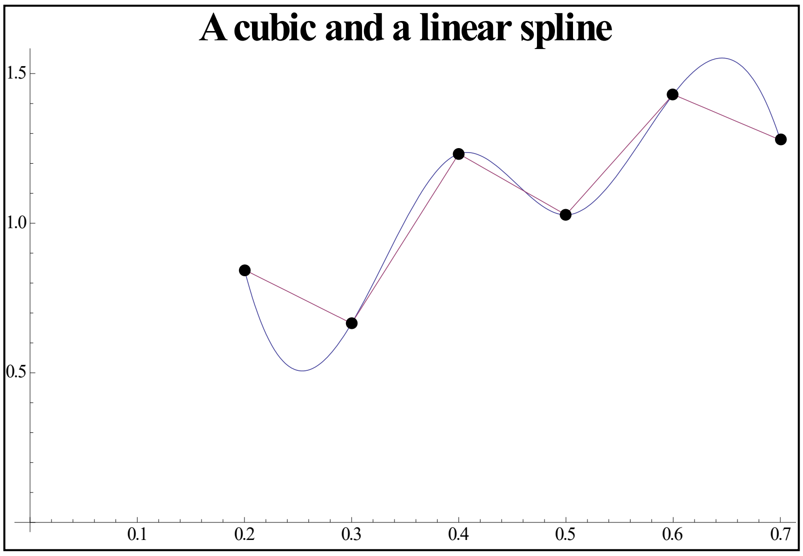
Spline-Interpolation

Outline of methods

1. One-dimensional Splining



When a (rather large) set of measurement points with arguments $x_0, x_1, \ldots, x_{n-1}, x_n$ have to be met and interpolated by a curve the method of collocation (= interpolation by a single polynomial of degree at most n) often runs into oscillation problems of the error towards the boundaries of the interval $\{\min x_i, \max x_i\}_{i=0,1,\ldots,n}$.

This problem is called Runge phenomenon. The concept of splines avoids this phenomenon by "patching" together many (n) polynomial curves of low degrees on the n segments ("patches") $(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)$. Of course, the low-degree polynomial curves meet one another at the measurement points with interior arguments $x_1, \ldots, x_{n-1}$ making the whole "patched" curve continuous (abbreviated by the symbol C^0 or saying it is a C^0-curve). But often more is required: The curve must fulfill so-called smoothness conditions at the interior points meaning that it has to be continuously differentiable up to a certain given order k (abbreviated by the symbol C^k or saying it is a C^k -curve).

A typical situation is the case where the maximum degree of the "patching" polynomials is d and the "patched" curve must be a C^{d-1}-curve ((d -1) times continuously differentiable). It is assumed that $n > d$ -1. Each polynomial has $d+1$ coefficients giving a total of $n(d+1)$ undetermined coefficients. On the other side we have the following sets of conditions:

# of conditions	Description
2 n	Each "patching" polynomial curve meets the two measurement points corresponding to the endpoints of its segment ("patch").
$(d$-1) (n-1)	The derivatives of order 1, 2, …, d-1 are continuous at the interior arguments $x_1, \ldots, x_{n-1}$.

Table 1.1: Conditions for a typical spline (it is assumed that $n > d$ -1)

This gives a total of $n(d-1) + 2n - (d-1) = n(d+1) - (d-1)$ conditions. Since this is smaller than the number of undetermined coefficients a set of (d -1) further conditions may be met at the arguments $x_0, x_1, \ldots, x_{n-1}, x_n$. This makes the concept of splines rather flexible.

A very important case for practical purposes is cubic splining: $d = 3$ and $n > d$ -1. In cubic splining there are 4n undetermined coefficients and 4n – 2 conditions at the interior arguments. Depending on the nature of the "two further conditions" to be met at (two of) the arguments we may distinguish several concepts of cubic splining.

1.1 Cubic spline

For $i = 0, 1, \ldots, n-1$ we define a cubic polynomial

$$\boxed{S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3} \tag{1.1}$$

belonging to the segment ("patch") $[x_i, x_{i+1}]$. Further we define $h_i = x_{i+1} - x_i = \Delta x_i$. By examining the conditions mentioned in Table 1.1 above we get the following system of equations.

1. Since the spline curve has to meet the measurement points $(x_i, y_i) \quad (i = 0,1,\ldots,n)$ we get:

$\boxed{a_i = y_i \quad (i = 0,\ldots,n-1)}$. The argument x_n has not been used yet.

2. Since the spline is a C^2-curve: $$\boxed{S_i''(x_i) = S_{i-1}''(x_i) \Rightarrow d_{i-1} = \frac{c_i - c_{i-1}}{3h_{i-1}} \quad (i = 1,\ldots,n-1)} \tag{1.2}$$

The coefficient d_{n-1} will be computed later by using special conditions at the boundary arguments x_0 and x_n, respectively.

3. Since the spline curve is continuous:

$$\boxed{S_i(x_i) = S_{i-1}(x_i) \Rightarrow b_{i-1} = \frac{a_i - a_{i-1}}{h_{i-1}} - \frac{2c_{i-1} + c_i}{3} h_{i-1} \quad (i = 1,\ldots,n-1)} \tag{1.3}$$

The coefficient b_{n-1} will be computed later by using the argument x_n and the coefficients d_{n-1} and c_{n-1}, respectively.

$$\boxed{b_{n-1} = \frac{y_n - a_{n-1}}{h_{n-1}} - c_{n-1}h_{n-1} - d_{n-1}h_{n-1}^{\ 2}} \tag{1.3'}$$

4. Since the spline is a C^1-curve:

$$\boxed{\begin{array}{l} S_i'(x_i) = S_{i-1}'(x_i) \Rightarrow b_i = b_{i-1} + 2c_{i-1}h_{i-1} + 3d_{i-1}h_{i-1}^{\ 2} \quad (i = 1,\ldots,n-1) \quad \Rightarrow \\[2ex] c_{i-1}h_{i-1} + 2(h_{i-1} + h_i)c_i + c_{i+1}h_i = 3\left(\dfrac{a_{i+1} - a_i}{h_i} - \dfrac{a_i - a_{i-1}}{h_{i-1}}\right) \qquad (i = 1,\ldots,n-2) \end{array}} \tag{1.4}$$

Here the equation (1.3) has been used to replace b_{i-1} and $b_i \quad (i = 1,\ldots n-2)$.

The last n-2 equations (1.4) form a linear system with a tri-diagonal band matrix for the n coefficients $c_0, c_1, \ldots, c_{n-1}$. The matrix has dimensions n-2 by n.

$$\begin{pmatrix} h_0 & 2(h_0+h_1) & h_1 & & & \\ & h_1 & 2(h_1+h_2) & h_2 & & \\ & & h_2 & 2(h_2+h_3) & h_3 & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} 3\left(\frac{y_{i+1}-y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right) \end{pmatrix}_{i=1,\dots,n-2}$$

Table 1.2: Incomplete tri-diagonal system of n-2 equations for n coefficients for a cubic spline

1.2 Natural cubic spline

The natural cubic spline S fulfills the boundary conditions $\boxed{S''(x_0) = S''(x_n) = 0}$. These conditions have a physical ("natural") interpretation: The natural cubic spline minimizes the mean total curvature $\int_{x_0}^{x_n} |f''(x)|^2 dx$ among all C^2-functions in the interval $[x_0, x_n]$ that meet all the measurement points (here it is assumed that x_0 and x_n are minimal and maximal, respectively). This fact is called Holladay-Theorem.

The development of the tri-diagonal system in Table 1.2 has not yet considered the additional (two) conditions at the boundary arguments x_0 and x_n. In the case of natural splining this means that

$$\begin{aligned} &S_0''(x_0) = 2c_0 = 0 \Rightarrow c_0 = 0 \\ &S_{n-1}''(x_n) = 2c_{n-1} + 6d_{n-1}h_{n-1} = 0 \Rightarrow d_{n-1} = -\frac{c_{n-1}}{3h_{n-1}} \end{aligned}$$

Now we use the first equation of (1.4) for $i = n$-1 and (1.3') as well as (1.2) and (1.3) to conclude that

$$\begin{aligned} &b_{n-1} = \frac{y_n - a_{n-1}}{h_{n-1}} - c_{n-1}h_{n-1} - d_{n-1}h_{n-1}^2 = \frac{y_n - y_{n-1}}{h_{n-1}} - \frac{2}{3}c_{n-1}h_{n-1} \\ &b_{n-1} = b_{n-2} + 2c_{n-2}h_{n-2} + 3d_{n-2}h_{n-2}^2 = \frac{a_{n-1} - a_{n-2}}{h_{n-2}} - \frac{2c_{n-2} + c_{n-1}}{3}h_{n-2} + 2c_{n-2}h_{n-2} + (c_{n-1} - c_{n-2})h_{n-2} \end{aligned}$$

Subtraction of these two equalities yields a further condition on the c-coefficients:

$$\boxed{c_{n-2}h_{n-2} + 2c_{n-1}(h_{n-2} + h_{n-1}) = 3\left(\frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}}\right)}$$

Now the tri-diagonal system of equations of Table 1.2 takes the following regular form (n-1 equations for the n-1 coefficients $c_1, \dots, c_{n-1}$):

$$\begin{pmatrix} 2(h_0+h_1) & h_1 & & & & \\ h_1 & 2(h_1+h_2) & h_2 & & & \\ & h_2 & 2(h_2+h_3) & h_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} \\ & & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} \\ 3\left(\frac{y_{i+1}-y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right) \\ \\ \end{pmatrix}_{i=1,\dots,n-1}$$

Table 1.3: Regular tri-diagonal system of n-1 equations for n-1 coefficients

From the c-coefficients we compute the d-coefficients by (1.2) (d_{n-1} by $-\frac{c_{n-1}}{3h_{n-1}}$) and the b-coefficients by (1.3) or (1.3'), respectively.

1.3 Complete cubic spline ("clamped")

In complete or clamped cubic splining there are <u>two first order derivative conditions on the boundary arguments</u>:

$$S'(x_0) = y'_0 \quad S'(x_n) = y'_n$$

with given values y'_0 and y'_n. The formulas (1.1) to (1.4) and the tri-diagonal linear system of equations in Table 1.2 hold for all cubic splines. The first order boundary conditions yield the following equations:

$$\begin{aligned} &S'(x_0) = y'_0 \Rightarrow b_0 = y'_0 \\ &S'(x_n) = y'_n \Rightarrow b_{n-1} + 2c_{n-1}h_{n-1} + 3d_{n-1}h_{n-1}^{\;2} = y'_n \end{aligned} \qquad \textbf{(1.5)}$$

We proceed as in the case of the natural spline and use equations (1.3), (1.3') and (1.4) to derive further equations for the c-coefficients.

From (1.3) we get that

$$y'_0 = b_0 = \frac{a_1 - a_0}{h_0} - \frac{2c_0 + c_1}{3} h_0 \Rightarrow 2c_0 h_0 + c_1 h_0 = 3\left(\frac{a_1 - a_0}{h_0} - y'_0\right) \qquad \textbf{(1.6)}$$

From (1.3'), (1.4) and (1.5) we conclude the following three equations:

$$b_{n-1} = \frac{y_n - a_{n-1}}{h_{n-1}} - c_{n-1}h_{n-1} - d_{n-1}h_{n-1}^2$$

$$b_{n-1} = b_{n-2} + 2c_{n-2}h_{n-2} + 3d_{n-2}h_{n-2}^2 = \frac{a_{n-1} - a_{n-2}}{h_{n-2}} - \frac{2c_{n-2} + c_{n-1}}{3}h_{n-2} + 2c_{n-2}h_{n-2} + (c_{n-1} - c_{n-2})h_{n-2}$$

$$b_{n-1} = y'_n - 2c_{n-1}h_{n-1} - 3d_{n-1}h_{n-1}^2$$

Multiplying the first equation by 3 and subtracting the third equation eliminates d_{n-1}. The substitution of b_{n-1} with the term in the second equation then yields a further condition on the c-coefficients:

$$2b_{n-1} = 3\frac{y_n - a_{n-1}}{h_{n-1}} - c_{n-1}h_{n-1} - y'_n \Rightarrow$$

$$2\left(\frac{a_{n-1} - a_{n-2}}{h_{n-2}} - \frac{2c_{n-2} + c_{n-1}}{3}h_{n-2} + 2c_{n-2}h_{n-2} + (c_{n-1} - c_{n-2})h_{n-2}\right) = 3\frac{y_n - a_{n-1}}{h_{n-1}} - c_{n-1}h_{n-1} - y'_n \Rightarrow$$

$$6\left(\frac{a_{n-1} - a_{n-2}}{h_{n-2}} - \frac{2c_{n-2} + c_{n-1}}{3}h_{n-2} + 2c_{n-2}h_{n-2} + (c_{n-1} - c_{n-2})h_{n-2}\right) = 9\frac{y_n - a_{n-1}}{h_{n-1}} - 3c_{n-1}h_{n-1} - 3y'_n \Rightarrow$$

$$-4c_{n-2}h_{n-2} - 2c_{n-1}h_{n-2} + 12c_{n-2}h_{n-2} + 6c_{n-1}h_{n-2} - 6c_{n-2}h_{n-2} =$$
$$9\frac{y_n - a_{n-1}}{h_{n-1}} - 6\frac{a_{n-1} - a_{n-2}}{h_{n-2}} - 3c_{n-1}h_{n-1} - 3y'_n \Rightarrow$$

$$2c_{n-2}h_{n-2} + 4c_{n-1}h_{n-2} + 3c_{n-1}h_{n-1} = 9\frac{y_n - a_{n-1}}{h_{n-1}} - 6\frac{a_{n-1} - a_{n-2}}{h_{n-2}} - 3y'_n \qquad \textbf{(1.7)}$$

The system in Table 1.2 combined with (1.6) and (1.7) forms a complete linear system for the c-coefficients:

$$\begin{pmatrix} 2h_0 & h_0 & & & & & \\ h_0 & 2(h_0+h_1) & h_1 & & & & \\ & h_1 & 2(h_1+h_2) & h_2 & & & \\ & & h_2 & 2(h_2+h_3) & h_3 & & \\ & & \ddots & \ddots & \ddots & & \\ & & & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} \\ & & & & 2h_{n-2} & 4h_{n-2}+3h_{n-1} \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} 3\left(\frac{a_1 - a_0}{h_0} - y'_0\right) \\ \vdots \\ \left(3\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right)\right)_{i=1,\dots n-2} \\ \vdots \\ 9\frac{y_n - a_{n-1}}{h_{n-1}} - 6\frac{a_{n-1} - a_{n-2}}{h_{n-2}} - 3y'_n \end{pmatrix}$$

Table 1.4: Regular complete tri-diagonal system of n equations for n coefficients for a clamped cubic spline

From (1.2) we get $d_{0,.}\dots,d_{n-2}$, from (1.3) we get $b_{0,.}\dots,b_{n-2}$, from (1.4) then we get b_{n-1} and (finally) from (1.3') we compute d_{n-1}.

1.4 Other cubic spline methods

Hermite cubic spline: C^1-curve patched from cubic polynomials with given first order derivatives at the arguments $x_0, x_1, \dots, x_{n-1}, x_n$ (n+1 conditions). This gives a total amount of exactly 4n conditions. An elegant solution method is based on a linear combination of Hermite basis functions (cubic polynomials) on the special interval [0,1] ("single special patch"). By linear transformations from the interval [0,1] to the segments $[x_i, x_{i+1}]$ ($i = 0, 1, \dots, n-1$). it is rather easy to compute in a piecewise manner a compound solution fulfilling the conditions on the derivatives at the arguments $x_0, x_1, \dots, x_{n-1}, x_n$.

The Hermite cubic spline S allows for an application of the <u>osculation error formula</u>:

For any segment $[x_i, x_{i+1}]$ we have an osculation error (!), thus

$$y(x) - S(x) = \frac{y^{(4)}(\xi)}{4!}(x - x_i)^2 (x - x_{i+1})^2 \qquad x, \xi \in (x_i, x_{i+1})$$

From this it follows by maximization that

$$|\, y(x) - S(x) \,| = \max \frac{|\, y^{(4)}(x) \,|}{4!} \underbrace{\max |\, (x - x_i)^2 (x - x_{i+1})^2 \,|}_{\le \frac{1}{16} h_i^4} \qquad x \in [x_i, x_{i+1}]$$

and thus

$$\boxed{\begin{array}{l} |\, y(x) - S(x) \,| \le \max \dfrac{|\, y^{(4)}(x) \,|}{4!} \dfrac{H^4}{16} = \max |\, y^{(4)}(x) \,| \dfrac{H^4}{384} \\ x \in [x_0, x_n], \quad H = \max\limits_{i=0,\dots,n-1} h_i \end{array}} \qquad \textbf{(1.8)}$$

Similar error formulas for other spline methods exist but their derivations generally are much more difficult[1].

Periodic spline: C^2-curve S patched from cubic polynomials with periodic properties (provided that $y_0 = y_n \Rightarrow S(x_0) = S(x_n)$): The two periodic standard conditions are $S'(x_0) = S'(x_n)$ or $S''(x_0) = S''(x_n)$, alternatively. This gives a total amount of exactly 4n conditions.

---

[1] Cf. C.A. Hall & W.W. Meyer: *Optimal error bounds for cubic spline interpolation*. Journal Approx. Theory 16, p. 105-122 (1976). This paper gives a survey on error bounds and a main theorem, as well: If y is a C^4-function and S a cubic C^2-spline interpolant that coincides with y' or, alternatively, with y'' at the boundary arguments $\{x_0, x_n\}$ then

$$|\, y(x) - S(x) \,| \le \max \frac{|\, y^{(4)}(x) \,|}{4!} \frac{5H^4}{16} = \max |\, y^{(4)}(x) \,| \frac{5}{384} H^4$$

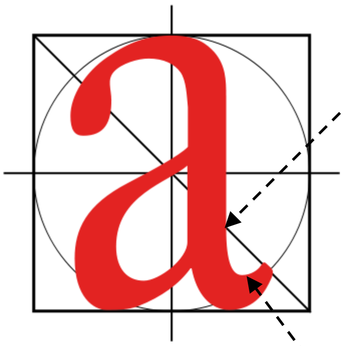
$$|\, y'(x) - S'(x) \,| \le \max \frac{|\, y^{(4)}(x) \,|}{4!} H^3 = \max |\, y^{(4)}(x) \,| \frac{1}{24} H^3$$

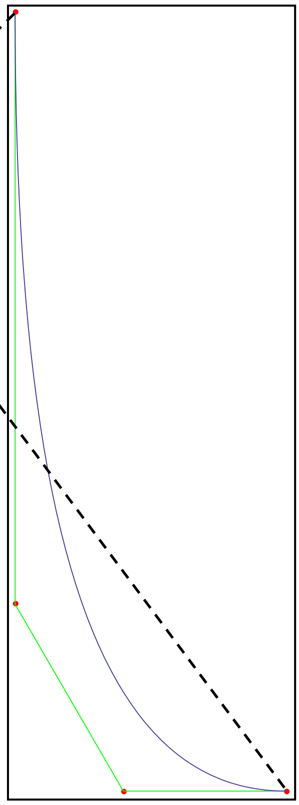
$$|\, y''(x) - S''(x) \,| \le \max |\, y^{(4)}(x) \,| \frac{3}{8} H^2 \qquad x \in [x_0, x_n], \quad H = \max_{i=0,\dots,n-1} h_i$$

"Not – a – Knot" – Spline: C^2-curve S patched from cubic polynomials when the knots x_1 and x_{n-1} are ignored and not considered as interior arguments. The number of patches is reduced to n-2 and the number of interior points to n-3. This yields 4(n-2) = 4n – 8 for the number of undetermined coefficients and 2(n-2) + (n-3)(d-1) = 4n – 10 for the number of conditions. Again there is an excess of 2 undetermined coefficients.

As the Hermite cubic spline described above also the Bézier-Spline or the Basis-Spline are composed from a cleverly devised set of basis functions. These are discussed in the next section.

2. Bernstein-Bézier Splines (B-B-Splines)





The methods described in the previous section led to rather efficient implementations because of the tri-diagonal band structure of the linear systems in Table 1.2 and 1.3. On the other side, the consideration of special, uncomplicated conditions like "natural" or "clamped" at the boundary arguments was not easily and generically transformable into mathematical formulaes or algorithms despite the fact that a large portion of the linear system of a cubic spline was determined only by the interior arguments (cf. Table 1.2). Moreover, the lack of genericity increases when solving the problem of higher-order splines with higher degrees d and higher-order conditions on the smoothness at interior points (cf. Table 1.1). Another disadvantage of the methods in section 1 is that they refer to a set of measurement points and thus to a set of points on a function graph. This is very restrictive.

Improvements with respect to these disadvantages are necessary and – luckily – corresponding methods exist. Examples are Hermite interpolation, Bernstein-Bézier splines (B-B-splines) and Basis splines (B-splines). These methods have in common that they are based on cleverly devised sets of basis functions called Hermite basis polynomials, Bernstein polynomials or just Basis functions (in the case of Basis splines).

2. 1 Bernstein-Polynomials

The set of basis functions for Bézier splines consists of the Bernstein polynomials:

$$B_{in}(t) = \binom{n}{i}(1-t)^{n-i}t^i \qquad t \in [0,1] \quad (i = 0,1,\cdots,n) \tag{2.1}$$

This elementary polynomials refer to the interval [0, 1]. By an affine transformation of [0, 1] into the interval [a, b] we immediately get the transformed Bernstein polynomials which refer to [a, b]:

$$B_{in}(u,a,b) = B_{in}\left(\frac{u-a}{b-a}\right) = \frac{1}{(b-a)^n}\binom{n}{i}(b-u)^{n-i}(u-a)^i \qquad u \in [a,b] \quad (i = 0,1,\cdots,n) \qquad \textbf{(2.1')}$$

The following properties of Bernstein polynomials are obvious or their proofs elementary:

Property 2.1: Some elementary facts on Bernstein polynomials.

a) The Bernstein polynomials of order n form a linear basis for the polynomials of order n.

b) $B_{in}(t)$ has a exactly one maximum at $t = \frac{i}{n}$.

c) $B_{in}(t)$ has a zero at 0 of order i and a zero at 1 of order n-i.

d) Symmetry: $B_{in}(t) = B_{n-i,n}(1-t)$

e) The Bernstein polynomials are <u>bounded</u> in [0,1]: $0 \le B_{in}(t) \le 1 \quad t \in [0,1]$

f) The Bernstein polynomials are a <u>partition of unity</u>: $\sum_{i=0}^{n} B_{in}(t) = 1$.

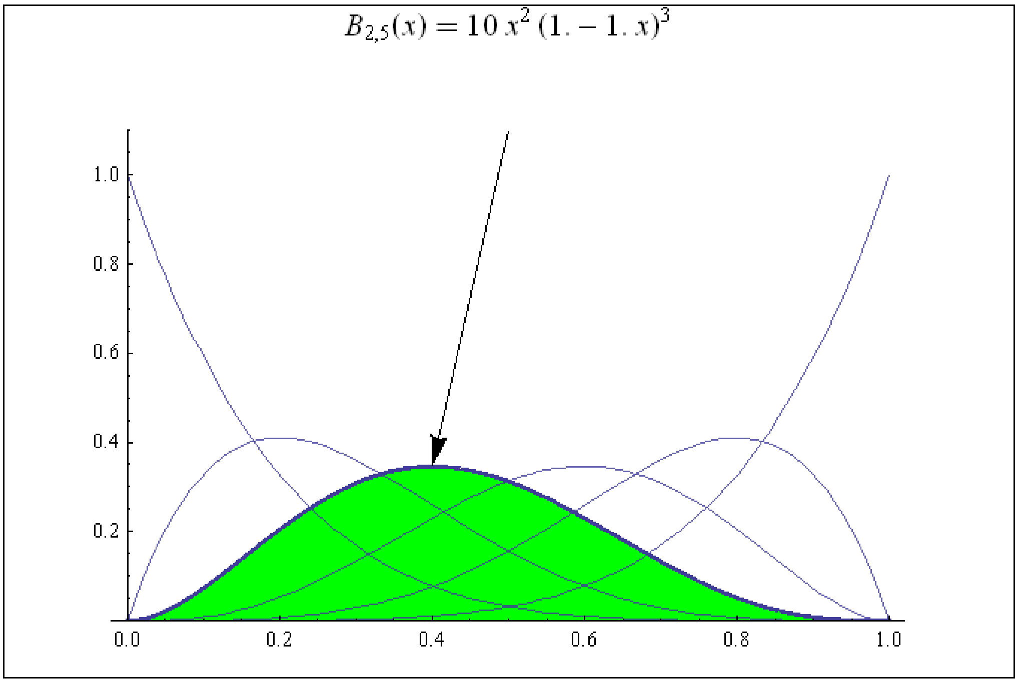


Figure 2.1: The six Bernstein polynomials of order 5 with $B_{2,5}$ highlighted.

For many formulas it is technically advantageous to define $B_{i,n}(t)$ identically 0 whenever $i < 0$ or $i > n$.

Property 2.2: <u>Recurrence relations</u> of the Bernstein polynomials for order $n \ge 1$:

$$B_{in}(t) = tB_{i-1,n-1}(t) + (1-t)B_{i,n-1}(t) \quad (i = 1, \cdots, n-1)$$
$$B_{0n}(t) = (1-t)B_{0,n-1}(t)$$
$$B_{nn}(t) = tB_{n-1,n-1}(t)$$

This says how $B_{in}(t)$ can be computed from lower order Bernstein polynomials. The proof of this is based on the recurrence relation $\binom{n-1}{i-1} + \binom{n-1}{i} = \binom{n}{i} \quad (i = 1, \cdots, n-1)$ for binomial coefficients.

The recursion relation is crucial for the efficient evaluation of Bernstein polynomials. This will lead to <u>Casteljau's algorithm</u> for the evaluation of points lying on Bézier curves (cf. section below).

Property 2.3: <u>Differentiation</u> of the Bernstein polynomials:

a) $$\frac{d}{dt} B_{in}(t) = n(B_{i-1,n-1}(t) - B_{i,n-1}(t)) = -n\,\Delta B_{i-1,n-1}(t)$$

b) $$\frac{d}{dt^2} B_{in}(t) = n(n-1)(B_{i-2,n-2}(t) - 2B_{i-1,n-2}(t) + B_{i,n-2}(t)) = n(n-1)\,\Delta^2 B_{i-2,n-2}(t)$$

c) $$\frac{d}{dt^k} B_{in}(t) = (-1)^k n(n-1)(n-2)\cdots(n-k+1)\Delta^k B_{i-k,n-k}(t)$$

The symbol Δ^k denotes the forward difference operator k-times iterated.

(Proof: a) is an elementary consequence of the product differentiation rule. Then b) and c) are iterated applications of a) □)

2.2 Simple Bézier curves

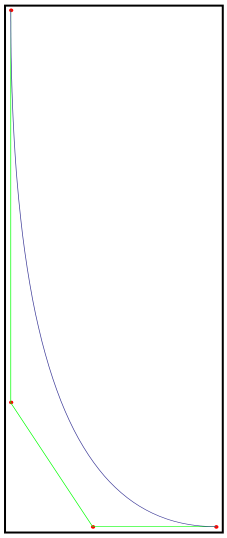
The Bernstein polynomials and their properties (cf. section above) allow for an elegant construction of spline curves that are controlled by a set of d-dimensional control points (vectors) $\vec{P}_0, \vec{P}_1, \cdots, \vec{P}_n \quad (n \geq 2)$ in R^d. These spline curves of order n are named Bernstein-Bézier curves, B-B-splines or Bézier curves and their definition is simple and elegant:

$$\vec{r}(t) = \sum_{i=0}^{n} \vec{P}_i B_{in}(t) \quad t \in [0,1] \tag{2.2}$$

The Bernstein-Bézier polygon is generated by connecting $\vec{P}_i, \vec{P}_{i+1}$ with straight lines (i = 0, 1, …, n-1).

Example 2.1: The figure on the right shows a 2-dimensional cubic Bézier spline with four control points in R^2 and corresponding Bernstein-Bézier polygon.

The figure represents a part of an image encoded in .svg (scalable vector graphics). The four control points are:



$$\vec{P}_0 = \begin{pmatrix} 221.852 \\ -222.566 \end{pmatrix}, \quad \vec{P}_1 = \begin{pmatrix} 221.852 \\ -257.911 \end{pmatrix},$$

$$\vec{P}_2 = \begin{pmatrix} 228.191 \\ -269.071 \end{pmatrix}, \quad \vec{P}_3 = \begin{pmatrix} 237.743 \\ -269.071 \end{pmatrix}$$

The vector formula for the Bézier curve is as follows:

$$\vec{r}(t) = \sum_{i=0}^{n} \vec{P}_i B_{in}(t) =$$

$$\begin{pmatrix} 221.852(1-t)^3 + 3\cdot 221.852t(1-t)^2 + 3\cdot 228.191t^2(1-t)^1 + 237.743t^3 \\ -222.566(1-t)^3 + 3\cdot(-257.911)t(1-t)^2 + 3\cdot(-269.071)t^2(1-t)^1 - 269.071t^3 \end{pmatrix} \quad t \in [0,1]$$

Property 2.4: Convex hull property of Bézier curves

The Bézier curve lies in the convex hull of its control points. The convex hull is defined as the set of all convex combinations $\{x \in R^d \mid x = \sum_{i=0}^{n} \lambda_i \vec{P}_i \wedge \sum_{i=0}^{n} \lambda_i = 1 \wedge 0 \le \lambda_i \le 1\}$.

(Proof: This follows immediately from (2.2) and Property 2.1e and f (partition of unity) ∎)

Property 2.5: Tangential directions and differential properties

For the Bézier curve (2.2) we have the following identities:

a) $\vec{r}(0) = \vec{P}_0, \quad \vec{r}(1) = \vec{P}_n$

b) $\vec{r}\,'(0) = n(\vec{P}_1 - \vec{P}_0), \quad \vec{r}\,'(1) = n(\vec{P}_n - \vec{P}_{n-1})$

c) $\vec{r}\,''(0) = n(n-1)(\vec{P}_2 - 2\vec{P}_1 + \vec{P}_0), \quad \vec{r}\,''(1) = n(n-1)(\vec{P}_n - 2\vec{P}_{n-1} + \vec{P}_{n-2})$

d) $\frac{d}{dt^k}\vec{r}(0) = n(n-1)\cdots(n-k+1)\Delta^k \vec{P}_0, \quad \frac{d}{dt^k}\vec{r}(1) = n(n-1)\cdots(n-k+1)\Delta^k \vec{P}_{n-k}$

(Proof: a) follows from (2.2). Part b) follows from (2.2) and Property 2.3:

$$\vec{r}\,'(t) = \sum_{i=0}^{n} \vec{P}_i B_{in}\,'(t) = n \cdot \sum_{i=0}^{n} \vec{P}_i (B_{i-1,n-1}(t) - B_{i,n-1}(t)).$$

Setting t = 0: The expressions $(B_{i-1,n-1}(t) - B_{i,n-1}(t))$ are 0 except for i = 0 and i = 1, where they evaluate to (0 – 1) and (1 – 0), respectively. Therefore, the sum simplifies to $(\vec{P}_1 - \vec{P}_0)$.

Setting t = 1: The expressions $(B_{i-1,n-1}(t) - B_{i,n-1}(t))$ are 0 except for i = n and i = n–1 , where they evaluate to (1 – 0) and (0 – 1), respectively. This time, the sum simplifies to $(\vec{P}_n - \vec{P}_{n-1})$. The proposition in c) is proved similarly. In general we have the differential identity

$$\frac{d}{dt^k}\vec{r}(0) = n(n-1)\cdots(n-k+1)\Delta^k \vec{P}_0, \quad \frac{d}{dt^k}\vec{r}(1) = n(n-1)\cdots(n-k+1)\Delta^k \vec{P}_{n-k}$$ ∎)

These identitites will be important to examine smoothness when several Bézier curves have to be composed (cf. section below).

2.3 Casteljau recurrence

The Castelau recurrence is similar in its idea to the Neville-Aitken recurrence for polynomial interpolation. It allows to compute a point on a Bézier curve corresponding to control points $\vec{P}_0, \vec{P}_1, \cdots, \vec{P}_n \quad (n \ge 2)$ as a convex linear combination of points of lower order Bézier curves.

$$\vec{r}_{\vec{P}_0,\vec{P}_1,\cdots,\vec{P}_n}(t) = (1-t)\vec{r}_{\vec{P}_0,\vec{P}_1,\cdots,\vec{P}_{n-1}}(t) + t \cdot \vec{r}_{\vec{P}_1,\vec{P}_2,\cdots,\vec{P}_n}(t) \qquad t \in [0,1] \tag{2.3}$$

(Proof: This follows from the recursion identities of the Bernstein polynomials in Property 2.2:

$$\vec{r}_{\vec{P}_0,\vec{P}_1,\cdots,\vec{P}_n}(t) = \sum_{i=0}^{n} \vec{P}_i B_{in}(t) = \sum_{i=0}^{n} \vec{P}_i (t \cdot B_{i-1,n-1}(t) + (1-t) B_{i,n-1}(t)) =$$

$$t \cdot \sum_{i=1}^{n} \vec{P}_i B_{i-1,n-1}(t) + (1-t) \cdot \sum_{i=0}^{n-1} \vec{P}_i B_{i,n-1}(t) = t \cdot \vec{r}_{\vec{P}_1,\vec{P}_2,\cdots,\vec{P}_n}(t) + (1-t) \vec{r}_{\vec{P}_0,\vec{P}_1,\cdots,\vec{P}_{n-1}}(t) \qquad t \in [0,1]$$

□)

The recurrence relation in (2.3) allows for an efficient computation of $\vec{r}_{\vec{P}_0,\vec{P}_1,\cdots,\vec{P}_n}(t)$ for any $t \in [0,1]$. Below, this is illustrated graphically for the case $n = 3$ with four control points:

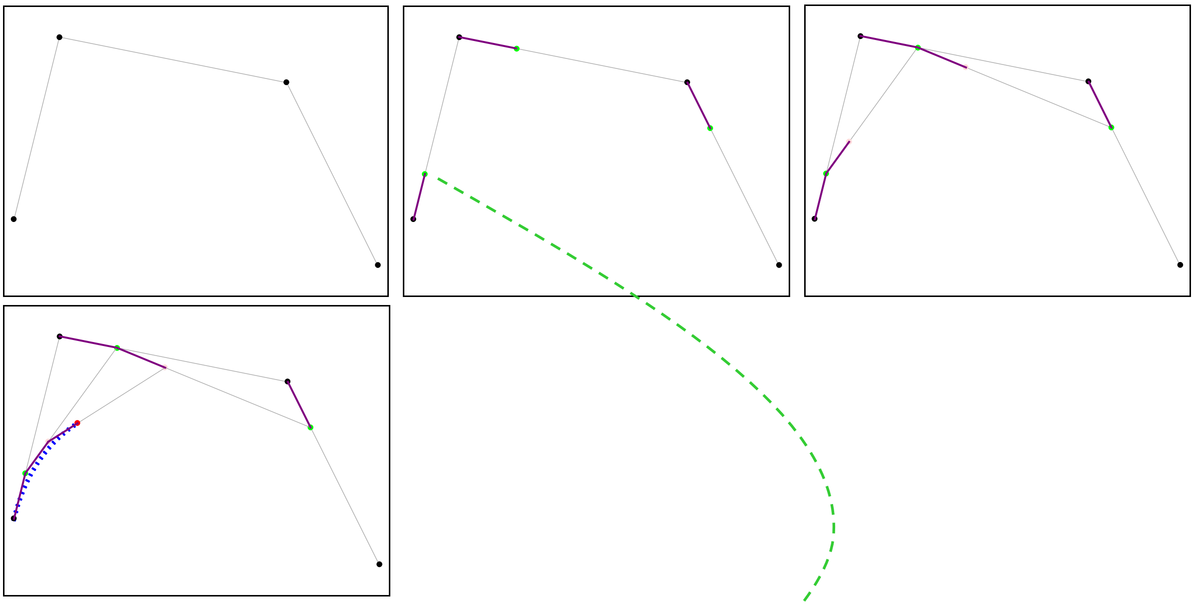


Figure 2.2: Graphical illustration of the Casteljau recurrence. Here $t = 0.25$. The segments are subdivided with $t = 0.25$.

In the first step four 0-order Bézier polynomials are computed (black control points), then these are used to compute first order values using (2.3), then the first order values are used to compute the second order values etc. Normally, this is written in a tableau; each picture in Fig. 2.2 corresponds to a column in the tableau. The entries in the tableau are vectors:

$\vec{r}_{\vec{P}_0}(t)$			
$\vec{r}_{\vec{P}_1}(t)$	$(1-t)\vec{r}_{\vec{P}_0}(t) + t \cdot \vec{r}_{\vec{P}_1}(t) = \vec{r}_{\vec{P}_0,\vec{P}_1}(t)$		
$\vec{r}_{\vec{P}_2}(t)$	$(1-t)\vec{r}_{\vec{P}_1}(t) + t \cdot \vec{r}_{\vec{P}_2}(t) = \vec{r}_{\vec{P}_1,\vec{P}_2}(t)$	$(1-t)\vec{r}_{\vec{P}_0,\vec{P}_1}(t) + t \cdot \vec{r}_{\vec{P}_1,\vec{P}_2}(t) = \vec{r}_{\vec{P}_0,\vec{P}_1,\vec{P}_2}(t)$	
$\vec{r}_{\vec{P}_3}(t)$	$(1-t)\vec{r}_{\vec{P}_2}(t) + t \cdot \vec{r}_{\vec{P}_3}(t) = \vec{r}_{\vec{P}_2,\vec{P}_3}(t)$	$(1-t)\vec{r}_{\vec{P}_1,\vec{P}_2}(t) + t \cdot \vec{r}_{\vec{P}_2,\vec{P}_3}(t) = \vec{r}_{\vec{P}_1,\vec{P}_2,\vec{P}_3}(t)$	$\underbrace{(1-t)\vec{r}_{\vec{P}_0,\vec{P}_1,\vec{P}_2}(t) + t \cdot \vec{r}_{\vec{P}_1,\vec{P}_2,\vec{P}_3}(t)}_{\vec{r}_{\vec{P}_0,\vec{P}_1,\vec{P}_2,\vec{P}_3}(t)}$

Table 2.1: The Casteljau tableau to compute a specific point $\vec{r}_{\vec{P}_0,\vec{P}_1,\vec{P}_2,\vec{P}_3}(t)$ on the Bézier curve.

2.4 Composite Bézier curves

In many applications (especially in design applications like font design for example) it is important and common to compose simple Bézier curves.

The figure on the right shows the composition of three simple cubic Bézier curves as a part of the font design for the letter . The simple Bézier curves <u>meet at common control points</u>. This is a continuity condition (C^0). Often higher order (smoothness) conditions are required as common tangent lines or even C^1, C^2 etc. in the common control points.

Property 2.5 will help to find the appropriate conditions for different <u>orders of smoothness</u>. The next theorems, e.g., are immediate consequences of Property 2.5.

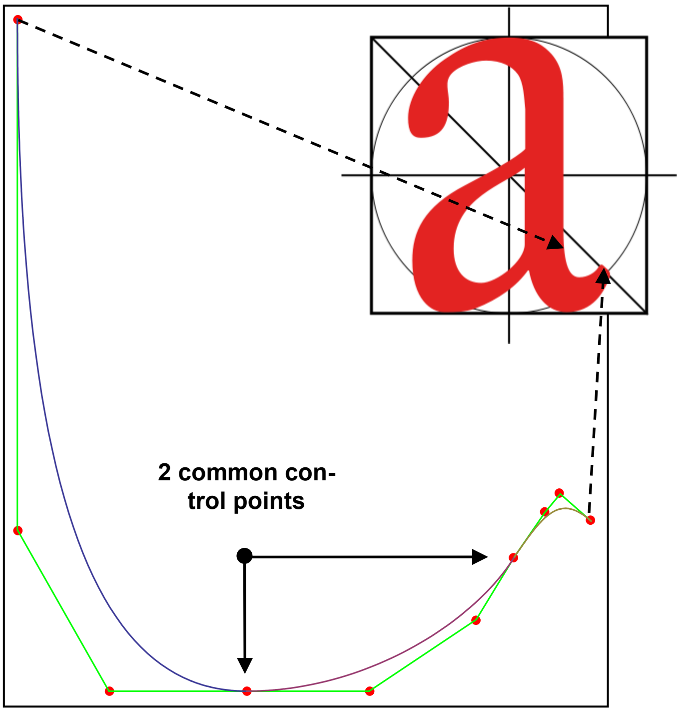


Figure 2.3: Three simple Bézier curves composed with two common control points

Theorem 2.1: C^1-smoothness condition

If $\vec{r}_P(t) = \sum_{i=0}^{n} \vec{P}_i B_{in}(t) \quad t \in [0,1]$ is a Bézier curve of order $n \geq 1$ with respect to the control points $\vec{P}_0, \vec{P}_1, \cdots, \vec{P}_n$ and $\vec{r}_Q(t) = \sum_{i=0}^{m} \vec{Q}_i B_{im}(t) \quad t \in [0,1]$ a Bézier curve of order $m \geq 1$ for the control points $\vec{Q}_0, \vec{Q}_1, \cdots, \vec{Q}_m$ with common control point $\vec{P}_n = \vec{Q}_0$ then the composite Bézier curve is continuously differentiable at the common control point if and only if $\vec{r}_P{}'(1) = \boxed{n(\vec{P}_n - \vec{P}_{n-1}) = m(\vec{Q}_1 - \vec{Q}_0)} = \vec{r}_Q{}'(0)$.

C^1-smoothness means that the curves have a common tangent vector (common in direction and length) at the common control point. The condition $\boxed{n(\vec{P}_n - \vec{P}_{n-1}) = m(\vec{Q}_1 - \vec{Q}_0) \Leftrightarrow n\Delta\vec{P}_{n-1} = m\Delta\vec{Q}_0}$ implies that the three points $\vec{P}_{n-1}, \vec{P}_n, \vec{Q}_1$ must lie on a straight line.

Theorem 2.2: C^2-smoothness condition

If $\vec{r}_P(t) = \sum_{i=0}^{n} \vec{P}_i B_{in}(t) \quad t \in [0,1]$ is a Bézier curve of order $n \geq 2$ with respect to the control points $\vec{P}_0, \vec{P}_1, \cdots, \vec{P}_n$ and $\vec{r}_Q(t) = \sum_{i=0}^{m} \vec{Q}_i B_{im}(t) \quad t \in [0,1]$ a Bézier curve of order $m \geq 2$ for the control points $\vec{Q}_0, \vec{Q}_1, \cdots, \vec{Q}_m$ with common control point $\vec{P}_n = \vec{Q}_0$ then the composite Bézier curve is twice continuously differentiable at the common control point if and only if

$$\vec{r}_P{}'(1) = \boxed{n(\vec{P}_n - \vec{P}_{n-1}) = m(\vec{Q}_1 - \vec{Q}_0)} = \vec{r}_Q{}'(0)$$

and $\vec{r}_P''(1) = \boxed{n(n-1)(\vec{P}_n - 2\vec{P}_{n-1} + \vec{P}_{n-2}) = m(m-1)(\vec{Q}_2 - 2\vec{Q}_1 + \vec{Q}_0)} = \vec{r}_Q''(0)$.

The condition $\boxed{n(n-1)(\vec{P}_n - 2\vec{P}_{n-1} + \vec{P}_{n-2}) = m(m-1)(\vec{Q}_2 - 2\vec{Q}_1 + \vec{Q}_0)}$ implies that the second difference vectors $\vec{P}_n - \vec{P}_{n-1} - (\vec{P}_{n-1} - \vec{P}_{n-2})$ and $\vec{Q}_2 - \vec{Q}_1 - (\vec{Q}_1 - \vec{Q}_0)$ are collinear.

Example 2.2: In Figure 2.3 we have $n = m = 3$ and the following sets of control points for the two first spline curves:

$$\vec{P}_0 = \begin{pmatrix} 221.852 \\ -222.566 \end{pmatrix}, \quad \vec{P}_1 = \begin{pmatrix} 221.852 \\ -257.911 \end{pmatrix}, \quad \vec{P}_2 = \begin{pmatrix} 228.191 \\ -269.071 \end{pmatrix}, \quad \vec{P}_3 = \begin{pmatrix} 237.743 \\ -269.071 \end{pmatrix}$$

$$\vec{Q}_0 = \begin{pmatrix} 237.743 \\ -269.071 \end{pmatrix}, \quad \vec{Q}_1 = \begin{pmatrix} 246.216 \\ -269.071 \end{pmatrix}, \quad \vec{Q}_2 = \begin{pmatrix} 253.634 \\ -264.123 \end{pmatrix}, \quad \vec{Q}_3 = \begin{pmatrix} 256.285 \\ -259.777 \end{pmatrix}$$

The condition of Theorem 2.1 reads $3(\vec{P}_3 - \vec{P}_2) = 3(\vec{Q}_1 - \vec{Q}_0) \Leftrightarrow \begin{pmatrix} 9.552 \\ 0 \end{pmatrix} = \begin{pmatrix} 8.473 \\ 0 \end{pmatrix}$ and this is obviously wrong. Thus the composite Bézier curve is not C^1-smooth at the first common control point $\vec{P}_3 = \vec{Q}_0$. Nevertheless a common tangent direction exists at this control point (but not at the second one).

The general smoothness conditions follows from Property 2.5d.

Theorem 2.3: C^k-smoothness condition

If $\vec{r}_P(t) = \sum_{i=0}^{n} \vec{P}_i B_{in}(t) \quad t \in [0,1]$ is a Bézier curve of order $n \geq k$ with respect to the control points $\vec{P}_0, \vec{P}_1, \cdots, \vec{P}_n$ and $\vec{r}_Q(t) = \sum_{i=0}^{m} \vec{Q}_i B_{im}(t) \quad t \in [0,1]$ a Bézier curve of order $m \geq k$ for the control points $\vec{Q}_0, \vec{Q}_1, \cdots, \vec{Q}_m$ with common control point $\vec{P}_n = \vec{Q}_0$ then the composite Bézier curve is k times continuously differentiable at the common control point if and only if

$$\vec{r}_P^{\,(j)}(1) = \boxed{n(n-1)\cdots(n-j+1)(\Delta^j \vec{P}_{n-j}) = m(m-1)\cdots(m-j+1)(\Delta^j \vec{Q}_0)} = \vec{r}_Q^{\,(j)}(0)$$

for $j = 0, 1, 2, \ldots, k$.

For interpolation problems it is often desirable to parametrize a <u>composite Bézier curve with one parameter interval</u> $[u_0, u_m]$ and a parameter u. The interval is subdivided into m "patches" $[u_j, u_{j+1}]$ $(j = 0, 1, 2, \ldots, m-1)$ by $m-1$ interior knots: $u_0 < \underbrace{u_1 < u_2 < \cdots < u_{m-1}}_{\text{interior}} < u_m$. For each "patch" $[u_j, u_{j+1}]$ $(j = 0, 1, 2, \ldots, m-1)$ we have a Bézier curve of (the same) order n with $n+1$ control points $\vec{P}_{ij} \quad (i = 0, 1, \cdots, n)$ by the formulas

$$\vec{r}_j(u)=\sum_{i=0}^{n}\vec{P}_{ij}B_{in}(u,u_j.u_{j+1}) \quad u\in[u_j.u_{j+1}] \quad (j=0,1,\cdots,m-1) \qquad \textbf{(2.4)}$$

Here the transformed Bernstein polynomials of (2.1') are used. Continuity of the composite curve is fulfilled if $\vec{P}_{nj}=\vec{P}_{0,j+1} \quad (j=0,1,\cdots,m-2)$. There is a total of at most $m(n+1)$-$(m-1)$=$mn+1$ distinct control points when taking into account the continuity conditions $\vec{P}_{nj}=\vec{P}_{0,j+1} \quad (j=0,1,\cdots,m-2)$. The smoothness conditions now read as follows.

Theorem 2.4: The composite Bézier curve (2.4) is C^1-smooth if and only if

$$\frac{n(\vec{P}_{n,j}-\vec{P}_{n-1,j})}{h_j}=\frac{n(\vec{P}_{1,j+1}-\vec{P}_{0,j+1})}{h_{j+1}} \quad (j=0,1,\cdots,m-2)$$

where $h_j=u_{j+1}-u_j$.

(Proof: This follows from Theorem 2.1 by differentiating with the chain rule since we have

$$\frac{d}{du}B_{in}(u,u_{j+1},u_j)=\frac{d}{du}B_{in}\left(\frac{u-u_j}{u_{j+1}-u_j}\right)=\frac{d}{dt}B_{in}(t)\cdot\frac{1}{h_j} \quad (j=0,1,\cdots,m-2)$$ □)

The condition in Theorem 2.4 can be formulated as

$\vec{P}_{n,j}=\frac{h_{j+1}}{h_j+h_{j+1}}\vec{P}_{n-1,j}+\frac{h_j}{h_j+h_{j+1}}\vec{P}_{1,j+1} \quad (j=0,1,\cdots,m-2)$. This says that the the common control point must be an appropriate convex combination of the neighboured control points $\vec{P}_{n-1,j}$ and $\vec{P}_{1,j+1}$ and that the line from $\vec{P}_{n-1,j}$ to $\vec{P}_{1,j+1}$ is subdivided by $\vec{P}_{nj}$ in a ratio equal to $\frac{h_j}{h_{j+1}}$.

The generalization of Theorem 2.2 now is obvious.

Theorem 2.5: The composite Bézier curve (2.4) is C^2-smooth if and only if

$$\frac{n(\vec{P}_{n,j}-\vec{P}_{n-1,j})}{h_j}=\frac{n(\vec{P}_{1,j+1}-\vec{P}_{0,j+1})}{h_{j+1}} \quad (j=0,1,\cdots,m-2)$$ and

$$\frac{n(n-1)(\vec{P}_{n,j}-2\vec{P}_{n-1,j}+\vec{P}_{n-2,j})}{h_j^{\,2}}=\frac{n(n-1)(\vec{P}_{2,j+1}-2\vec{P}_{1,j+1}+\vec{P}_{0,j+1})}{h_{j+1}^{\,2}} \quad (j=0,1,\cdots,m-2)$$

where $h_j=u_{j+1}-u_j$.

(Proof: As in Theorem 2.4 this follows from Theorem 2.2 by differentiating with the chain rule since we have

$$\frac{d}{du^2}B_{in}(u,u_{j+1},u_j)=\frac{d}{du^2}B_{in}(\frac{u-u_j}{u_{j+1}-u_j})=\frac{d}{dt^2}B_{in}(t)\cdot\frac{1}{h_j^{\ 2}}\quad (j=0,1,\cdots,m-2)$$

)

A rather general smoothness theorem is the following one. It is a consequence of Property 2.5d.

Theorem 2.6: The composite Bézier curve (2.4) is C^k-smooth (n k) if and only if

$$\boxed{\frac{\Delta^\ell \vec{P}_{n-\ell,j}}{h_j^{\ \ell}}=\frac{\Delta^\ell \vec{P}_{0,j+1}}{h_{j+1}^{\ \ell}}\quad (j=0,1,\cdots,m-2\qquad \ell=0,1,2,\cdots,k)}$$

where $h_j=u_{j+1}-u_j$.

The conditions in Theorem 2.6 constitute a linear system of $(k+1)(m-1)$ equations for the $m(n+1)$ control points. This is the key to a generic solution to the interpolation problem in any dimension and with smoothness conditions of any order.

If given $m+1$ arguments $u_0<u_1<u_2<\cdots<u_{m-1}<u_m$ and corresponding points $\vec{Q}_j\quad (j=0,1,\cdots,m)$ („measurement vectors“) then Theorem 2.6 makes possible the following procedure to find a C^k-smooth composite Bézier curve interpolating $\vec{Q}_j\quad (j=0,1,\cdots,m)$ and being composed of simple Bézier curves of order n. This procedure also covers the special cases of “natural” or “clamped” splines.

Define a doubly indexed sequence of $m(n+1)$ control points $\vec{P}_{ij}\quad (i=0,1,\cdots,n,\quad j=0,1,\cdots,m-1)$ with $\vec{P}_{nj}=\vec{Q}_{j+1}=\vec{P}_{0,j+1}\quad (j=0,1,\cdots,m-2)$ and $\vec{P}_{0,0}=\vec{Q}_0$, $\vec{P}_{n,m-1}=\vec{Q}_m$. This gives a total of at most $mn+1$ distinct points. The points $\vec{Q}_j\quad (j=1,\cdots,m-1)$ are interior points playing the role of common control points. Then there are at most $m(n-1)$ control points in the set $\vec{P}_{ij}\quad (i=1,\cdots,n-1,\quad j=0,1,\cdots,m-1)$ different from all the measurement points $\vec{Q}_j\quad (j=0,1,\cdots,m)$. If smoothness conditions of order $k=n-1$ are required for the composite continuous curve we get a linear system of $(n-1)(m-1)$ equations for the $(n-1)m$ control points $\vec{P}_{ij}\quad (i=1,\cdots,n-1,\quad j=0,1,\cdots,m-1)$ by Theorem 2.6:

$$\boxed{\frac{\Delta^\ell \vec{P}_{n-\ell,j}}{h_j^{\ \ell}}=\frac{\Delta^\ell \vec{P}_{0,j+1}}{h_{j+1}^{\ \ell}}\quad (j=0,1,\cdots,m-2\qquad \ell=1,2,\cdots,n-1)}\tag{2.5}$$

This is in accordance with the second condition in Table 1.1. The difference of the number of equations and the number of control points $\vec{P}_{ij}\quad (i=1,\cdots,n-1,\quad j=0,1,\cdots,m-1)$ is $n-1$. This allows for a set of $n-1$ additional conditions on the composite Bézier curve.

Example 2.3: Interpolation of $y = \sin(x)$ from the four points $\left\{\vec{Q}_j \middle| j = 0,1,\cdots,m\right\}$ =

$\left\{(0,0),(\frac{\pi}{3},\frac{\sqrt{3}}{2}),(\frac{2\pi}{3},\frac{\sqrt{3}}{2}),(\pi,0)\right\}$. Here $m = n = 3$ and $h_j = h = \frac{\pi}{3}$. Formula (2.5) gives $(n\text{-}1)(m\text{-}1)$ = 4 equations for $(n\text{-}1)m$ = 6 unknown control points $\vec{P}_{ij} \quad (i = 1,\cdots,n-1, \quad j = 0,1,\cdots,m-1)$:

$$\vec{Q}_1 - \vec{P}_{2,0} = \vec{P}_{1,1} - \vec{Q}_1$$
$$\vec{Q}_2 - \vec{P}_{2,1} = \vec{P}_{1,2} - \vec{Q}_2$$
$$\vec{Q}_1 - 2\vec{P}_{2,0} + \vec{P}_{1,0} = \vec{P}_{2,1} - 2\vec{P}_{1,1} + \vec{Q}_1$$
$$\vec{Q}_2 - 2\vec{P}_{2,1} + \vec{P}_{1,1} = \vec{P}_{2,2} - 2\vec{P}_{1,2} + \vec{Q}_2$$

Since there are 4 equations for 6 unknown points two conditions can be met in addition. The natural spline, e.g., requires that the second derivatives at the boundary arguments 0 and π, respectively, are 0. By Theorem 2.5 it follows for the “natural” boundary conditions that:

$$\vec{P}_{2,0} - 2\vec{P}_{1,0} + \vec{Q}_0 = \vec{0}$$
$$\vec{Q}_3 - 2\vec{P}_{2,2} + \vec{P}_{1,2} = \vec{0}$$

The solution for this system of 6 linear equations in 6 unknown vectors (control points) is:

$$\left\{\left\{\frac{\pi}{9}, \frac{\sqrt{3}}{5}\right\}, \left\{\frac{2\pi}{9}, \frac{2\sqrt{3}}{5}\right\}, \left\{\frac{4\pi}{9}, \frac{3\sqrt{3}}{5}\right\}, \left\{\frac{5\pi}{9}, \frac{3\sqrt{3}}{5}\right\}, \left\{\frac{7\pi}{9}, \frac{2\sqrt{3}}{5}\right\}, \left\{\frac{8\pi}{9}, \frac{\sqrt{3}}{5}\right\}\right\}$$

The figure below shows the 10 distinct control points together with the sine-curve (dashed) and the composite Bézier curve. The red points are the common (interior) control points or boundary control points, respectively.

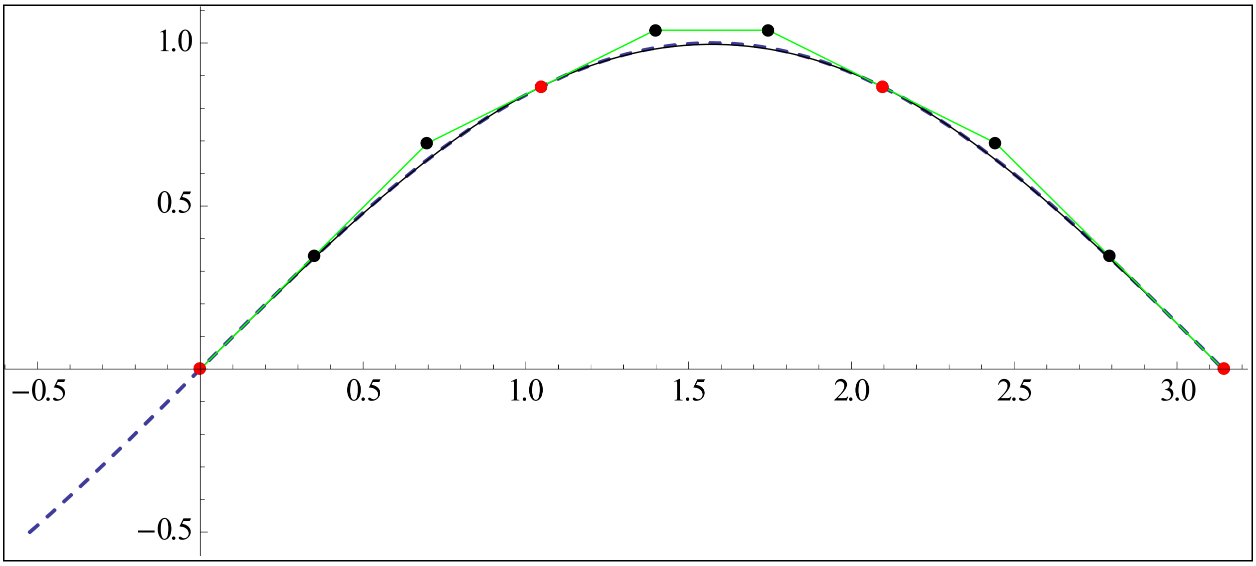


Figure 2.4: 10 distinct control points (two of them as common) for the sine-curve in the interval [0, π] and the composite Bézier curve of order 3. The difference between the sine-curve (dashed) and the Bézier curve is rather small. The computations could be reduced by using symmetries for the sine-curve, of course.

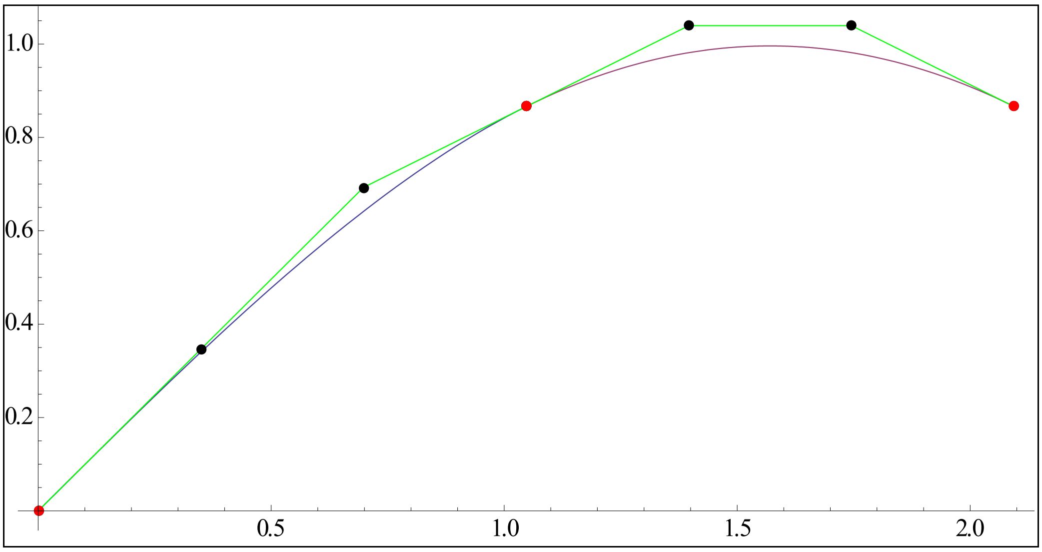


Figure 2.5: The first 7 control points and their composite Bézier spline of order 3. Its vector formulas $\vec{r}_1(t)$ and $\vec{r}_2(t)$, respectively, for $t \in [0,1]$ are

$$\begin{pmatrix} \frac{1}{9}\pi B_{1,3}(t) + \frac{2}{9}\pi B_{2,3}(t) + \frac{1}{3}\pi B_{3,3}(t) \\ \frac{1}{5}\sqrt{3}\, B_{1,3}(t) + \frac{2}{5}\sqrt{3}\, B_{2,3}(t) + \frac{1}{2}\sqrt{3}\, B_{3,3}(t) \end{pmatrix},$$

and

$$\begin{pmatrix} \frac{1}{3}\pi B_{0,3}(t) + \frac{4}{9}\pi B_{1,3}(t) + \frac{5}{9}\pi B_{2,3}(t) + \frac{2}{3}\pi B_{3,3}(t) \\ \frac{1}{2}\sqrt{3}\, B_{0,3}(t) + \frac{3}{5}\sqrt{3}\, B_{1,3}(t) + \frac{3}{5}\sqrt{3}\, B_{2,3}(t) + \frac{1}{2}\sqrt{3}\, B_{3,3}(t) \end{pmatrix}$$

The second derivative of $\vec{r}_1(t)$ is $\begin{pmatrix} 0 \\ \frac{-3\sqrt{3}}{\sqrt{5}} B_{1,1}(t) \end{pmatrix}$ which yields $\vec{0}$ for t = 0 and $\begin{pmatrix} 0 \\ \frac{-3\sqrt{3}}{\sqrt{5}} \end{pmatrix}$ for t = 1.

The second derivative of $\vec{r}_2(t)$ is $\begin{pmatrix} 0 \\ \frac{-3\sqrt{3}}{\sqrt{5}} (B_{0,1}(t) + B_{1,1}(t)) \end{pmatrix}$ which yields $\begin{pmatrix} 0 \\ \frac{-3\sqrt{3}}{\sqrt{5}} \end{pmatrix}$ again

for t = 0.

2.5 Bézier surfaces as tensor splines

Bézier surfaces can be generated rather easily by combining Bézier curves with a tensor product of Bernstein polynomials. The formula for a Bézier surface on the parameter range $[0,1]$x$[0,1]$ is:

$$\vec{z}(s,t) = \sum_{i=0}^{n}\sum_{j=0}^{m} \vec{P}_{ij} B_{in}(s) B_{jm}(t) \qquad s,t \in [0,1] \tag{2.6}$$

Here $\vec{P}_{ij}$ denotes a matrix of $(n+1)(m+1)$ control points (in R^d), also called Bézier points.

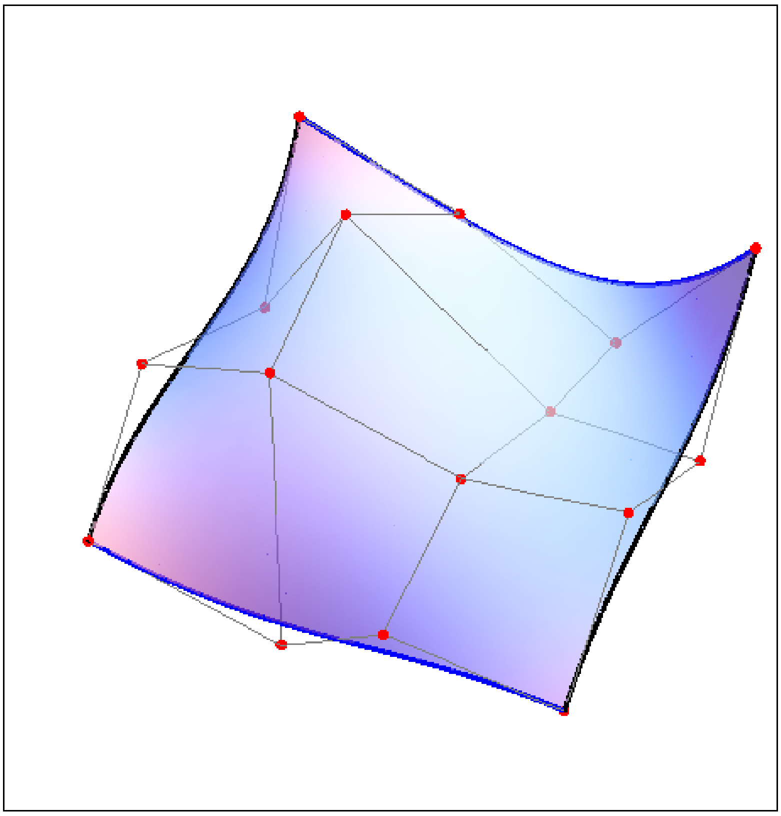


Figure 2.6: A Bézier tensor spline with m = n = 3 and sixteen control points in R^3. The four highlighted boundary curves are Bézier curves with corresponding boundary control points. On the boundary we have s = 0 or 1 and t = 0 or 1, respectively.

The following properties for a Bézier surface follow immediately from (2.6) and the properties of Bézier curves.

Property 2.6: Elementary properties for the Bézier tensor spline (2.6)

a) Corner points: $\vec{z}(0,0) = \vec{P}_{0,0}$, $\vec{z}(0,1) = \vec{P}_{0,m}$, $\vec{z}(1,0) = \vec{P}_{n,0}$, $\vec{z}(1,1) = \vec{P}_{n,m}$

b) Boundary curves:

$$\vec{z}(0,t) = \sum_{j=0}^{m} \vec{P}_{0j} B_{jm}(t), \qquad \vec{z}(1,t) = \sum_{j=0}^{m} \vec{P}_{nj} B_{jm}(t)$$

$$\vec{z}(s,0) = \sum_{i=0}^{n} \vec{P}_{i0} B_{in}(s), \qquad \vec{z}(s,1) = \sum_{i=0}^{n} \vec{P}_{im} B_{in}(s)$$

c) Convex hull: The set of points of the Bézier surface $Z = \{\vec{z}(s,t) | s,t \in [0,1]\times[0,1]\}$ lies in the convex hull of the control points.

From Property 2.3 we get derivative formulas for the Bézier surface at the boundary.

Property 2.7: First and second order partial derivatives:

a)

$$\frac{\partial}{\partial s}\vec{z}(s,t) = n\sum_{i=0}^{n-1}\sum_{j=0}^{m}(\vec{P}_{i+1,j} - \vec{P}_{i,j})B_{i,n-1}(s)B_{jm}(t)$$

$$\frac{\partial}{\partial t}\vec{z}(s,t) = m\sum_{i=0}^{n}\sum_{j=0}^{m-1}(\vec{P}_{i,j+1} - \vec{P}_{i,j})B_{i,n}(s)B_{j,m-1}(t) \qquad s,t\in[0,1]$$

b)

$$\frac{\partial^2}{\partial s^2}\vec{z}(s,t) = n(n-1)\sum_{i=0}^{n-2}\sum_{j=0}^{m}\underbrace{(\vec{P}_{i+2,j} - 2\vec{P}_{i+1,j} + \vec{P}_{i,j})}_{\Delta^2\vec{P}_{i,j}}B_{i,n-2}(s)B_{jm}(t)$$

$$\frac{\partial^2}{\partial t^2}\vec{z}(s,t) = m(m-1)\sum_{i=0}^{n}\sum_{j=0}^{m-2}\underbrace{(\vec{P}_{i,j+2} - 2\vec{P}_{i,j+1} + \vec{P}_{i,j})}_{\Delta^2\vec{P}_{i,j}}B_{i,n}(s)B_{j,m-2}(t)$$

$$\frac{\partial^2}{\partial t\partial s}\vec{z}(s,t) = nm\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\underbrace{(\vec{P}_{i+1,j+1} - \vec{P}_{i+1,j} - \vec{P}_{i,j+1} + \vec{P}_{i,j})}B_{i,n-1}(s)B_{j,m-1}(t)$$

$$\frac{\partial^2}{\partial s\partial t}\vec{z}(s,t) = mn\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\underbrace{(\vec{P}_{i+1,j+1} - \vec{P}_{i,j+1} - \vec{P}_{i+1,j} + \vec{P}_{i,j})}B_{i,n-1}(s)B_{j,m-1}(t) \qquad s,t\in[0,1]$$

There is, of course, also a scheme for higher order partial derivatives but this is neither handsome nor very often used. The following property is important for smooth composition of Bézier surfaces along common boundary control points. It is a consequence of Property 2.7.

Property 2.8: First and second order partial derivatives at boundary curves

a)

$$\frac{\partial}{\partial s}\vec{z}(0,t) = n\sum_{j=0}^{m}(\vec{P}_{1,j} - \vec{P}_{0,j})B_{jm}(t) \qquad \frac{\partial}{\partial s}\vec{z}(1,t) = n\sum_{j=0}^{m}(\vec{P}_{n,j} - \vec{P}_{n-1,j})B_{jm}(t)$$

$$\frac{\partial}{\partial t}\vec{z}(s,0) = m\sum_{i=0}^{n}(\vec{P}_{i,1} - \vec{P}_{i,0})B_{in}(s) \qquad \frac{\partial}{\partial t}\vec{z}(s,1) = m\sum_{i=0}^{n}(\vec{P}_{i,m} - \vec{P}_{i,m-1})B_{in}(s)$$

b1)

$$\frac{\partial^2}{\partial s^2}\vec{z}(0,t) = n(n-1)\sum_{j=0}^{m}\underbrace{(\vec{P}_{2,j} - 2\vec{P}_{1,j} + \vec{P}_{0,j})}_{\Delta^2\vec{P}_{0,j}}B_{jm}(t)$$

$$\frac{\partial^2}{\partial s^2}\vec{z}(1,t) = n(n-1)\sum_{j=0}^{m}\underbrace{(\vec{P}_{n,j} - 2\vec{P}_{n-1,j} + \vec{P}_{n-2,j})}_{\Delta^2\vec{P}_{n-2,j}}B_{jm}(t)$$

$$\frac{\partial^2}{\partial t^2}\vec{z}(s,0) = m(m-1)\sum_{i=0}^{n}\underbrace{(\vec{P}_{i,2} - 2\vec{P}_{i,1} + \vec{P}_{i,0})}_{\Delta^2\vec{P}_{i,0}}B_{i,n}(s)$$

$$\frac{\partial^2}{\partial t^2}\vec{z}(s,1) = m(m-1)\sum_{i=0}^{n}\underbrace{(\vec{P}_{i,m} - 2\vec{P}_{i,m-1} + \vec{P}_{i,m-2})}_{\Delta^2\vec{P}_{i,m-2}}B_{i,n}(s)$$

b2)

$$\frac{\partial^2}{\partial t\partial s}\vec{z}(0,t) = nm\sum_{j=0}^{m-1}\underbrace{(\vec{P}_{1,j+1} - \vec{P}_{1,j} - \vec{P}_{0,j+1} + \vec{P}_{0,j})}B_{j,m-1}(t)$$

$$\frac{\partial^2}{\partial t\partial s}\vec{z}(1,t) = nm\sum_{j=0}^{m-1}\underbrace{(\vec{P}_{n,j+1} - \vec{P}_{n,j} - \vec{P}_{n-1,j+1} + \vec{P}_{n-1,j})}B_{j,m-1}(t)$$

$$\frac{\partial^2}{\partial t\partial s}\vec{z}(s,0) = nm\sum_{i=0}^{n-1}\underbrace{(\vec{P}_{i+1,1} - \vec{P}_{i+1,0} - \vec{P}_{i,1} + \vec{P}_{i,0})}B_{i,n-1}(s)$$

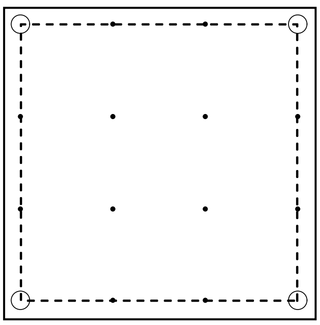
$$\frac{\partial^2}{\partial t\partial s}\vec{z}(s,1) = nm\sum_{i=0}^{n-1}\underbrace{(\vec{P}_{i+1,m} - \vec{P}_{i,m} - \vec{P}_{i+1,m-1} + \vec{P}_{i,m-1})}B_{i,n-1}(s)$$

2.6 Composite Bézier surfaces and bi-cubic interpolation

As Bézier curves also Bézier surfaces can be composed. Thereby Properties 2.7 and 2.8 have to be applied to guarantee smoothness conditions along the common boundary curves. A typical case is the composition of cubic splines in order to interpolate values on a rectangular grid. This is bi-cubic Bézier spline interpolation.

Example 2.4: Bi-cubic Bézier interpolation in R^3

Given four points $\{(x_0,y_0),(x_0+h_1,y_0),(x_0+h_1,y_0+h_2),(x_0,y_0+h_2)\}$ in the corners of a square grid one has to find a bi-cubic polynomial $p(s,t)$ with $(s,t) \in [0,1]\times[0,1]$ with the following conditions:



(1) The polynomial p takes given values $z_{0,0}, z_{1,0}, z_{1,1}, z_{0,1}$ at the four corner points: This gives a number of 4 conditions.

(2) The surface $\vec{z}(s,t) := (s,t,p(s,t)) \quad (s,t \in [0,1])$ has slopes $z_s'_{0,0}, z_s'_{1,0}, z_s'_{1,1}, z_s'_{0,1}$ and $z_t'_{0,0}, z_t'_{1,0}, z_t'_{1,1}, z_t'_{0,1}$, respectively, for its partial first order derivative vectors at the corner points. The index s or t, respectively, refer to whether the partial derivatives is with respect to s or t, respectively. This, additionally, gives 2*4 conditions by using Property 2.8a.

Figure 2.7: Uniform grid of 16 projected control points (z = 0).

(3) Mixed partial derivative conditions: The surface $\vec{z}(s,t) := (s, t, p(s,t)) \quad (s,t \in [0,1])$ has slope values $z_s''_{0,0}, z_s''_{1,0}, z_s''_{1,1}, z_s''_{0,1}$ for the partial derivatives $\frac{\partial}{\partial t}\vec{z}(s,t)$ at the corner points. This gives 4 additional conditions by using Property 2.8b2).

The conditions (1) to (3) give a total of 16 conditions for the 16 control points. The projections (z = 0) of the control points subdivide the square grid uniformly as indicated in Figure 2.7. The bi-cubic Bézier polynomial is defined by (2.6). Because conditions on function values, partial derivatives and mixed partial derivatives at the corners are met it is possible to compose bi-cubic Bézier surfaces for neighboured square grids such that values, partial derivatives and mixed partial derivatives coincide at the common corner points.

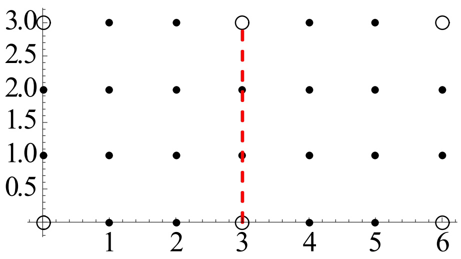


Figure 2.8: Two neighboured square grids sharing two common corner points on the red dashed line.

Example 2.5: Composition of bi-cubic Bézier surfaces in R^d

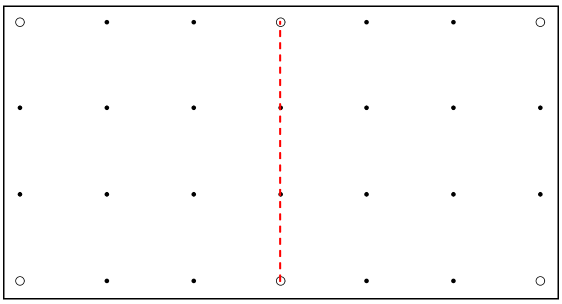


Figure 2.9: Scheme representation of two matrices of control points. The $n+1$ = 4 control points in the red column are common. The composite Bézier surface is continuous along the Bézier curve belonging to the common $n+1$ = 4 control points. When the $(m+1)(n+1)$ = 16 control points $\vec{P}_{ij} \quad (i = 0,\ldots,n, \quad j = 0,\ldots,m)$ of the first (left side) matrix are given their Bézier surface $\vec{z}_1(s,t)$ is defined by (2.6). Denoting the points in the second (right side) matrix by $\vec{Q}_{ij} \quad (i = 0,\ldots,n, \quad j = 0,\ldots,m)$ we have that $\vec{Q}_{i0} = \vec{P}_{im} \quad (i = 0,\ldots,n)$ (four common control points indicated by the red dashed line). The Bézier surface to be defined by $\vec{Q}_{ij} \quad (i = 0,\ldots,n, \quad j = 0,\ldots,m)$ is denoted by $\vec{z}_2(s,t)$. We have that $\vec{z}_1(s,1) = \vec{z}_2(s,0)$ (common Bézier curve according to four common control points).

First-order conditions: $\frac{\partial}{\partial t}\vec{z}_1(s,1) = \frac{\partial}{\partial t}\vec{z}_2(s,0)$. By Property 2.8a we get the following n+1 linear equations:

$$\boxed{\vec{P}_{i,m} - \vec{P}_{i,m-1} = \vec{Q}_{i,1} - \vec{Q}_{i,0} \quad (i = 0,\ldots,n)} \tag{2.7}$$

Second-order conditions: $\frac{\partial^2}{\partial t^2}\vec{z}_1(s,1) = \frac{\partial^2}{\partial t^2}\vec{z}_2(s,0)$. By Property 2.8b1 we get the following n+1 linear equations:

$$\boxed{\vec{P}_{i,m} - 2\vec{P}_{i,m-1} + \vec{P}_{i,m-2} = \vec{Q}_{i,2} - 2\vec{Q}_{i,1} + \vec{Q}_{i,0} \quad (i = 0,\ldots,n)} \tag{2.8}$$

Mixed second-order conditions: $\frac{\partial^2}{\partial t \partial s}\vec{z}_1(s,1) = \frac{\partial^2}{\partial t \partial s}\vec{z}_2(s,0)$. By Property 2.8b2 we get the following n linear equations:

$$\boxed{\vec{P}_{i+1,m} - \vec{P}_{i,m} - \vec{P}_{i+1,m-1} + \vec{P}_{i,m-1} = \vec{Q}_{i+1,1} - \vec{Q}_{i+1,0} - \vec{Q}_{i,1} + \vec{Q}_{i,0} \quad (i = 0,\ldots,n-1)} \tag{2.8'}$$

Thus there is a total of $2(n+1)+n$ = 11 equations in the $2(n+1)$ = 8 unknown variable control points $\vec{Q}_{ij} \quad (i = 0,\ldots,n, \quad j = 1,\ldots,2)$. Generally, the n mixed second-order conditions (2.8') are dropped. On the other side the control points $\vec{Q}_{ij} \quad (i = 0,\ldots,n \quad j = 3,\ldots,m)$ do not occur in the linear equations (2.7) and (2.8). Thus,e.g., the boundary curve $\vec{z}_2(s,1)$ may be defined freely as a Bézier curve or subject to other contraints than (2.7 or 2.8).