interactively modified in real time by adjusting the positions and radii of the spheres. New branches can also be added by simply defining new sequences of spheres. The result is a piecewisely defined surface, where the branches are connected automatically in a smooth (G^1 continuous) way. In Figure 3.24 we can see a comparison between our result and the result of ZSpheres[®].

At first we address the problem of skinning of a sequence of circles. This problem is sometimes called 2D ball skinning, with a natural extension to 3D, where a skinning surface of a set of given spheres is computed.

Since the problem of skinning is not necessarily defined in a unique way in the literature, here we formally describe what type of input is admissible for us and what type of output we are searching for.

Definition 3.1. A sequence of circles $C = \{c_1, c_2, c_3, \dots, c_n\}$ ($n \in \mathbb{N}$) is called *admissible configuration* if the following conditions are fulfilled (d_i denotes the closed disk defined by circle c_i):

- $d_i \subset \bigcup_{j=1, j \neq i}^n d_j, \quad i \in \{1, 2, \dots, n\}$
- $d_i \cap d_j = \emptyset, \quad i, j \in \{1, 2, \dots, n\}, j \notin \{i-2, i-1, i, i+1, i+2\}$
- if $d_{i-1} \cap d_{i+1} \neq \emptyset$, then $d_{i-1} \cap d_{i+1} \subset d_i$

These assumptions also yield

$$\mathbf{r}_i \notin \bigcup_{j=i-1}^{i+1} d_j, \quad i = 2, \dots, n-1$$

where \mathbf{r}_i is the radical center of three consecutive circles c_{i-1}, c_i, c_{i+1} . The second restriction is for avoiding closed loops of four or more circles. In Fig. 3.8 one can see an admissible configuration and a configuration where some circles do not fulfill the conditions.

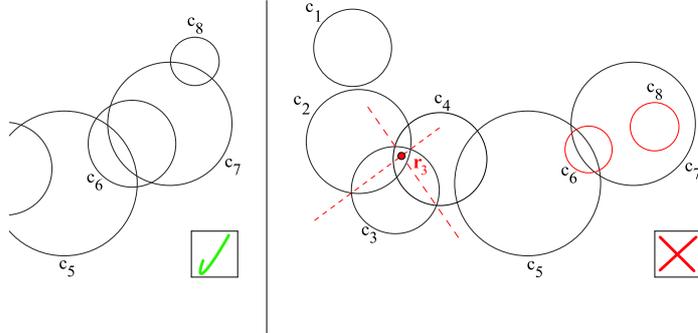


Figure 3.8. An admissible configuration (left) and a non-admissible configuration (right) where circles $c_2 - c_4$, c_6 and c_8 do not fulfill the conditions: circles are not allowed to be entirely in the union of other circles while radical center \mathbf{r}_3 of three consecutive circles must be out of c_3

Since the position of radical center will be of central importance in the algorithm, we briefly remind the reader to the definition and computation of it. The radical axis (or radical line) of two circles is the locus of points at which tangents drawn to both circles have the same length. Since it is evidently orthogonal to the line passing through the centers, it is enough to compute the distances $dist_1$ and $dist_2$ of the axis to the centers:

$$dist_1 = \frac{1}{2} \left(dist + \frac{r_1^2 - r_2^2}{dist} \right) \quad dist_2 = \frac{1}{2} \left(dist - \frac{r_1^2 - r_2^2}{dist} \right)$$

where $dist$ is the distance of the two centers, r_1, r_2 are the radii of the circles. By a classical theorem of Monge [19] the radical lines of three circles are either concurrent in a point, known as radical center, or parallel iff the three circle centers are congruent.

Now we define the desired output.

Definition 3.2. Given an admissible configuration of circles $C = \{c_1, c_2, c_3, \dots, c_n\}$, we are looking for two, at least G^1 continuous curves $\mathbf{s}(t)$ and $\bar{\mathbf{s}}(t)$, called *skins* of the given circles satisfying the following requirements (see also Fig. 3.9):

- There is a point of contact $\mathbf{p}_i \in c_i$ for all $i = 1, \dots, n$ such that $\mathbf{p}_i \in \mathbf{s}(t)$ and tangent lines of circle c_i and $\mathbf{s}(t)$ are identical at \mathbf{p}_i . Analogously exist points $\bar{\mathbf{p}}_i$ for $\bar{\mathbf{s}}(t)$.
- Tangent vector \mathbf{v}_i of skin $\mathbf{s}(t)$ at \mathbf{p}_i can be rotated to the direction of the center of c_i by 90° in clockwise direction. Analogously this rotation is in counterclockwise direction for tangent vectors of $\bar{\mathbf{s}}(t)$.

- $\mathbf{p}_i \notin \bigcup_{j=1, j \neq i}^n d_j$, and $\bar{\mathbf{p}}_i \notin \bigcup_{j=1, j \neq i}^n d_j$, $i \in \{1, 2, \dots, n\}$

For the sake of simplicity $s(t)$ will be called “left” skin and $\bar{s}(t)$ will be called “right” skin, based on the second requirement which ensures us that running along the skins, at the points of contact the circles will always be in one (right or left) side of the curves.

Note, that due to the last restriction, points of contact on the actual circle are required to be out of other circles, which seemed to us a natural condition for skinning.

This or similar problem - beside its theoretical interest - frequently arises in applications like designing tubular structures, covering problems, molecule modeling [16, 21]. Medical image processing applies these methods e.g. in blood vessel reconstruction [97, 104]. In computer animation, characters can also be constructed from a skeletal structure and a corresponding geometric skin [102].

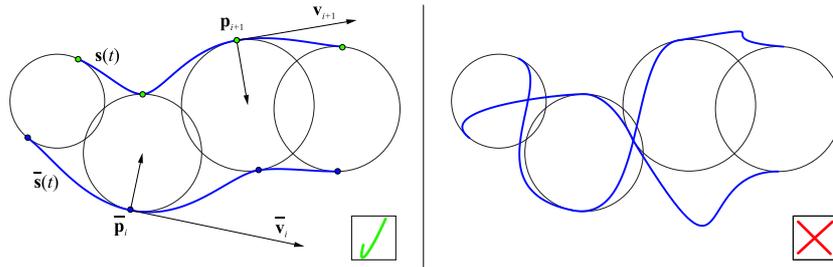


Figure 3.9. On the left, two curves satisfying all the requirements to be skins. At right there are two curves which also touch each circles but they do not fulfill the requirements: some of the touching points are inside of other circles and separation of the two sides is not appropriate

After briefly discussing the previous approaches, we describe the new method and show that for an admissible configuration of circles it always works. Detailed algorithm, several examples and comparison to Slabaugh’s method can also be found in this section. Spatial extension to sphere skinning by a surface is also provided.

3.2.1 Previous work

The first and most natural approach of the problem would be the application of the deep theoretical knowledge of the computation of envelope curves and surfaces, dated back to Monge [81], who first dealt with canal surfaces. Skin is definitely not an envelope, since this latter notion is defined for continuous data set, for a one- or two-parameter family of curves or surfaces. The first requirement at Definition 3.4, however may be considered as the discrete version of envelope property. An important contribution of this topic with computational aspects is the PhD thesis of Josef Hoschek [48]. Since then a large number of papers have dealt with envelope design, most of them with numerical computation (for the survey see e.g. [25]). For circles and spheres, exact computation of rational envelopes are presented in [87, 86], based on a cyclographic approach. In 2D cyclography defines a one-to-one correspondence between the oriented circles with center x, y and radius r of the plane and spatial points (x, y, r) (if the orientation is counterclockwise) or $(x, y, -r)$ (if the orientation is clockwise). This way the sequence of given circles can be

transformed to a sequence of spatial points. An interpolating curve through these points can be defined and finally points of this spatial curve can be transferred back to planar circles by the same transformation. The envelope of these circles is obtained as the intersection of the plane and the envelope surface of the cones defined by the spatial points as apices and the corresponding circles. Similar correspondence works for spheres and points in 4D space. For a more detailed description, see [67, 86] .

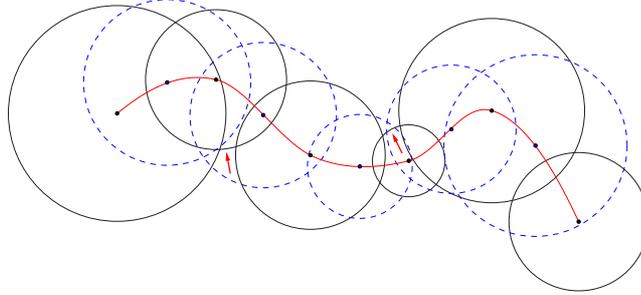


Figure 3.10. Given a set of discrete circles (black), classical interpolation may yield further circles (dashed blue) in a way that the skin cannot be constructed for the original circles since the new set of circles do not satisfy the requirements to form an admissible configuration (positions pointed by red arrows)

Although the papers mentioned above do not deal with skinning, one may try to transfer the discrete data set to a one-parameter family of circles/spheres, having centers and radii as functions of a parameter. These functions can be achieved from the set of discrete data by classical interpolating methods in the space, but this way the set of new circles do not necessarily satisfy our requirements to be an admissible configuration, as one can observe in Fig. 3.10.

A recent approach to the skinning problem for circles and spheres is Slabaugh's method [103, 104]. It is an iterative way to construct the desired curves or surfaces. Let the sequence of circles with centers \mathbf{o}_i and radii $r_i, (i = 1, \dots, n)$ be given. Considering two neighboring circles c_i and c_{i+1} , initial Hermite arcs are specified with touching points $\mathbf{p}_i, \mathbf{p}_{i+1}$ and tangents $\mathbf{v}_i, \mathbf{v}_{i+1}$ for the skin. The final positions of these points and tangents are obtained by the end of several iteration steps.

The iteration itself is based on the minimization of a predefined energy function. For computational reasons the positions of the touching points and the tangents are transferred into one single variable, namely the angle α_i between the x axis and the radius pointing towards the touching point.

$$\mathbf{p}_i = \mathbf{o}_i + \begin{bmatrix} r_i \cos \alpha_i \\ r_i \sin \alpha_i \end{bmatrix}$$

$$\mathbf{v}_i = \begin{bmatrix} -k_i \sin \alpha_i \\ k_i \cos \alpha_i \end{bmatrix},$$

where k_i is a predefined constant for each circle, half of the distance between centers \mathbf{o}_i and \mathbf{o}_{i+1} .

The method provides energy-minimized, C^1 continuous skin without any user interaction,

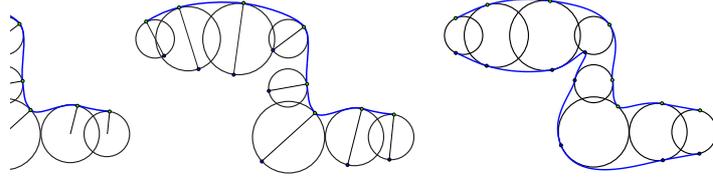


Figure 3.11. A one-sided (left) skin obtained by the method of Slabaugh (left figure, from [103]). For two-sided skin they constrain the points of contact to be separated by 180° , but this way some of the points of contact may fall into other circles (middle figure). Right figure shows the result of the proposed method: the green touching points of the left skin are almost identical to the ones obtained by Slabaugh. The right (lower) skin has significantly different touching points.

which, in this sense, the optimal solution, if it exists. But the method also suffers from problems. The touching points are not guaranteed to be out of the circles, especially not for two-sided skin, when the two touching points at each circle are constrained to be separated by 180° (c.f. Fig. 3.11). There are simple configurations when it is theoretically impossible to find two diametrically opposite points on a circle being out of other given circles, see e.g. the second circle in the leftmost figure of Fig. 3.16 or the lower right figure of Fig. 3.12. Slabaugh's method does not provide acceptable skin by our definition, especially the last requirement in Definition 3.2 is not necessarily fulfilled. From this point of view our method can handle a larger class of data sets. One would try to omit this 180° constraint from that method but then we are facing to solve the separation of touching points, which is far from being trivial in a numerical iteration. The same problem arises in 3D-ball skinning, where Slabaugh's method allows only great circles as possible touching circles. A further problem is that the convergence of iteration to a global minimum is not proved and the number of iterations can be over 100 which is time consuming. Moreover, the process has to be restarted after any modification of data, thus this method is not suitable for real time modeling and adjustment. For comparison to our method: the one-sided skin in Figure 3.11 (left) has been computed by Slabaugh in 143 milliseconds [103], the two-sided skin (right) has been computed by our method in 14 milliseconds (both at single core 3GHz CPU).

3.2.2 Skinning of circles

Now our task of skinning can be divided into the following steps:

- check if the given circles form an admissible configuration
- find appropriate points of contact for each circle $c_i, i = 1, \dots, n$
- separate points into two classes, denoted by \mathbf{p}_i and $\bar{\mathbf{p}}_i$ for left and right skin
- define tangent vectors \mathbf{v}_i and $\bar{\mathbf{v}}_i$
- compute the skins

The admissible criteriae of Definition 3.4 can be tested by elementary computation.

Localization of touching points As we have learned from the previous section, the localization of possible touching points on the given circles is essential for skinning. Our solution of finding the touching points is based on the circles of Apollonius, which are touching circles to three given circles. This ancient construction provides suitable touching points to each inner circle in the data set, while the first and last circles are handled in a simple special way.

A classical result [84] on the possible positions of three circles and solutions of problem of Apollonius states that for three given circles c_{i-1}, c_i, c_{i+1} , ($i \notin \{1, n\}$) which satisfy our admissible conditions, exactly one of the following statements holds:

- There exist exactly two circles touching externally by all the three given ones
- There exist exactly two circles touching internally by all the three given ones
- There exists exactly one circle touching externally and another one touching internally by all the three given ones.

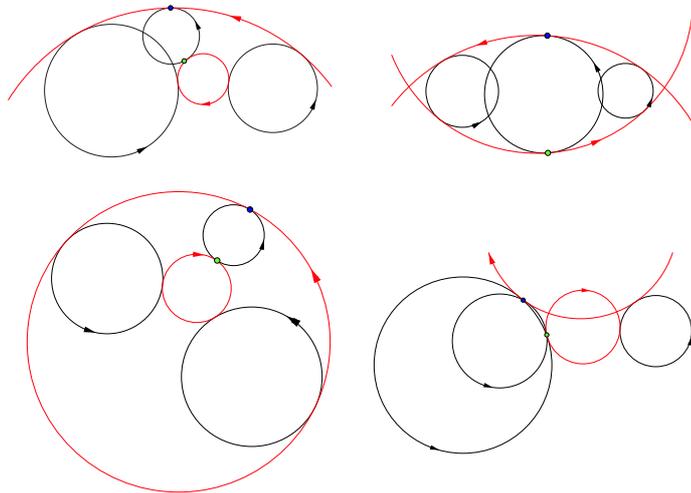


Figure 3.12. Considering three consecutive given circles (black), the touching points of the second circle are computed by the two circles of Apollonius (red). Orientation of circles help to find the correct solutions. The method always works for circles in admissible configuration

These touching circles together with their touching points can be found by a classical geometric method, the cyclography. Consider the three given circles to be oriented, having the same orientation (all clockwise or all counterclockwise). The touching circle is always expected to have the same orientation at each touching point as the given circles. The solutions of these two cases provide the touching points \mathbf{p}_i and $\bar{\mathbf{p}}_i$ of the circle c_i (c.f. Fig.3.12). These points can be constructed and computed by known methods [83, 17]. Moreover, it is also proved [83] that these touching points and the radical center of the three given circles are collinear (see Fig.3.13).

In the very special situation, when centers of the three circles are collinear and having equal radii, simply the common tangent lines give the points of contact.

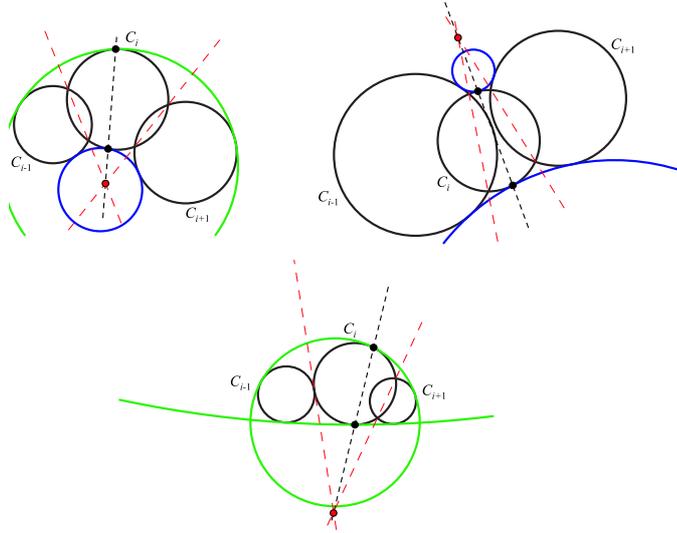


Figure 3.13. Given three circles c_{i-1} , c_i and c_{i+1} in admissible configuration, one can find the future points of contact for skinning at c_i by the two special solutions of the problem of Apollonius. The three possible situations: one circle touching externally and another one touching internally by all the three given ones (upper left); two circles touching externally (upper right); and two circles touching internally (middle). Radical center and touching points are collinear.

After defining the touching points at each inner circle, the first and last circles in the sequence have to be handled as well. Touching points to these two circles can also be defined by the common external tangent lines of the first two and the last two circles, respectively.

Separation to left and right groups Finally two points have been localized at the circles c_i which all satisfy the last requirement of Definition 3.2. The next step is to separate them for “left” and “right” classes, i.e. to identify which one should be denoted by \mathbf{p}_i and which one by $\bar{\mathbf{p}}_i$.

At first it is proved in [83] that these points of c_i can always be separated by the circle with radical center \mathbf{r}_i as center and intersecting orthogonally the three given circles c_{i-1} , c_i and c_{i+1} . Moreover it is mentioned that the radical center and the two touching points are collinear. Thus the separation can easily be computed by the following steps (see notations of Fig. 3.14): if the vector $\mathbf{o}_{i-1}\mathbf{o}_i$ can be rotated to the direction of vector $\mathbf{o}_{i-1}\mathbf{o}_{i+1}$ by a positive angle (in counterclockwise direction, with less than 180°) then the touching point being closer to the radical center \mathbf{r}_i will be in the left group, i.e. will be denoted by \mathbf{p}_i . If the direction of rotation is opposite (as it is for the next circle in Fig. 3.14) then the touching point being closer to the radical center \mathbf{r}_{i+1} is in the right group: $\bar{\mathbf{p}}_{i+1}$. Special attention must be paid to the first and last circle as well as for circles with collinear centers. In these cases the vector $\mathbf{o}_{i-1}\mathbf{o}_i$ is rotated to the direction of $\mathbf{o}_{i-1}\mathbf{p}_i$ and the angle is similarly measured and evaluated as above.

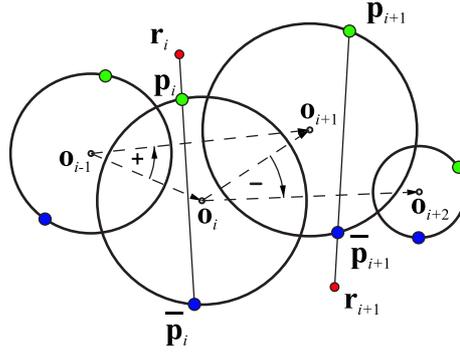


Figure 3.14. Grouping of the constructed touching points into two groups, “left” and “right” (green and blue). Detailed explanation can be found in the text.

Definition of the tangent vectors Finally we have two groups of well defined touching points on the circles. Between each pair of points an Hermite interpolation curve will be computed at each group, separately. To these arcs, one has to define the length of the tangent vectors at these points (the direction of the tangents is inherited from the actual circle). In Slabaugh’s method the length of the tangent vectors was a simple function of the radius of the current circle. This method works well if there is no large difference between the radii and the distance of the consecutive circles. Contrary to that method we specify the length of the tangent in a way that beside the radii, the distance of the circles, as significant information, is also incorporated. The radical line of two circles provides information about the radii and the distance of the circles as well. Thus we use this line to obtain unified information about the positions and size of the circles. For two consecutive circles and touching points \mathbf{p}_i and \mathbf{p}_{i+1} , the distances of these points to the radical line are computed. Multiplying this distance by a scalar value the length of the tangent vector is given. This scalar can be considered as a global shape parameter of the skin, similarly to the scalar at the energy function of Slabaugh’s method. In our experience the value 2 provided the most natural shape, thus this value is applied throughout our approach (see Fig. 3.15).

Construction of the skin Our final step is to construct the curve which is now a simple interpolation problem for given points \mathbf{p}_i and tangent vectors \mathbf{v}_i . At this point it is irrelevant that these data are computed from a set of circles, so we have to emphasize that other interpolation methods may work as well as our choice, the Hermite interpolation. We define a cubic curve $\mathbf{q}(t)$, $t \in [0, 1]$, where

$$\mathbf{q}(0) = \mathbf{p}_i, \quad \mathbf{q}(1) = \mathbf{p}_{i+1}, \quad \mathbf{q}'(0) = \mathbf{v}_i, \quad \mathbf{q}'(1) = \mathbf{v}_{i+1},$$

and

$$\mathbf{q}(t) = H_0^3(t)\mathbf{q}(0) + H_1^3(t)\mathbf{q}(1) + H_2^3(t)\mathbf{q}'(0) + H_3^3(t)\mathbf{q}'(1), \quad t \in [0, 1],$$

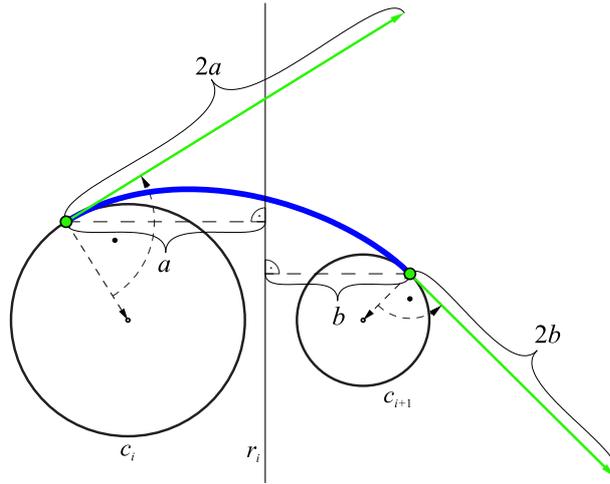


Figure 3.15. Definition of tangent vectors at the touching points by the help of radical line r_i

where

$$H_0^3(t) = 2t^3 - 3t^2 + 1$$

$$H_1^3(t) = -2t^3 + 3t^2$$

$$H_2^3(t) = t^3 - 2t^2 + t$$

$$H_3^3(t) = t^3 - t^2.$$

The Hermite interpolation arcs computed from these data serve as a G^1 continuous skin of the given circles. Results can be seen in Fig. 3.16, comparison to Slabaugh's method is in Fig. 3.11 and Fig. 3.17.

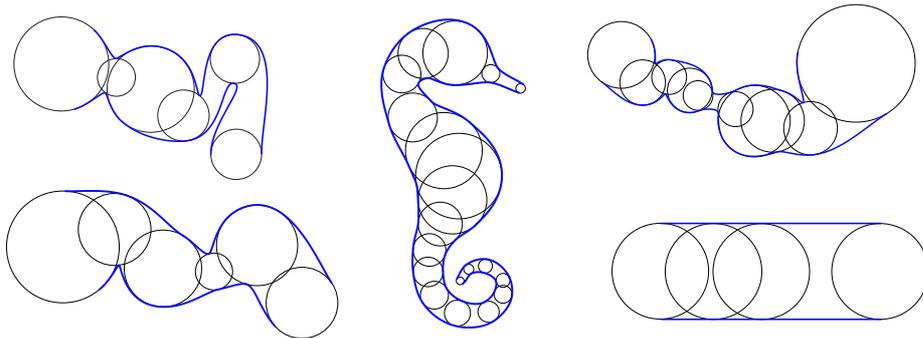


Figure 3.16. Results of the proposed method. It works for rather complicated data sets while provides correct result in the simplest case as well.

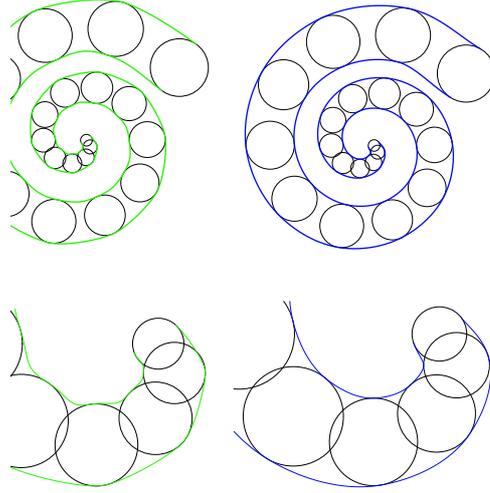


Figure 3.17. Comparison of Slabaugh's method (left) and the proposed method (right). Note the difference in the zoomed part below.

3.2.3 Extension to spheres

As aforesaid, we can define the problem in larger dimensions, too. Given an ordered set of (hyper)spheres, we are looking for a skinning (hyper)surface with similar properties as it had in 2D. Definition 3.4 of admissible configuration of circles can directly be applied to spheres. It is a consequence of the second point of this definition, that the union of balls may not be connected but each connected component must have genus 0. The definition of skin is however, different from the one given in Def. 3.2. We would like to obtain a G^1 continuous surface, which touches each sphere along a circle, that is tangent to the spheres.

Definition 3.3. Given an admissible configuration of spheres $C = \{s_1, s_2, s_3, \dots, s_n\}$, we are looking for a G^1 continuous surface $\mathbf{s}(\phi, t)$ of the given spheres, called *skin*, satisfying the following requirements:

- There is a circle of contact (touching circle) c_i for all $i = 1, \dots, n$ such that the skin $\mathbf{s}(\phi, t)$ and sphere s_i have common tangent planes at each point of c_i . Circle c_i is an isoparametric curve of $\mathbf{s}(\phi, t)$.

- $c_i \subset \bigcup_{j=1, j \neq i}^n d_j, \quad i \in \{1, 2, \dots, n\}.$

We have to emphasize again that, due to the last restriction, circle of contact on the actual sphere is required to be out of other spheres, which is a natural condition for skinning from our point of view.

Steps of our solution are analogous to that ones applied in the planar case.

First of all, we have to localize the touching circles with centers $\tilde{\mathbf{o}}_i$ and radii \tilde{r}_i ($i = 1, \dots, n$). For this step we can invoke the solution of the planar problem. Let us consider a sphere s_i , where $i = 2, \dots, n - 1$, that is we exclude the first and the last spheres for a moment (Fig.3.19).

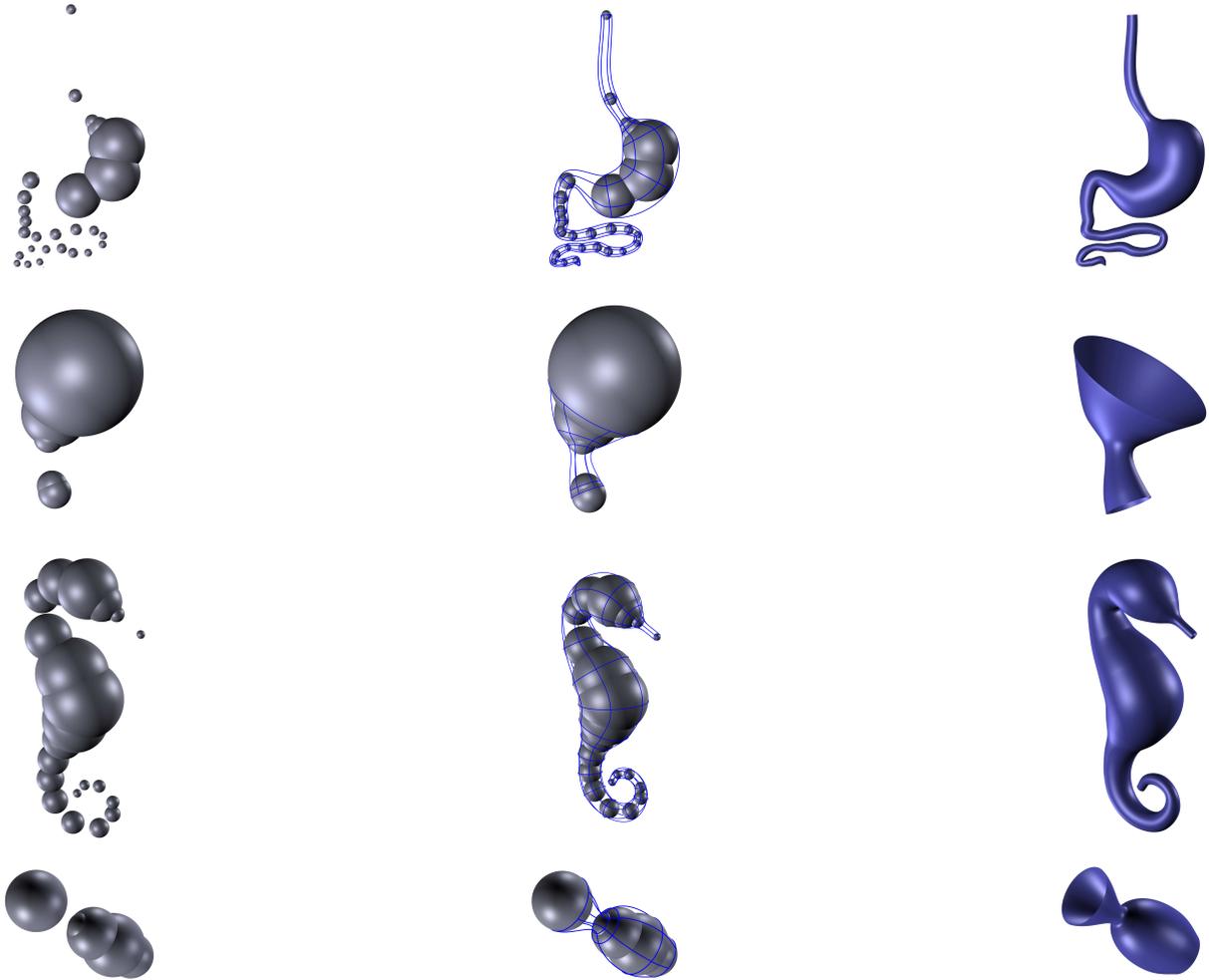


Figure 3.18. The algorithm can handle distant spheres as well as sudden changes in size. Note that the upper part of the stomach or the neck of the vase cannot be modeled by great circles as touching circles.

Now consider the plane P_i , determined by the centers $\mathbf{o}_{i-1}, \mathbf{o}_i, \mathbf{o}_{i+1}$ of the considered sphere and its neighbors. Intersecting the spheres by this plane we obtain three circles. With the help of the above mentioned planar method with Apollonius circles, we can find two points in the second circle. There exists exactly one plane T_i (for all $i = 2, \dots, n - 1$), which passes through these two points and orthogonal to plane P_i . The intersection of sphere s_i and this orthogonal plane T_i is the touching circle for the future skinning surface. We can localize a circle by this method on every sphere, which has two neighbors. Due to the facts that the touching circles are directly computed from the planar touching points and the admissible configuration guarantees the existence and proper positions of these points, the touching circles exist and satisfy the requirements of Definition 3.

The Apollonius problem itself can also be generalized in 3D, where touching spheres of three given spheres have to be found. The envelope of these spheres is the Dupin cyclide, which surface is widely used in CAGD (for an overview, see [25, 94, 95]). The touching circle we

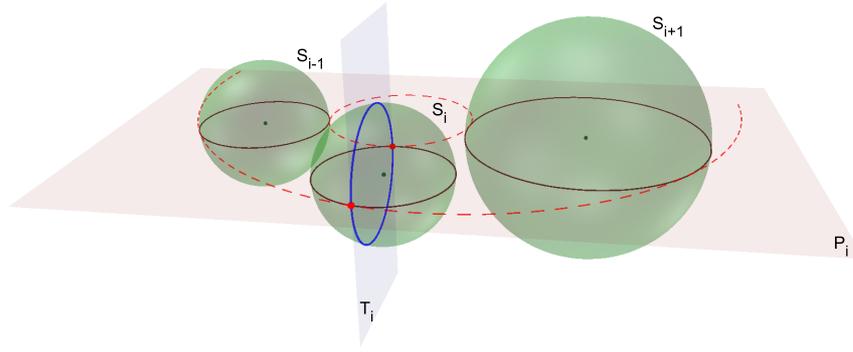


Figure 3.19. Touching circle localization on the sphere s_i , where $i \neq 1, n$. $T_i \perp P_i$, where T_i is the plane of the touching circle and P_i is the plane passing through the centers. Dashed line shows Apollonius circles as solution of the planar problem in P_i . Note that in general the touching circle is not a great circle of the sphere s_i .

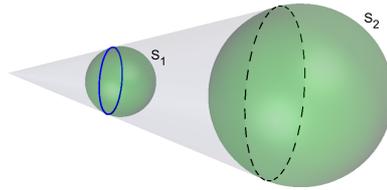


Figure 3.20. Construction of touching circle on s_1

constructed now is identical to the one in which the Dupin cyclide defined by the three given spheres s_{i-1}, s_i, s_{i+1} touches the sphere s_i .

Touching circles for the first and the last spheres have to be defined in a different way. Let us consider the first and the second spheres and the regular cone which touches both spheres. The touching circle of this cone on the first sphere will be the touching circle for the skinning surface as well (Fig. 3.20). The circle on the last sphere is defined analogously.

It directly follows from the planar construction, that this method always works in every admissible case.

Now we obtained touching circle with center $\tilde{\mathbf{o}}_i$ and radius \tilde{r}_i on each sphere, thus we can start to create the skin, following the ideas developed in the planar case: patches are defined successively to each pair of spheres using Hermite interpolants through corresponding points of the touching circles.

Consider the future patch $\mathbf{s}_i(\phi, t)$ of the skin between touching circle c_i on sphere s_i and touching circle c_{i+1} on sphere s_{i+1} . Circle c_i is the isoparametric curve $\mathbf{s}_i(\phi, 0)$, while c_{i+1} is the isoparametric curve $\mathbf{s}_i(\phi, 1)$ of this patch. At first we will define the starting point on c_i as $\mathbf{z}_i = \mathbf{s}_i(0, 0)$ and on c_{i+1} as $\mathbf{z}_{i+1} = \mathbf{s}_i(0, 1)$, then rotating them by the same angle $\phi \in [0, 2\pi]$ along the circles, corresponding pairs of points $\mathbf{z}_i(\phi) = \mathbf{s}_i(\phi, 0)$, $\mathbf{z}_{i+1}(\phi) = \mathbf{s}_i(\phi, 1)$ will be defined.

Lengths of tangent vectors are computed by the help of the radical plane of the two spheres. Lines of all the tangents pass through the pole \mathbf{w}_i of the plane of the touching circle with respect to the sphere s_i ($i = 1, \dots, n$), see Fig. 3.21. To avoid unnecessary torsion, corresponding points are selected by the help of a fixed spatial direction \mathbf{e} , which is not parallel to any of the vectors $\mathbf{w}_i - \tilde{\mathbf{o}}_i$, for example $\mathbf{e} = (\mathbf{w}_i - \tilde{\mathbf{o}}_i) \times (\mathbf{w}_{i+1} - \tilde{\mathbf{o}}_{i+1})$.

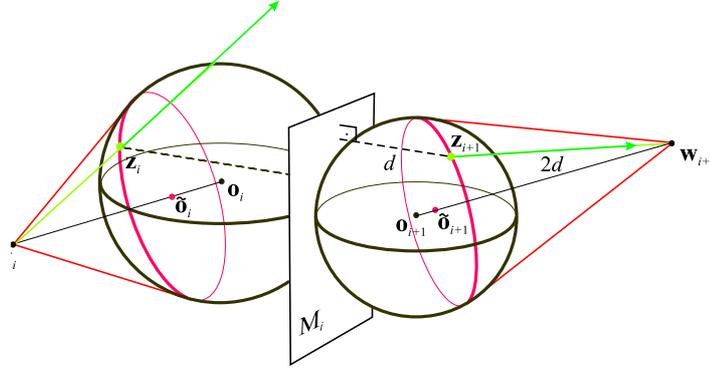


Figure 3.21. Computation of tangent lengths is analogous to the planar case, now using radical plane M_i . Corresponding points \mathbf{z}_i and \mathbf{z}_{i+1} are connected by isoparametric curve of the skin surface.

Let $s: \mathbb{R} \rightarrow \{-1, 1\}$ and $p: \{s_i\} \rightarrow \{-1, 1\}$ be functions defined by

$$s(x) = \begin{cases} -1 & \text{if } x < 0, \\ 1 & \text{else} \end{cases}$$

$$p(s_i) = \begin{cases} s\left(\left\langle \frac{\mathbf{w}_i - \mathbf{o}_i}{\|\mathbf{w}_i - \mathbf{o}_i\|}, \frac{\mathbf{o}_{i+1} - \mathbf{o}_i}{\|\mathbf{o}_{i+1} - \mathbf{o}_i\|} \right\rangle\right) & \text{if } i \neq n, \\ s\left(\left\langle \frac{\mathbf{w}_i - \mathbf{o}_i}{\|\mathbf{w}_i - \mathbf{o}_i\|}, \frac{\mathbf{o}_i - \mathbf{o}_{i-1}}{\|\mathbf{o}_i - \mathbf{o}_{i-1}\|} \right\rangle\right) & \text{else} \end{cases}$$

where $\langle \cdot, \cdot \rangle$ is the standard inner product.

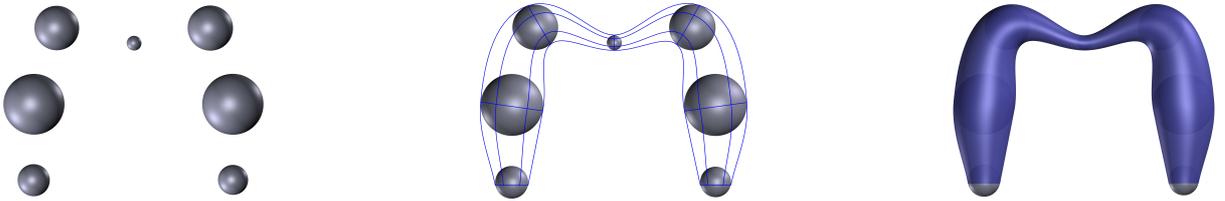


Figure 3.22. Result of the proposed algorithm. This data set has also been used in [104], but, contrary to that algorithm, our method always provide symmetric skin for symmetric data.

Let \mathbf{z}_i be defined by

$$\mathbf{z}_i = \tilde{\mathbf{o}}_i + \tilde{r}_i \cdot \frac{\mathbf{e} \times (p(s_i) \cdot (\mathbf{w}_i - \tilde{\mathbf{o}}_i))}{\|\mathbf{e} \times (p(s_i) \cdot (\mathbf{w}_i - \tilde{\mathbf{o}}_i))\|} \quad (i = 1, 2, \dots, n).$$

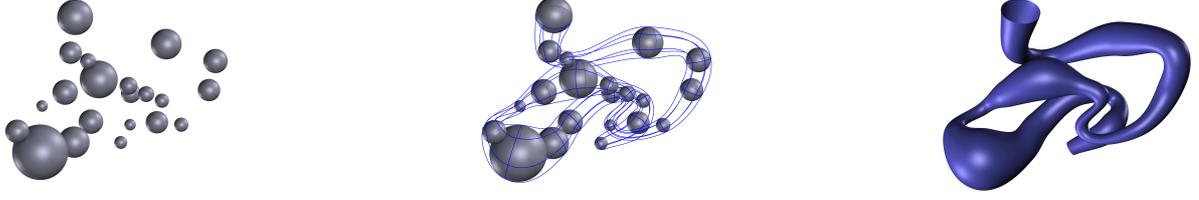


Figure 3.23. A complex data set and its skin produced by the proposed algorithm. Note, that centers of spheres are not coplanar.

Further corresponding points $\mathbf{z}_i(\phi)$ of the touching circles are defined by rotating \mathbf{z}_i by angle ϕ around the line passing through $\tilde{\mathbf{o}}_i$ and having direction $\mathbf{p}(s_i) \cdot (\mathbf{w}_i - \tilde{\mathbf{o}}_i)$.

Let M_i be the radical plane of spheres s_i and s_{i+1} . Our skinning surface contains $n - 1$ patches, where the i th patch is defined as:

$$\begin{aligned} \mathbf{s}_i(\phi, t) = & H_0^3(t) \mathbf{z}_i(\phi) + H_1^3(t) \mathbf{z}_{i+1}(\phi) + \\ & H_2^3(t) \cdot \mathbf{p}(s_i) \cdot 2 \cdot d(M_i, \mathbf{z}_i(\phi)) \cdot \frac{\mathbf{w}_i - \mathbf{z}_i(\phi)}{\|\mathbf{w}_i - \mathbf{z}_i(\phi)\|} + \\ & H_3^3(t) \cdot \mathbf{p}(s_{i+1}) \cdot 2 \cdot d(M_i, \mathbf{z}_{i+1}(\phi)) \cdot \frac{\mathbf{w}_{i+1} - \mathbf{z}_{i+1}(\phi)}{\|\mathbf{w}_{i+1} - \mathbf{z}_{i+1}(\phi)\|} \\ & t \in [0, 1], \phi \in [0, 2\pi], i = 1, \dots, n - 1 \end{aligned}$$

where $d()$ is the Euclidean distance function and H_i^3 are the cubic Hermite-polynomials.

Results of the algorithm can be seen in Figure 3.22, 3.23, and 3.18.

3.2.4 Surfaces with multiple branches

The main contribution of this subsection is to extend our previous method in order to handle surfaces with multiple branches, based on the algorithm developed in [2]. The method presented here is based on the idea of our prior algorithm, but it is extended to shapes with several branches, which can help the creation of animation characters or more structured tubular surfaces.

The final input of our method is a rooted tree graph in 3D, where at each node we have a sphere. The spheres can have different radii. During the modeling phase the graph is built from one single sequence of edges (spheres) and can be modified in an interactive way, by adding new branches to the tree and altering the radii and positions of the spheres. In one branch (one sequence of spheres) the admissible positions of spheres are defined as follows.

Definition 3.4. A sequence of spheres $C = \{s_1, s_2, s_3, \dots, s_n\}$ ($n \in \mathbb{N}$) is called *admissible configuration* if the following conditions are fulfilled:

- $s_i \subset \bigcup_{j=1, j \neq i}^n s_j, \quad i \in \{1, 2, \dots, n\}$
- $s_i \cap s_j = \emptyset, \quad i, j \in \{1, 2, \dots, n\}, j \notin \{i - 2, i - 1, i, i + 1, i + 2\}$

- if $s_{i-1} \cap s_{i+1} \neq \emptyset$, then $s_{i-1} \cap s_{i+1} \subset s_i$

These conditions can easily be checked by simple computation.

Conditions of adding a new branch to the existing tree are as follows:

- spheres of the new branch have to form an admissible configuration described in Def.3.4
- the first sphere of the new branch has to be one of the spheres of the existing structure
- none of the spheres of the new branch (except, of course the first one) can intersect any of the other spheres of the existing structure.

These assumptions are natural restrictions in order to avoid intersecting branches and closed loops, but at the same time the construction involves (i.e. the surface touches) all of the given spheres. As it is described in [69] the G^1 continuity of the surface is guaranteed along the branches, while at the junctions of the branches it is also assured by the new method as described in Section 3.2.4. All given spheres are touched along a circle, and the surface does not intersect the spheres, as it follows from the original algorithm.

This structure is appropriate for an important set of applications, including medical and biological applications and constructing characters. In theory, our method could handle more than three neighbors of one single sphere, which would yield junctions with more than two branches started from the same sphere. But in this case the smoothness of the joining patches is not always ensured, especially when the curve along which a new branch is connected to the existing structure has arcs in the surface of two or more existing branches. In Figure 3.33 one can observe this kind of junctions. The possible extension of the presented method in order to handle more complicated structures can be the direction of future research.

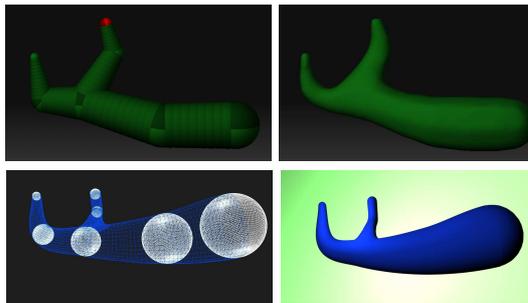


Figure 3.24. Defined by an ordered set of spheres, the software ZSpheres[®] (above) applies subdivided cylinders and cones, while our method (below) computes a parametric surface between the spheres. The given spheres with the blending pieces (left) and the final, rendered surfaces (right) can be seen in both methods.

New branches To extend the original method, in this section we provide the algorithm for adding new branches to the existing structure. The first step is to choose the starting sphere among the existing spheres and to define a new admissible sequence of spheres by the user. Let

us denote the starting sphere by s_i , chosen from the existing branch of spheres $\dots s_{i-1}, s_i, s_{i+1}, \dots$, and denote the spheres of the new branch by $p_{ij}, j = 1, \dots, m$, where $p_{i1} = s_i$. The new spheres $p_{ij}, j = 2, \dots, m$ can be blended by the original algorithm described above, thus the main goal is to smoothly connect this new branch with the branch of s_i (see Fig.3.28, where the blue patches are computed by the original algorithm in both branches, while the new red patch is to connect the new branch to the original one). Since the original algorithm provides G_1 continuous surface along the branches, it is a natural requirement to produce a G_1 connection between the branches as well. It is also worth mentioning that this method is not symmetric in terms of branches, that is there is a "parent" branch and "child" branch(es) in each connection, similarly to a rooted tree graph.

Details of the computation of the connecting (red) patch of the two branches are provided in the next subsections.

The new touching circle Let us consider three neighbouring spheres s_{i-1}, s_i, s_{i+1} from the original sequence and assume that we would like to connect a new branch starting at s_i .

At first we determine a new touching circle on s_i from where the new branch can start. For this purpose we apply the basic algorithm for sphere triplets s_{i-1}, s_i, p_{i2} and s_{i+1}, s_i, p_{i2} , respectively, to obtain two circles on s_i , c_{i1} and c_{i2} (Figure 3.25).

The new touching circle (let us denote it by c'_i) is fitting on the common points of c_{i1} and c_{i2} . We can determine its normal vector \mathbf{n}'_i as the sum of the normalized normal vectors of c_{i1} and c_{i2} , $\mathbf{n}_{i1}, \mathbf{n}_{i2}$, respectively. Actually the plane of c'_i is the bisector plane of the planes of c_{i1} and c_{i2} . In most cases the common points of c_{i1} and c_{i2} exist, if not, then we can consider the plane passing through the center of c_{i1} and having normal vector \mathbf{n}'_i . The intersection of this plane and the sphere p_{i1} will be the circle c'_i in question.

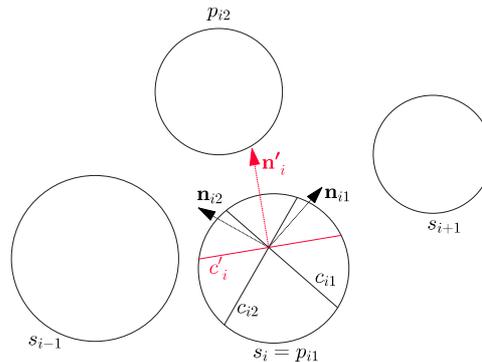


Figure 3.25. Constructing new touching circle (red) for the joining branch.

Although other methods of creating the new circle may also work well, it is important to note that we have constructed a new touching circle on s_i with a method which is simple, and sensitive to its neighbours. The simplicity is important in order to preserve the real time computation ability, while the sensitivity is especially advantageous when neighbouring spheres have drastically different radii. In our practice the method behaved correctly in any admissible

circumstances.

After this step we can construct a new branch starting from $s_i = p_{i1}$ with the help of the original algorithm, blending the spheres p_{ij} , ($j = 1, \dots, m$). This way the branch surfaces will not be connected smoothly, but they will have a sharp intersection.

In the following part we describe how we can achieve a G^1 continuous connection of the branches. The original algorithm is applied only from the second sphere of the new branch (to the spheres p_{i2}, p_{i3}, \dots , etc.), and a smooth connection patch is created between the sphere p_{i2} and the original branch (the branch of s_i). For this purpose we create a boundary curve on the original branch and the sphere p_{i2} will be connected to the original branch by a patch which will touch the original branch along this curve in a G_1 continuous way.

Boundary curve for G^1 continuous connection At first we determine a point \mathbf{m}_i on circle c_i which is the original touching circle on the sphere s_i for the original ("parent") branch. This point will be the so-called "midpoint" of the closed boundary curve.

To define point \mathbf{m}_i , let \mathbf{n}_i denote the normal vector of c_i , $\|\mathbf{n}_i\| = 1$. We would like to find vector \mathbf{h}_i such that

$$\mathbf{n}'_i = \mathbf{h}_i + \lambda \cdot \mathbf{n}_i \quad \text{and} \quad \langle \mathbf{h}_i, \mathbf{n}_i \rangle = 0,$$

where $\lambda \in \mathbb{R}$.

From equation $\langle \mathbf{n}'_i - \lambda \cdot \mathbf{n}_i, \mathbf{n}_i \rangle = 0$ we can easily calculate the value of λ , so \mathbf{h}_i can be determined as $\mathbf{h}_i = \mathbf{n}'_i - \lambda \cdot \mathbf{n}_i$. After this step \mathbf{h}_i can be used to describe vector \mathbf{m}_i :

$$\mathbf{m}_i \doteq \tilde{\mathbf{o}}_i + \tilde{r}_i \cdot \frac{\mathbf{h}_i}{\|\mathbf{h}_i\|}.$$

Practically this is an orthogonal projection of a special representant of \mathbf{n}'_i to the plane of c_i (for

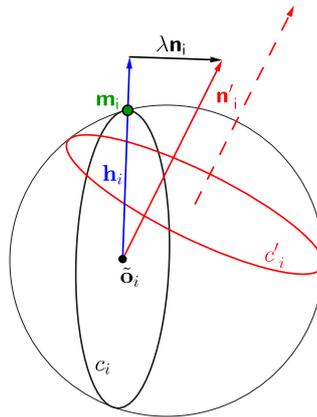


Figure 3.26. Constructing the center of the boundary curve.

the notations see Figure 3.26).

Now we can define a continuous boundary curve on the blending surface of the original branch along which the new branch will touch this original branch. It is clear from the computation (see

Figure 3.26) that \mathbf{m}_i fits on c_i . Let us consider \mathbf{z}_i , the starting point of the parameterization of circle c_i . Based on the notes of [69] we can determine an angle $\alpha_i \in [0, 2\pi]$, such that rotating \mathbf{z}_i along the circle c_i by α_i , we reach the point \mathbf{m}_i , that is $\mathbf{z}_i(\alpha_i) = \mathbf{m}_i$ holds.

Now we define the two arcs \mathbf{L}_{i1} and \mathbf{L}_{i2} of the boundary curve. Let us consider the following arc:

$$\begin{aligned} \mathbf{L}_{i1}(\theta) = & H_0^3(t_0) \mathbf{z}_i(\alpha_i + \theta) + H_1^3(t_0) \mathbf{z}_{i+1}(\alpha_i + \theta) + \\ & H_2^3(t_0) \cdot p(s_i) \cdot 2 \cdot d(M_i, \mathbf{z}_i(\alpha_i + \theta)) \cdot \frac{\mathbf{w}_i - \mathbf{z}_i(\alpha_i + \theta)}{\|\mathbf{w}_i - \mathbf{z}_i(\alpha_i + \theta)\|} + \\ & H_3^3(t_0) \cdot p(s_{i+1}) \cdot 2 \cdot d(M_i, \mathbf{z}_{i+1}(\alpha_i + \theta)) \cdot \\ & \frac{\mathbf{w}_{i+1} - \mathbf{z}_{i+1}(\alpha_i + \theta)}{\|\mathbf{w}_{i+1} - \mathbf{z}_{i+1}(\alpha_i + \theta)\|}, \end{aligned}$$

where $t_0 = \frac{4q}{\pi} \sqrt{(\frac{\pi}{4})^2 - \theta^2}$ (based on the equation $y = \sqrt{r^2 - x^2}$ of a semicircle with radius r and centered at the origin), $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ and $q \in]0, 1[$. Increasing q the boundary curve will run closer to the neighbouring spheres. This curve will be one arc of the boundary curve between s_i and s_{i+1} . The second arc is defined from s_i to s_{i-1} by

$$\begin{aligned} \mathbf{L}_{i2}(\theta) = & H_0^3(1 - t_0) \mathbf{z}_{i-1}(\alpha_i + \theta) + H_1^3(1 - t_0) \mathbf{z}_i \cdot \\ & (\alpha_i + \theta) + H_2^3(1 - t_0) \cdot p(s_{i-1}) \cdot 2 \cdot d(M_{i-1}, \mathbf{z}_{i-1} \cdot \\ & (\alpha_i + \theta)) \cdot \frac{\mathbf{w}_{i-1} - \mathbf{z}_{i-1}(\alpha_i + \theta)}{\|\mathbf{w}_{i-1} - \mathbf{z}_{i-1}(\alpha_i + \theta)\|} + H_3^3(1 - t_0) \cdot p(s_i) \cdot \\ & 2 \cdot d(M_{i-1}, \mathbf{z}_i(\alpha_i + \theta)) \cdot \frac{\mathbf{w}_i - \mathbf{z}_i(\alpha_i + \theta)}{\|\mathbf{w}_i - \mathbf{z}_i(\alpha_i + \theta)\|}, \end{aligned}$$

where t_0 and θ has the same value as above. The two arcs of the boundary curve can be seen in Figure 3.27.

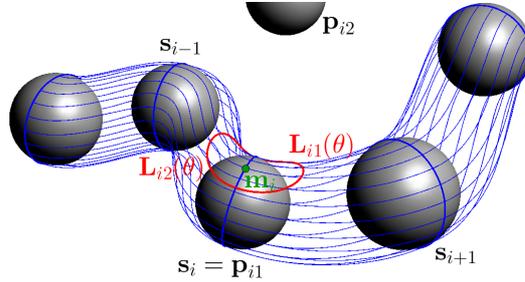


Figure 3.27. The two arcs of the boundary curve, $q = 0.3$.

As we have mentioned previously, with the help of the basic algorithm and the new touching circle c'_i on sphere s_i we can determine touching circles on the spheres of the new branch. So to create a G^1 continuous connection from the boundary curve we have to consider its points and assign endpoints on the touching circle of p_{i2} to them. Let us denote this circle by c_{i2} .

With the above mentioned technique based on orthogonal projection we can localize a matching point for $\mathbf{L}_{i1}(0)$ by projecting vector $\mathbf{L}_{i1}(0) - \mathbf{m}_i$ (the starting point is the center of c'_i) onto the plane of c_{i2} . This point will be the endpoint of the Hermite arc starting at $\mathbf{L}_{i1}(0)$. Then with rotations by angles between 0 and 2π the first part of the blending surface of the new branch can be constructed from s_i to p_{i2} analogously to the basic algorithm (see Figure 3.28).

To compute the tangent vectors at the points of the boundary curve, the tangent plane of

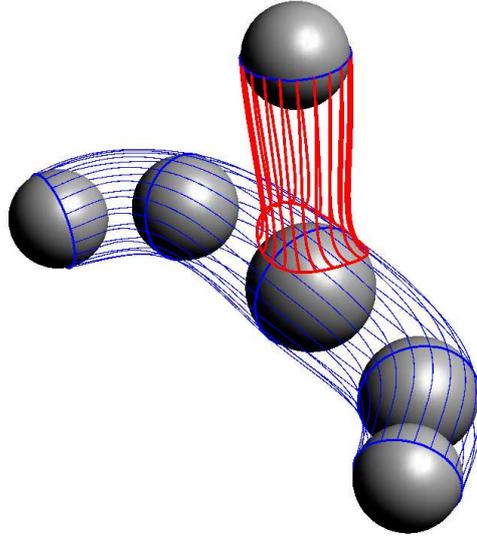


Figure 3.28. Connection of two branches. Hermite arcs (isoparametric curves of the patch), starting at the boundary curve can be seen.

the original surface $\mathbf{s}_i(\phi, t)$ has to be computed first. The partial derivatives of the surface are as follows

$$\begin{aligned} \frac{\partial}{\partial \phi} \mathbf{s}_i(\phi, t) = & H_0^3(t) \dot{\mathbf{z}}_i(\phi) + H_1^3(t) \dot{\mathbf{z}}_{i+1}(\phi) \\ & + H_2^3(t) \cdot \dot{\mathbf{v}}_i(\phi) + H_3^3(t) \cdot \dot{\mathbf{v}}_{i+1}(\phi) \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial t} \mathbf{s}_i(\phi, t) = & \frac{d}{dt} H_0^3(t) \mathbf{z}_i(\phi) + \frac{d}{dt} H_1^3(t) \mathbf{z}_{i+1}(\phi) \\ & + \frac{d}{dt} H_2^3(t) \cdot \mathbf{v}_i(\phi) + \frac{d}{dt} H_3^3(t) \cdot \mathbf{v}_{i+1}(\phi). \end{aligned}$$

The normal vector of the tangent plane will be the cross product of the partial derivatives.

Now we use the described orthogonal projection again to create tangent vectors at each point of the boundary curve. For this purpose we project orthogonally the vector $\mathbf{L}_{i1}(\theta) - \mathbf{m}_i$ onto the tangent plane at $\mathbf{L}_{i1}(\theta)$ for each $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. The length of the tangent vector is the distance of the point from the radical plane of the two spheres multiplied by 2.

3.2.5 Results and comparison with other methods

Based on the theoretical results described above, an easy-to-use software tool is provided to create surfaces and characters by spheres. We used our own C libraries for the required geometrical calculations and OpenGL for the rendering process. The software is able to generate its output in EPS format, using the Asymptote vector graphics language. We can define spheres, adjust their positions and radii, and choose a sphere from where a new branch will start. The blending surface is computed in real-time, automatically, with several possibilities of modification (colour, rendering etc.).

As we have mentioned, this kind of tools have already been introduced in computer graphics, but in several cases our method provides better results, especially in terms of connection of

branches. This problem of smooth connection is especially noteworthy when branches meet at small spheres, that is the radius of the sphere $p_{i1} = s_i$ is much smaller than the radii of s_{i-1} , s_{i+1} and p_{i2} . It can cause unwanted and sometimes unacceptable forms in other softwares, while our method is not sensitive to the suddenly changed radii of the given spheres. In Figure 3.29 and 3.30 one can observe the problematic issue of connection of branches in the case of ZSpheres[®]. If the new branch starts at a relatively small sphere, then the obtained branches can be connected in an unpredictable manner (Figure 3.30), and/or can have unwanted shape, such as flat triangle-like shape at the connection (Figure 3.29). In these cases our software provides a more natural connecting patch. In case of SporeTM, the other alternative of sphere based modeling tools, the smooth connection of branches is not everywhere solved in a satisfactory way, the surface can have crisps or sharp edges at this point (see Figure 3.31), while our method can provide smooth connection of different branches.

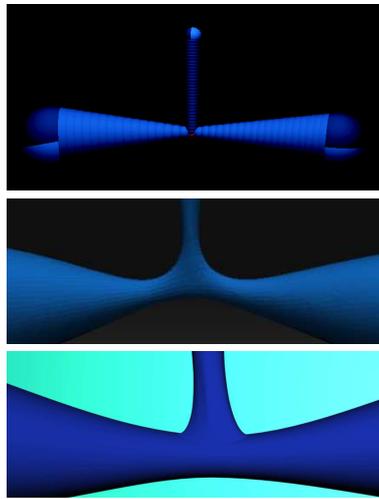


Figure 3.29. Given a simple join in ZSpheres[®] (above and middle) with small sphere at the connection, the resulted ZSpheres[®] surface can have unwanted triangle-shape flat part at the connection (middle). With similar input, our method provides more natural connection of branches (below).

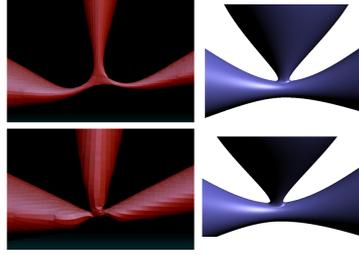


Figure 3.30. If the radii of spheres are drastically changed around the connection, ZSpheres[®] surfaces (left) can have unpredictable behaviour at the join. Our method (right) can handle this problem (branch connection with smoothing).

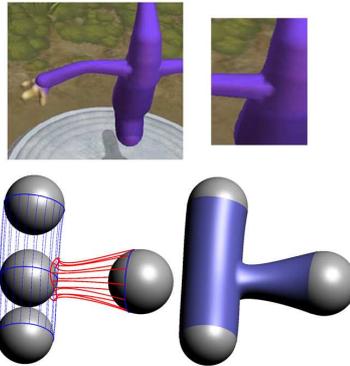


Figure 3.31. In Spore[™], connection of branches are less attractively solved (above). Our software provides smoother (G^1 continuous) connection (below).

Our method can also handle several branches of different size and shape, multiple connections, spheres with neighbours having significantly different radius, as well as neighbours intersecting each other, see Figure 3.32 and Figure 3.33.

dc_933_14

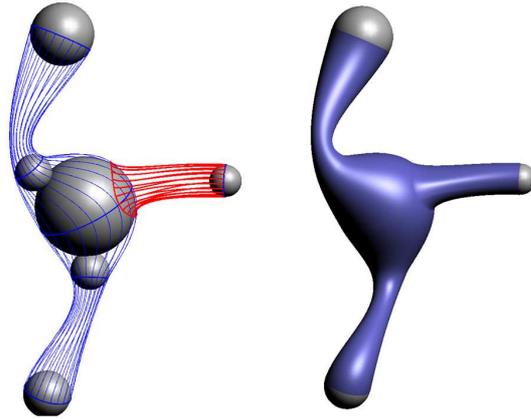


Figure 3.32. Our presented method can also handle situations when the radii and distance of the neighbouring spheres vary significantly, and the neighbouring spheres can also be intersecting.

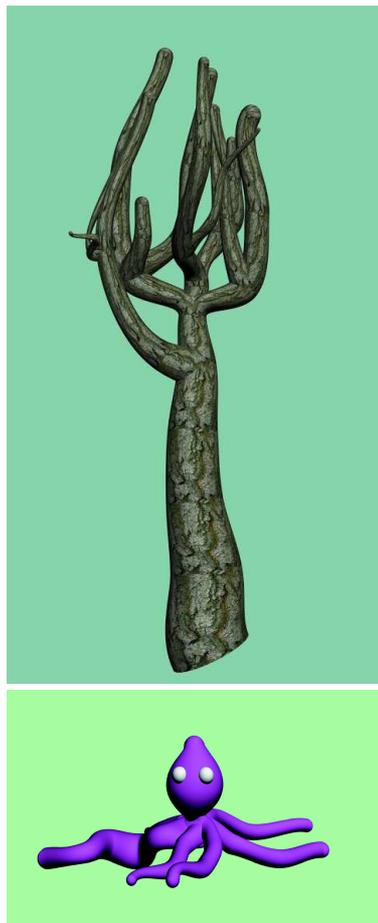


Figure 3.33. Our method can handle several branches and multiple connections as well.

Bibliography

- [1] Au, C. K., Yuen, M. M. F.: Unified approach to NURBS curve shape modification, *Computer-Aided Design*, 27 (1995), 85–93.
- [2] Bana, K., Kruppa, K., Kunkli, R., Hoffmann, M.: KSpheres - An Efficient Algorithm for Joining Skinning Surfaces, *Computer Aided Geometric Design*, 31 (2014), 499–509.
- [3] Barhak, J., Fischer, A.: Parameterization and reconstruction from 3D scattered points based on neural network and PDE techniques, *IEEE-TVCG* 7 (2001), 1–16.
- [4] Barhak, J., Fischer, A.: Adaptive reconstruction of freeform objects with 3D SOM neural network grids, *Computers & Graphics*, 26 (2002), 745–751.
- [5] Barsky, B.A.: *Computer graphics and geometric modeling using β -splines*, Springer-Verlag, Berlin, 1988.
- [6] Barsky, B.A., Beatty, J.C.: Local control of bias and tension in β -splines, *ACM Transactions on Graphics*, 2 (1983), 109–134.
- [7] Bastl, B., Kosinka, J., Lavicka, M.: Simple and branched skins of systems of circles and convex shapes, *Graphical Models*, doi:10.1016/j.gmod.2014.12.001 (megjelenés alatt)
- [8] Boehm, W.: Inserting new knots into B-spline curves, *Computer-Aided Design*, 12 (1980), 199–201.
- [9] Boehm, W.: On the efficiency of knot insertion algorithms, *Computer Aided Geometric Design*, 2 (1985), 141–143.
- [10] de Boor, C.: *A Practical Guide to Splines*. Springer-Verlag, 1978.
- [11] Bosner, T., Rogina, M.: Variable degree polynomial splines are Chebyshev splines. *Advances in Computational Mathematics*, 38 (2013), 383–400.
- [12] Bressoud, D.M.: *Proofs and Confirmations: The Story of the Alternating Sign Matrix Conjecture*, MAA Spectrum Series. Cambridge University Press, Cambridge, 1999, 162–164.
- [13] Butterfield, K. R.: The computation of all the derivatives of a B-spline basis. *Jour. Inst. Math. Appl.*, 17 (1976), 15–25.
- [14] de Casteljau, P.: *Courbes à pôles*. Francia Ipari Szabványügyi Hivatal, 1959.
- [15] Chen, Q., Wang, G.: A class of Bézier-like curves, *Computer Aided Geometric Design*, 20 (2003), 29–39.
- [16] Cheng, H., Shi, X., 2005. Quality mesh generation for molecular skin surfaces using restricted union of balls, *Proc. IEEE Visualization Conference (VIS2005)*, 2005 399–405.

- [17] Coaklay, G. W.: Analytical Solutions of the Ten Problems in the Tangencies of Circles and also of the Fifteen Problems in the Tangencies of Spheres. *The Mathematical Monthly* 2 (1860), 116–126.
- [18] Costantini, P., Lyche, T., Manni, C.: On a class of weak Tchebycheff systems. *Numerische Mathematik*, 101 (2005), 333–354.
- [19] Dörrie, H.: 100 Great Problems of Elementary Mathematics: Their History and Solutions. Dover Publishing, New York, 1965.
- [20] Echevarría, G., Iglesias, A., Gálvez, A.: Extending neural networks for B-spline surface reconstruction, *Lecture Notes in Computer Science* 2330 (2002), 305–331.
- [21] Edelsbrunner, H.: Deformable smooth surface design. *Discrete and Computational Geometry* 21 (1999), 87–115.
- [22] Farin, G., Rein, G., Sapidis, N., Worsey, A.J.: Fairing cubic B-spline curves, *Computer Aided Geometric Design*, 4 (1987), 91–103.
- [23] Farin, G.: Rational curves and surfaces, in: Lyche, T., Schumaker, L.L. (eds.), *Mathematical methods in computer aided geometric design*, Academic Press, Boston, 1989.
- [24] Farin, G.: *Curves and Surface for Computer-Aided Geometric Design*, 4th edition, Academic Press, New York, 1997.
- [25] Farin, G., Hoschek, J., Kim, M-S.: *Handbook of computer aided geometric design*, Elsevier, 2002.
- [26] Fowler, B., Bartels, R.: Constraint-based curve manipulation, *IEEE Computer Graphics and Applications*, 13 (1993), 43–49.
- [27] J. Freeman, D. Skapura: *Neural Networks: Algorithms, Applications and Programming Techniques*, Addison-Wesley, 1991.
- [28] Fritzke, B.: Growing cell structures - a self-organizing network for unsupervised and supervised learning, *Neural Networks*, 7 (1994), 1441–1460.
- [29] Guo, Q.: Cubic GB-spline curves. *Journal of Information and Computational Science*, 3 (2005), 465–471.
- [30] Han, X.: Piecewise quartic polynomial curves with shape parameter. *Journal of Computational and Applied Mathematics*, 195 (2006), 34–45.
- [31] Han, X., Zhu, Y.: Curve construction based on five trigonometric blending functions, *BIT Numerical Mathematics*, 52 (2012), 953–979.
- [32] Hoffmann, M., Várady, L.: Free-form Surfaces for Scattered Data by Neural Networks, *Journal for Geometry and Graphics*, 2 (1998), 1–6.

- [33] M. Hoffmann, Modified Kohonen Neural Network for Surface Reconstruction, *Publ. Math.*, 54 (1999), 857-864.
- [34] Hoffmann, M.: On the derivatives of a special family of B-spline curves, *Acta Acad. Paed. Agr. Sect. Math.*, 28 (2001), 61–68.
- [35] Hoffmann, M., Juhász, I.: Geometric aspects of knot modification of B-spline surfaces, *Journal for Geometry and Graphics*, 6 (2002), 141–149.
- [36] Hoffmann, M., Kovács, E.: Developable surface modeling by neural networks, *Mathematical and Computer Modelling*, 38 (2003), 849–853.
- [37] Hoffmann, M., Juhász, I.: On the knot modification of a B-spline curve, *Publicationes Mathematicae*, 65 (2004), 193–203.
- [38] Hoffmann, M.: Numerical control of Kohonen neural network for scattered data approximation, *Numerical Algorithms* 39 (2005), 175–186.
- [39] Hoffmann, M., Juhász, I.: Constrained shape control of bicubic B-spline surfaces by knots, in: Sarfraz, M., Banissi, E. (eds.): *Geometric Modeling and Imaging*, London, IEEE CS Press, 2006, 41–47.
- [40] Hoffmann, M., Juhász, I.: On the family of B-spline surfaces obtained by knot modification, *Mathematical Communications*, 11 (2006), 9–16.
- [41] Hoffmann, M., Y., Li, G., Wang, G.-Zh.: Paths of C-Bézier and C-B-spline curves, *Computer Aided Geometric Design*, 23 (2006), 463–475.
- [42] Hoffmann, M., Juhász, I.: Modifying the shape of FB-spline curves, *Journal of Applied Mathematics and Computing*, 27 (2008), 257–269.
- [43] Hoffmann, M., Juhász, I.: On interpolation by spline curves with shape parameters, *Lecture Notes in Computer Science*, 4975 (2008), 205–215.
- [44] Hoffmann, M., Juhász, I.: On parametrization of interpolating curves, *Journal of Computational and Applied Mathematics*, 216 (2008), 413–424.
- [45] Hoffmann, M., Juhász, I.: On the quartic curve of Han, *Journal of Computational and Applied Mathematics*, 223 (2009), 124–132.
- [46] Hoffmann, M., Juhász, I., Károlyi, Gy.: A control point based curve with two exponential shape parameters, *BIT Numerical Mathematics*, 54 (2014), 691–710.
- [47] V. Hlavaty: *Differential line geometry*, Nordhoff, 1953.
- [48] Hoschek, J.: *Zur Ermittlung von Hüllgebilden in der Kinematik*, PhD thesis, TH Darmstadt, 1964.
- [49] Hoschek, J., Lasser, D.: *Fundamentals of CAGD*, AK Peters, Wellesley, MA, 1993.

- [50] Hui, K., Lai, Y.: Smooth blending of subdivision surfaces, *Computer-Aided Design*, 38 (2006) 786–799.
- [51] Ivriissimtzis, I.P., Jeong, W-K., Seidel, H-P.: Using Growing Cell Structures for Surface Reconstruction, *Proceedings of the Shape Modeling International '03*, 2003, 78–86.
- [52] Ivriissimtzis, I.P., Jeong, W-K., Seidel, H-P.: Neural Meshes: Statistical Learning Methods in Surface Reconstruction, *Tech. Report of Max-Planck Institut für Math.*, MPI-I-2003-4-007, 2003.
- [53] Jeong, W-K., Ivriissimtzis, I. P., Seidel, H-P.: Neural Meshes: Statistical Learning based on Normals, *Proc. of Pacific Graphics 03*, IEEE Computer Society Press, 2003, 404–408.
- [54] Johnson, W.P.: The curious history of Faà di Bruno’s formula. *American Mathematical Monthly*, 109 (2002), 217–234.
- [55] Juhász, I.: Weight-based shape modification of NURBS curves, *Computer Aided Geometric Design*, 16 (1999), 377–383.
- [56] Juhász, I.: On the singularity of a class of parametric curves, *Computer Aided Geometric Design*, 23 (2006), 146–156.
- [57] Juhász, I., Hoffmann, M.: The effect of knot modifications on the shape of B-spline curves, *Journal for Geometry and Graphics*, 5 (2001), 111–119.
- [58] Juhász, I., Hoffmann, M.: Modifying a knot of B-spline curves, *Computer Aided Geometric Design*, 20 (2003), 243–245.
- [59] Juhász, I., Hoffmann, M.: Modifying the shape of cubic B-Spline and NURBS curves by means of knots, in: *Advances in Geometric Modeling*, J. Wiley & Sons, 299–313, 2004.
- [60] Juhász, I., Hoffmann, M.: Constrained shape modification of cubic B-spline curves by means of knots, *Computer-Aided Design*, 36 (2004), 437–445.
- [61] Juhász, I., Hoffmann, M.: On the quartic curve of Han, *Journal of Computational and Applied Mathematics*, 223 (2009), 124–132.
- [62] Károlyi, Gy., Keleti, T., Kós, G., Ruzsa, I.Z.: Periodic decomposition of integer valued functions. *Acta Mathematica Hungarica*, 119 (2008), 227–242.
- [63] Knopf, G. K., Sangole, A.: Interpolating scattered data using 2D self-organizing feature maps, *Graphical Models*, 66 (2004), 50–69.
- [64] T. Kohonen: *Self-organization and associative memory*. 3rd edition, Springer-Verlag, 1996.
- [65] Karciauskas, K., Krasauskas, R.: Rational rolling ball blending of natural quadrics, *Mathematical Modelling and Analysis*, 5 (2000) 97–107.

- [66] Krause, F. L., Fischer, A., Gross, N., Barhak, J.: Reconstruction of freeform objects with arbitrary topology using neural networks and subdivision techniques, *CIRP Annals - Manufacturing Technology* 52 (2003), 125–128.
- [67] Kruithof, N., Vegter, G.: Envelope Surfaces. *Proc. Annual Symposium on Computational Geometry*, 2006, 411–421.
- [68] Kunkli, R.: Localization of touching points for interpolation of discrete circles, *Annales Mathematicae et Informaticae*, 36 (2009) 103–110.
- [69] Kunkli, R., Hoffmann, M.: Skinning of circles and spheres, *Computer Aided Geometric Design*, 27 (2010), 611–621.
- [70] Kunkli, R., Papp, I., Hoffmann, M.: Isoptics of Bézier curves, *Computer Aided Geometric Design*, 30 (2013), 78–84.
- [71] Li, Y., Wang, G-Z.: Two kinds of B-basis of the algebraic hyperbolic space, *Journal of Zhejiang Univ. Sci.*, 6 (2005), 750–759.
- [72] Li, Y., Hoffmann, M., Wang, G-Z.: On the shape parameter and constrained modification of GB-spline curves, *Annales Mathematicae et Informaticae*, 34 (2007), 51–59.
- [73] Loe, K.F.: α B-spline: a linear singular blending B-spline, *The Visual Computer*, 12 (1996), 18–25.
- [74] Lü, Y., Wang, G-Z., Yang, G.: Uniform hyperbolic polynomial B-spline curves, *Computer Aided Geometric Design*, 19 (2002), 379–393.
- [75] Lyche, T.: Note on the Oslo algorithm, *Computer-Aided Design*, 20 (1988), 353–355.
- [76] Lyche, T., Mazure, M.L.: Total positivity and the existence of piecewise exponential B-splines. *Advances in Computational Mathematics*, 25 (2006), 105–133.
- [77] Ma, W., He, P.: B-spline surface local updating with unorganized points, *Computer-Aided Design*, 30 (1998), 853–862.
- [78] Mainar, E., Pena, J.M., Sanchez-Reyes, J.: Shape preserving alternatives to the rational Bézier model, *Computer Aided Geometric Design*, 18 (2001), 37–60.
- [79] Mainar, E., Pena, J.M.: A basis of C-Bézier splines with optimal properties, *Computer Aided Geometric Design*, 19 (2002), 291–295.
- [80] Mainar, E., Pena, J.M.: A general class of Bernstein-like bases. *Computers & Mathematics with Applications*, 53 (2007), 1686–1703.
- [81] Monge, G.: *Application de l’analyse à la géométrie*, Bachelier, Paris, 1850.
- [82] Morin, G., Warren, J., Weimer, H.: A subdivision scheme for surface of revolution. *Computer Aided Geometric Design* 18 (2001), 483–502.

- [83] Müller, E., Krames, J.L.: Vorlesungen über Darstellende Geometrie II: Die Zyklographie. Deuticke, Leipzig u. Wien, 1928.
- [84] Muirhead, R.F.: On the Number and Nature of the Solutions of the Apollonian Contact Problem. Proceedings of the Edinburgh Mathematical Society 14 (1896), 135–147.
- [85] Paluszny, M., Tovar, F.: Envelopes and tubular splines, Mathematics and Computers in Simulation, 79 (2009), 1971–1976.
- [86] Peternell, M., Odehnal, B., Sampoli, M.L.: On quadratic two-parameter families of spheres and their envelopes. Computer Aided Geometric Design, 25 (2008), 342–355.
- [87] Peternell, M., Pottmann, H.: A Laguerre geometric approach to rational offsets, Computer Aided Geometric Design, 15 (1998), 223–249.
- [88] Piegl, L.: Modifying the shape of rational B-splines. Part 1: curves, Computer–Aided Design, 21 (1989), 509–518.
- [89] Piegl, L.: On NURBS: a survey. IEEE Computer Graphics and Applications, 11 (1991), 55–71.
- [90] Piegl, L., Tiller, W.: The NURBS book, Springer–Verlag, 1995.
- [91] Piegl, L., Tiller, W.: Parametrization for surface fitting in reverse engineering, Computer-Aided Design, 33 (2001), 593–603.
- [92] Pottmann, H.: The geometry of Tchebycheffian splines, Computer Aided Geometric Design, 10 (1993), 181–210.
- [93] Pottmann, H., Wagner, M.: Helix splines as an example of affine tchebycheffian splines, Adv. Comput. Math., 2 (1994), 123–142.
- [94] Pratt, M.J.: Cyclides in computer aided geometric design. Computer Aided Geometric Design 7 (1990), 221–242.
- [95] Pratt, M.J.: Cyclides in computer aided geometric design II. Computer Aided Geometric Design 12 (1995), 131–152.
- [96] Rojas, R.: Neural Networks. A Systematic Introduction, Springer-Verlag, 1996.
- [97] Rossignac, J., Whited, B., Slabaugh, G., Fang, T., Unal, G.: Pearling: 3d interactive extraction of tubular structures from volumetric images, in: Proc. MICCAI Workshop: Interaction in Medical Image Analysis and Visualization, 2007.
- [98] Rossignac, J., Kim, J.J.: Helsweeper: Screw-sweeps of canal surfaces, Computer-Aided Design, 44 (2012), 113–122.
- [99] Róth, Á., Juhász, I., Schicho, J., Hoffmann, M.: A cyclic basis for closed curve and surface modeling, Computer Aided Geometric Design, 26 (2009), 528–546.

- [100] Sarfraz M.: Interactive curve modeling, Springer-Verlag, 2008.
- [101] Sánchez-Reyes, J.: A simple technique for NURBS shape modification, *IEEE Computer Graphics and Applications*, 17 (1997), 52–59.
- [102] Singh, K., Kokkevis, E.: Skinning characters using surface-oriented free-form deformations, in: *Proc. Graphics Interface 2000*, 2000, 35–42.
- [103] Slabaugh, G., Unal, G., Fang, T., Rossignac, J., Whited, B.: Variational skinning of an ordered set of discrete 2d balls, *Lecture Notes on Computer Science*, 4795 (2008), 450–461.
- [104] Slabaugh, G., Rossignac, J., Whited, B., Fang, T., Unal, G.: 3d ball skinning using pdes for generation of smooth tubular surfaces, *Computer-Aided Design*, 42 (2010), 18–26.
- [105] The Official Spore™ and Spore™ Creature Creator Site, www.spore.com (2014. december 16.).
- [106] Tai, C.L., Wang, G.J.: Interpolation with slackness and continuity control and convexity-preservation using singular blending, *Journal of Computational and Applied Mathematics*, 172 (2004), 337–361.
- [107] Tornai, R.: Measurement of visual smoothness of blending curves, *Annales Mathematicae et Informaticae*, 40 (2012), 155–160.
- [108] Troll, E., Hoffmann, M.: Geometric properties and constrained modification of trigonometric spline curves of Han, *Annales Mathematicae et Informaticae*, 37 (2010), 165–175.
- [109] Várady, T., Martin, R.R., Cox, J.: Reverse engineering of geometric models - an introduction, *Computer-Aided Design*, 29 (1997), 255–268.
- [110] Várady, L., Hoffmann, M., Kovács, E.: Improved Free-form Modelling of Scattered Data by Dynamic Neural Networks, *Journal for Geometry and Graphics*, 3 (1999), 177–181.
- [111] Wang, G., Chen, Q., Zhou, M.: NUAT B-spline curves, *Computer Aided Geometric Design*, 21 (2004), 193–205.
- [112] Wang, G., Fang, M.: Unified and extended form of three types of splines. *Journal of Computational and Applied Mathematics*, 216 (2008), 498–508.
- [113] Weiss, V., Andor, L., Renner, G., Várady, T.: Advanced surface fitting techniques, *CAGD* 19 (2002), 19–42.
- [114] Yang, Q., Wang, G.: Inflection points and singularities on C-curves, *Computer Aided Geometric Design* 21 (2004), 207–213.
- [115] Yin, M., Wang, Q.: B-spline curve of order 4 with two shape parameters, *Journal of Computational Information Systems*, 1 (2005), 609–614.

- [116] Yu, Y.: Surface reconstruction from unorganized points using self-organizing neural networks, Proc. of IEEE Visualization 99, 1999, 61–64.
- [117] Zhang, J.: C-curves: An extension of cubic curves, Computer Aided Geometric Design, 13 (1996), 199–217.
- [118] Zhang, J. W.: Two different forms of CB-splines, Computer Aided Geometric Design, 14 (1997), 31–41.
- [119] Zhang, J.: C-Bézier curves and surfaces. Graphical Models and Image Processing 61 (1999), 2–15.
- [120] Zhang, J. W., Krause, F.-L.: Extend cubic uniform B-splines by unified trigonometric and hyperbolic basis, Graphical Models, 67 (2005), 100–119.
- [121] Zhang, J. W., Krause, F.-L., Zhang, H.: Unifying C-curves and H-curves by extending the calculation to complex numbers, Computer Aided Geometric Design, 22 (2005), 865–883.
- [122] Zhou, P., Qian, W.-H.: Blending multiple parametric normal ringed surfaces using implicit functional splines and auxiliary spheres, Graphical Models, 73 (2011), 87–96.
- [123] Zhou, P., Qian, W.-H.: Gn-blending of multiple parametric normal ringed surfaces by adding implicit closings gn-continuous with the surfaces, Computers and Graphics, 36 (2012), 297–304.
- [124] Pixologic™ ZBrush Features, www.pixologic.com (2014. december 16.).