I. Curves, part 1

Summer Term 2025

Prof. Dr. Dr. Fellner
Dr. Daniel Ströter

GRIS

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

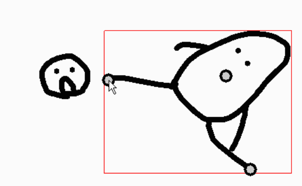
Why do we need curves in computer science/graphics?

GRIS

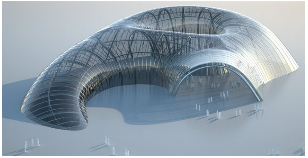
Motivation

Why do we need curves in computer science/graphics?

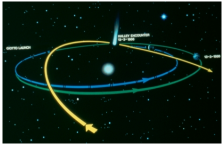
- Parametric design is art and architecture
- Describing trajectories
- Motion and smooth detail in movies
- Smooth shapes in product design/engineering
- . . .



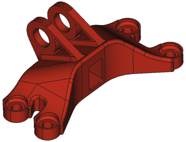
Igarashi (Eurographics ’08)



Pottmann et al. ”Freeform surfaces from single curved panels“



https://sci.esa.int/web/giotto/-/36674-comet-encounter-diagram



SimJEB Bracket 10

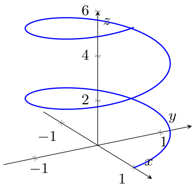
Parametric Curves

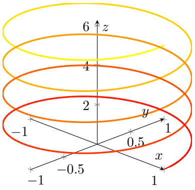
The function

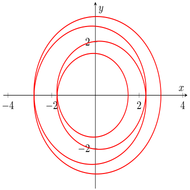
$$\mathbf{X}: \ [a,b] \to \mathbb{R}^d, \ \mathbf{X}(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_d(t) \end{pmatrix} \in \mathbb{R}^d, \ t \in [a,b]$$

is a **parametric** representation of the curve:

$$C_{\mathbf{X}} = \{\mathbf{X}(t) : \ t \in [a,b]\}.$$





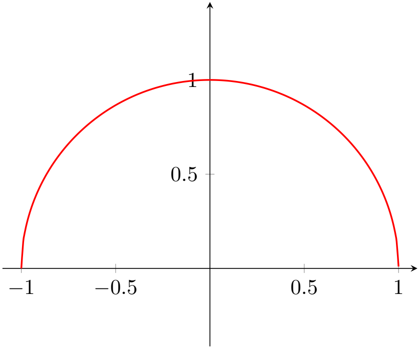


Facts & Examples

The functions

$$\mathbf{X}(t) = \begin{pmatrix} \cos(\pi + t) \\ \sin(t) \end{pmatrix}, \; t \in [0, \pi]$$

$$\tilde{\mathbf{X}}(t) = \begin{pmatrix} t \\ \sqrt{1 - t^2} \end{pmatrix}, \; t \in [-1, 1]$$



parameterize the same curve $C_{\mathbf{X}} = C_{\tilde{\mathbf{X}}}$ (semicircle).

Parametric Curves

Likewise, the terms

$$\begin{aligned} \mathbf{X}(t) &= \begin{pmatrix} t^2 \\ t^2 \end{pmatrix} = t^2 \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; t \in [0,1] \\ \tilde{\mathbf{X}}(t) &= \begin{pmatrix} t \\ t \end{pmatrix} = t \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; t \in [0,1] \end{aligned}$$

parameterize the same curves.

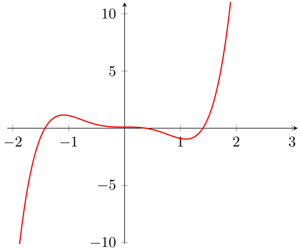
$\Rightarrow$ parametric representations of a curve are not unique!

Functional Curves

For the real function, $f : [a, b] \to \mathbb{R}$

$$\mathbf{X}_f(t) = \begin{pmatrix} t \\ f(t) \end{pmatrix} \in \mathbb{R}^2, \ t \in [a, b]$$

parameterizes a special curve $C_{\mathbf{X}_f}$:
the graph of f
(also: a **functional** curve).

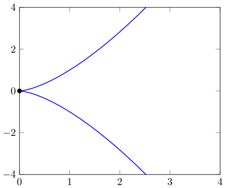


The curve $\mathbf{X}\colon [a,b] \to \mathbb{R}^n$ is regular iff. $\mathbf{X}'(t) \neq \mathbf{0}$ for all $t \in [a,b]$.

The curve $\mathbf{X}\colon [a,b] \to \mathbb{R}^n$ is singular at $\mathbf{X}(t)$ iff. $\mathbf{X}'(t) = \mathbf{0}$.

Neil's parabola . . .

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

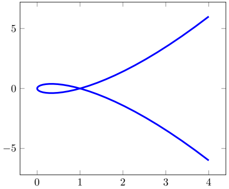


. . . is **non-regular**, because

$$\mathbf{X}'(0) = \begin{pmatrix} 2 \cdot 0 \\ 3 \cdot 0^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

For

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



applies $\mathbf{X}(-1) = \binom{1}{0} = \mathbf{X}(1)$.

$\mathbf{X}$ exhibits a **self-intersection** in $t = -1$.

Curves – Continuity

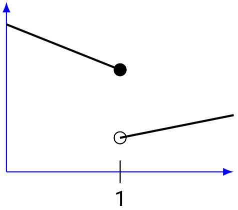
The function $f(t)$ is **continuous** in t_0 iff.

1. $\lim_{t \to t_0} f(t)$ exists and
2. $\lim_{t \to t_0} f(t) = f(t_0)$

In other words: f is continuous iff. for each $\epsilon > 0$ there is a $\delta > 0$ such that

$$|f(t) - f(t_0)| < \epsilon$$

in the local neighborhood of t_0 with radius δ (thus $|t - t_0| < \delta$).



non-continuous function in $t_0 = 1$

If f is continuous for all $t \in [a, b]$, then f is continuous ($f \in C^0$) in $[a, b]$.

Polynomial Curves

Special class of curves: **Polynomial curves** $C_{\mathbf{P}}$ determined by

$$\mathbf{P}(t) = \begin{pmatrix} p_1(t) \\ \vdots \\ p_d(t) \end{pmatrix} = \mathbf{a}_0 + t\,\mathbf{a}_1 + \ldots + t^q\,\mathbf{a}_q \in \mathbb{R}^d,\ t \in [a,b],$$

where $\mathbf{a}_i \in \mathbb{R}^d,\ i = 0, \ldots, q$ (coefficient vectors).

p_i are *polynomial functions*, q is the *degree* of $\mathbf{P}$.

Monomial representation (also referred to: Taylor representation in 0), because $q+1$ monomial

$$1, t, \ldots, t^q$$

are used. Therefore: $(q+1) \cdot d$ degrees of freedom.

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples

- Neil's parabola
- $$\mathbf{P}(t) = \begin{pmatrix} t + 3\,t^2 \\ -2 - 3t^2 \\ 1 + t \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot t + \begin{pmatrix} 3 \\ -3 \\ 0 \end{pmatrix} \cdot t^2$$
- corresponding tangent vector (velocity)
$$\mathbf{P}'(t) = \begin{pmatrix} 1 + 6\,t \\ -6t \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 6 \\ -6 \\ 0 \end{pmatrix} \cdot t$$
- 2. derivative $\mathbf{P}''(t) = \begin{pmatrix} 6 \\ -6 \\ 0 \end{pmatrix}$

GRIS

Polynomial Curves

Beneficial properties of polynomial curves $\mathbf{P} = \mathbf{P}(t) = \sum_{i=0}^{q} \mathbf{a}_i t^i$:

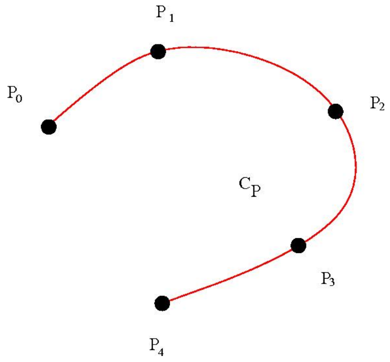
- differentiable as often as required
- Relies on vector space: If $\mathbf{P}_1,\ \mathbf{P}_2$ are polynomial curves and $\alpha, \beta \in \mathbb{R}$, then $\alpha \cdot \mathbf{P}_1 + \beta \cdot \mathbf{P}_2$ is also a polynomial curve.
- Interpolation can always be applied
- Fast and exact evaluation
- ...

Interpolation with Polynomial Curves

For an arbitrary choice of interpolation points $a \leq t_0 < \ldots < t_q \leq b$ and data points $\mathbf{P}_0, \ldots, \mathbf{P}_q \in \mathbb{R}^d$ there is **a unique** polynomial curve **P** of degree q such that

$$\mathbf{P}(t_i) = \mathbf{P}_i,\ i = 0, \ldots, q.$$

In the example: $q = 4$



Choice of Interpolation Points: Typical Parametrization

Typically only the data points $\mathbf{P}_0, \ldots, \mathbf{P}_q \in \mathbb{R}^d$ are given.
Frequent choice of parametrization:

equidistant $t_i = a + i \cdot \frac{b-a}{q},\ i = 0, \ldots, q.$

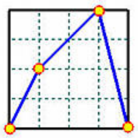
Other choices:

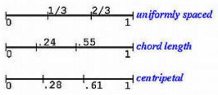
cordal $t_{i+1} - t_i = \|\mathbf{P}_{i+1} - \mathbf{P}_i\|_2,\ i = 0, \ldots, q-1$

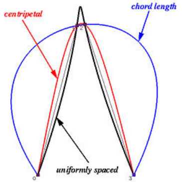
centripetal $t_{i+1} - t_i = \sqrt{\|\mathbf{P}_{i+1} - \mathbf{P}_i\|_2},\ i = 0, \ldots, q-1$

Tschebyscheff $t_i = \cos(\frac{2(q-i)+1}{q+1}\frac{\pi}{2}),\ i = 0, \ldots, q.$

($\|.\|_2$ is Euclidean norm)







C.K. Shene, `http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/INT-APP/PARA-centripetal.html`

TECHNISCHE UNIVERSITÄT DARMSTADT

Interpolation in monomial base

Determine polynomial $\mathbf{P}(t) = \sum_{i=0}^{q} \mathbf{a}_i t^i$ such that $\mathbf{P}(t_i) = \mathbf{P}_i$, for $i = 0, \ldots, q$
leads to **Vandermonde** matrix **V**

$$\begin{pmatrix} 1 & t_0 & t_0^2 & \ldots & t_0^q \\ 1 & t_1 & t_1^2 & \ldots & t_1^q \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_q & t_q^2 & \ldots & t_q^q \end{pmatrix} \begin{pmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_q \end{pmatrix} = \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_q \end{pmatrix}, \text{ i.e, } \mathbf{Va} = \mathbf{P}$$

Coefficient vector **a** computed as $\mathbf{a} = \mathbf{V}^{-1}\mathbf{P}$

- It works out, because determinant $\det(\mathbf{V}) \neq 0$ for $t_0 < t_1 < t_2 < \ldots$
- but: **V** can be ill-conditioned $\rightarrow$ numerically unstable
- expensive, because inverse of **V** needs to be computed (Cost: $\mathcal{O}(q^3)$).

GRIS

Interpolation with polynomial curves:
Lagrange Interpolation

Lagrange Representation

$$\mathbf{P}(t) = \sum_{i=0}^{q} \ell_i(t)\,\mathbf{P}_i,\; t \in [a,b]$$

relies on Lagrange base $\ell_i,\; i = 0,\dots,q$, (degree q):

$$\ell_i(t) = \prod_{j=0, j\neq i}^{q} \frac{t - t_j}{t_i - t_j} = \frac{(t-t_0)\dots(t-t_{i-1})(t-t_{i+1})\dots(t-t_q)}{(t_i-t_0)\dots(t_i-t_{i-1})(t_i-t_{i+1})\dots(t_i-t_q)},\; t \in [a,b]$$

The Lagrange polynomials form the partition of unity:

$$\sum_{i}^{q} \ell_i(t) = 1$$

Lagrange-Interpolation

The construction of ℓ_i implies: $\ell_i(t_j) = \delta_{ij} := \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$.

Thus the polynomial

$$\mathbf{P}(t) = \sum_{i=0}^{q} \ell_i(t)\, \mathbf{P}_i, \; t \in [a, b]$$

is the unique solution to $\mathbf{P}(t_j) = \mathbf{P}_j, \; j = 0, \ldots, q$.

- easy to calculate
- no matrix inversion
- relatively expensive evaluation of ℓ_i

Example: Your homework!

Interpolation with Polynomial Curves: Newton Interpolation

Newton Representation

$$\mathbf{P}(t) = \sum_{i=0}^{q} \omega_i(t)\, \Delta(t_0, \ldots, t_i),\ t \in [a, b]$$

relies on Newton base $\omega_i,\ i = 0, \ldots, q$, (degree i):

$$\omega_0(t) = 1,\ \omega_i(t) = (t - t_0)\ldots(t - t_{i-1}),\ t \in [a, b]$$

Newton's idea: For $j = 1, \ldots, q$, determine $\mathbf{P}^{[j]}$ as a solution to $\mathbf{P}^{[j]}(t_i) = \mathbf{P}_i,\ i = 0, \ldots, j$ by (re-)usage of $\mathbf{P}^{[j-1]}$.

q=0 $\mathbf{P}^{[0]}(t) = \mathbf{P}_0$

q=1 $\mathbf{P}^{[1]}(t) = \mathbf{P}_0 + \overbrace{(t - t_0)}^{\omega_1(t)} \frac{\mathbf{P}_1 - \mathbf{P}_0}{t_1 - t_0}$

Newton Representation

TECHNISCHE UNIVERSITÄT DARMSTADT

q=0 $\mathbf{P}^{[0]}(t) = \mathbf{P}_0$

q=1 $\mathbf{P}^{[1]}(t) = \mathbf{P}_0 + \overbrace{(t - t_0)}^{\omega_1(t)} \frac{\mathbf{P}_1 - \mathbf{P}_0}{t_1 - t_0}$

q=2 Approach $\mathbf{P}^{[2]}(t) = \mathbf{P}^{[1]}(t) + \overbrace{(t - t_0)(t - t_1)}^{\omega_2(t)}\,\mathbf{B}$
Straightforwardly, it holds that: $\mathbf{P}^{[2]}(t_i) = \mathbf{P}^{[1]}(t_i)$ for $i = 0, 1$.
Remaining condition $\mathbf{P}^{[2]}(t_2) = \mathbf{P}_2$ leads to

$$\begin{aligned}
\mathbf{B} &= \left(\left(\frac{\mathbf{P}_2 - \mathbf{P}_0}{t_2 - t_0}\right) - \left(\frac{\mathbf{P}_1 - \mathbf{P}_0}{t_1 - t_0}\right)\right) \Big/ (t_2 - t_1) \\
&= \left(\left(\frac{\mathbf{P}_2 - \mathbf{P}_1}{t_2 - t_1}\right) - \left(\frac{\mathbf{P}_1 - \mathbf{P}_0}{t_1 - t_0}\right)\right) \Big/ (t_2 - t_0)
\end{aligned}$$

GRIS

Newton Representation

For general $q \geq 0$

k-th difference regarding $a \leq t_j \dots t_{j+k} \leq b$ is recursively defined by $\Delta(t_i) := \mathbf{P}_i,\ i = j, \dots, j+k$, and

$$\Delta(t_j, \dots, t_{j+k}) := \frac{\Delta(t_{j+1}, \dots, t_{j+k}) - \Delta(t_j, \dots, t_{j+k-1})}{t_{j+k} - t_j}.$$

Computation as triangle scheme – Cost: $\mathcal{O}(q^2)$

Formula of Newton interpolation:

$$\begin{aligned}\mathbf{P}(t) &= \Delta(t_0) + \omega_1(t)\,\Delta(t_0, t_1) + \dots + \omega_q(t)\,\Delta(t_0, \dots, t_q)\\ &= \sum_{i=0}^{q} \omega_i(t)\,\Delta(t_0, \dots, t_i),\ t \in [a, b].\end{aligned}$$

Triangle scheme of shared differences

$$
\begin{array}{cccccc|ccccc}
 & & & & t_0 & & \Delta(t_0) & & & & \\
 & & & t_1 - t_0 & & & & \Delta(t_0, t_1) & & & \\
 & & t_2 - t_0 & & t_1 & & \Delta(t_1) & & \Delta(t_0, t_1, t_2) & & \\
 & & \vdots & t_2 - t_1 & & & & \Delta(t_1, t_2) & \vdots & & \\
 & \iddots & \vdots & \vdots & t_2 & & \Delta(t_2) & \vdots & \vdots & \ddots & \\
t_q - t_0 & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \Delta(t_0, \ldots, t_q) \\
 & \ddots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \iddots & \\
 & & t_q - t_{q-2} & \vdots & t_{q-1} & & \Delta(t_{q-1}) & & \Delta(t_{q-2}, t_{q-1}, t_q) & & \\
 & & & t_q - t_{q-1} & & & & \Delta(t_{q-1}, t_q) & & & \\
 & & & & t_q & & \Delta(t_q) & & & &
\end{array}
$$

Newton Representation Example

Example: determine polynomial $p\colon [-2,2] \to \mathbb{R}$ of degree 4 such that

$$p(i) = |i|,\ i = -2, \dots, 2.$$

Computation of shared differences (triangle scheme):

$$
\begin{array}{ccccc|ccccc}
 & & & & -2 & 2 & & & & \\
 & & & 1 & & & -1 & & & \\
 & & 2 & & -1 & 1 & & 0 & & \\
 & 3 & & 1 & & & -1 & & 1/3 & \\
4 & & 2 & & 0 & 0 & & 1 & & -1/6 \\
 & 3 & & 1 & & & 1 & & -1/3 & \\
 & & 2 & & 1 & 1 & & 0 & & \\
 & & & 1 & & & 1 & & & \\
 & & & & 2 & 2 & & & &
\end{array}
$$

Newton representation of p, $t \in [-2, 2]$:

$$p(t) = 2 + (-1)\ (t+2) + 0\ (t+2)(t+1) + 1/3\ (t+2)(t+1)t + (-1/6)\ (t+2)(t+1)t(t-1)$$

Newton Representation Example

Computation of shared differences (triangle scheme):

$$
\begin{array}{ccccc|ccccc}
 & & & & -2 & 2 & & & & \\
 & & & 1 & & & -1 & & & \\
 & & 2 & & -1 & 1 & & \boxed{0} & & \\
 & 3 & & 1 & & & \boxed{-1} & & \boxed{1/3} & \\
4 & & 2 & & 0 & \boxed{0} & & 1 & & \boxed{-1/6} \\
 & 3 & & 1 & & & 1 & & -1/3 & \\
 & & 2 & & 1 & 1 & & 0 & & \\
 & & & 1 & & & 1 & & & \\
 & & & & 2 & 2 & & & &
\end{array}
$$

Alternative Newton representation of p:

$$
\begin{aligned}
p(t) &= 2 + 1\,(t-2) + 0\,(t-2)(t-1) + (-1/3)\,(t-2)(t-1)t + (-1/6)\,(t-2)(t-1)t(t+1) \\
&= \boxed{0} + \boxed{(-1)}\,t + \boxed{0}\,t(t+1) + \boxed{1/3}\,t(t+1)(t+2) + \boxed{(-1/6)}\,t(t+1)(t+2)(t-1)
\end{aligned}
$$

- the insertion order of data points does not affect the resulting polynomial
- since ω_i changes accordingly

Evaluation of Polynomial Curves

Naive, direct evaluation: $\mathcal{O}(q^2)$

Monomial representation

$$\mathbf{P}(\xi) = \sum_{i=0}^{q} \underbrace{\overbrace{\xi \ldots \xi}^{(i-1)\text{ multiplications}}}_{\xi^i} \quad \mathbf{a}_i$$

Lagrange representation

$$\mathbf{P}(\xi) = \sum_{i=0}^{q} \underbrace{\overbrace{\frac{(\xi - t_0)\ldots(\xi - t_{i-1})(\xi - t_{i+1})\ldots(\xi - t_q)}{(t_i - t_0)\ldots(t_i - t_{i-1})(t_i - t_{i+1})\ldots(t_i - t_q)}}^{(2q-1)\text{ multiplications}}}_{\ell_i(\xi)} \mathbf{P}(t_i)$$

Newton representation

$$\mathbf{P}(\xi) = \sum_{i=0}^{q} \underbrace{\overbrace{(\xi - t_0)\ldots(\xi - t_{i-1})}^{(i-1)\text{ multiplications}}}_{\omega_i(\xi)} \mathbf{b}_i$$

Evaluation of Polynomial Curves

TECHNISCHE UNIVERSITÄT DARMSTADT

Suppose **P** is provided in Newton representation

$$\begin{aligned}\mathbf{P}(t) &= \mathbf{b}_0 + (t-t_0)\,\mathbf{b}_1 + (t-t_0)(t-t_1)\,\mathbf{b}_2 + \dots \\ &\quad +(t-t_0)\dots(t-t_{q-2})\,\mathbf{b}_{q-1} + (t-t_0)\dots(t-t_{q-1})\,\mathbf{b}_q \\ &= \mathbf{b}_0 + (t-t_0)\,(\mathbf{b}_1 + (t-t_1)\,(\mathbf{b}_2 + \dots \\ &\quad +(t-t_{q-2})\,(\mathbf{b}_{q-1} + (t-t_{q-1})\,\mathbf{b}_q)\dots))\,.\end{aligned}$$

Algorithm: modified Horner scheme

Compute $\mathbf{P}(\xi)$: Set $\mathbf{b}_i^{[0]} = \mathbf{b}_i,\ i = 0,\dots,q$, $\mathbf{b}_q^{[1]} := \mathbf{b}_q^{[0]}$,
and compute for each $i = q,\dots,1$ successively

$$\mathbf{b}_{i-1}^{[1]} = \mathbf{b}_{i-1}^{[0]} + (\xi - t_{i-1})\,\mathbf{b}_i^{[1]}.$$

Modified Horner Scheme

TECHNISCHE UNIVERSITÄT DARMSTADT

Compute $\mathbf{P}(\xi)$: Set $\mathbf{b}_i^{[0]} = \mathbf{b}_i,\ i = 0, \ldots, q$, $\mathbf{b}_q^{[1]} := \mathbf{b}_q^{[0]}$,
and compute for each $i = q, \ldots, 1$ successively

$$\mathbf{b}_{i-1}^{[1]} \quad = \quad \mathbf{b}_{i-1}^{[0]} + (\xi - t_{i-1})\, \mathbf{b}_i^{[1]}.$$

Scheme:

$$\begin{array}{ccccccc}
\mathbf{b}_q = \mathbf{b}_q^{[0]} & \mathbf{b}_{q-1} = \mathbf{b}_{q-1}^{[0]} & \mathbf{b}_{q-2} = \mathbf{b}_{q-2}^{[0]} & \ldots & \mathbf{b}_2 = \mathbf{b}_2^{[0]} & \mathbf{b}_1 = \mathbf{b}_1^{[0]} & \mathbf{b}_0 = \mathbf{b}_0^{[0]} \\
 & +(\xi - t_{q-1})\, \mathbf{b}_q^{[1]} & +(\xi - t_{q-2})\, \mathbf{b}_{q-1}^{[1]} & \ldots & +(\xi - t_2)\, \mathbf{b}_3^{[1]} & +(\xi - t_1)\, \mathbf{b}_2^{[1]} & +(\xi - t_0)\, \mathbf{b}_1^{[1]} \\
\hline
\mathbf{b}_q^{[1]} & \mathbf{b}_{q-1}^{[1]} & \mathbf{b}_{q-2}^{[1]} & \ldots & \mathbf{b}_2^{[1]} & \mathbf{b}_1^{[1]} & \mathbf{b}_0^{[1]} = \mathbf{P}(\xi)
\end{array}$$

Cost: $\mathcal{O}(q)$

Complete modified Horner scheme

Iterate

$\mathbf{b}_q = \mathbf{b}_q^{[0]}$	$\mathbf{b}_{q-1} = \mathbf{b}_{q-1}^{[0]}$	$\mathbf{b}_{q-2} = \mathbf{b}_{q-2}^{[0]}$	$\ldots$	$\mathbf{b}_2 = \mathbf{b}_2^{[0]}$	$\mathbf{b}_1 = \mathbf{b}_1^{[0]}$	$\mathbf{b}_0 = \mathbf{b}_0^{[0]}$
$\mathbf{b}_q^{[1]}$	$\mathbf{b}_{q-1}^{[1]}$	$\ldots$	$\ldots$	$\mathbf{b}_2^{[1]}$	$\mathbf{b}_1^{[1]}$	$\mathbf{b}_0^{[1]} = \mathbf{P}(\xi)$
$\mathbf{b}_q^{[2]}$	$\mathbf{b}_{q-1}^{[2]}$	$\ldots$	$\ldots$	$\mathbf{b}_2^{[2]}$	$\mathbf{b}_1^{[2]} = \mathbf{P}'(\xi)$	
$\mathbf{b}_q^{[3]}$	$\mathbf{b}_{q-1}^{[3]}$	$\ldots$	$\ldots$	$\mathbf{b}_2^{[3]} = \frac{\mathbf{P}''(\xi)}{2}$		
$\vdots$						
$\mathbf{b}_q^{[q]}$	$\mathbf{b}_{q-1}^{[q]} = \frac{\mathbf{P}^{(q-1)}(\xi)}{(q-1)!}$					

$$\mathbf{b}_q^{[q+1]} = \frac{\mathbf{P}^{(q)}(\xi)}{q!}$$

Determine $\mathbf{P}(\xi), \mathbf{P}'(\xi), \ldots, \mathbf{P}^{(q)}(\xi)$ with total expenditure of $\mathcal{O}(q^2)$

Complete modified Horner scheme Example

Suppose polynomial $p\colon [-1,2] \to \mathbb{R}$ of degree 3 is given by $p(t) = 1 + \frac{1}{2}\,(t+1)t + \frac{5}{6}\,(t+1)t(t-1)$

$t_0 = -1,\ t_1 = 0,\ t_2 = 1,\ t_3 = 2,$ Evaluate at $\xi = 0$

$$
\begin{array}{l|cccc}
b_i^{[0]},\ i = 3,2,1,0: & \frac{5}{6} & \frac{1}{2} & 0 & 1 \\
 & & (0-1)\,\frac{5}{6} & (0-0)\,\left(-\frac{1}{3}\right) & (0+1)\cdot 0 \\
\hline
b_i^{[1]},\ i = 3,2,1: & \frac{5}{6} & -\frac{1}{3} & 0 & \mathbf{1} = \frac{p(0)}{0!} \\
 & & (0-0) - \frac{5}{6} & (0+1)\,\left(-\frac{1}{3}\right) & \\
\hline
b_i^{[2]},\ i = 3,2: & \frac{5}{6} & -\frac{1}{3} & -\mathbf{\frac{1}{3}} = \frac{p'(0)}{1!} & \\
 & & (0+1)\,\frac{5}{6} & & \\
\hline
b_3^{[3]}: & \frac{5}{6} & \mathbf{\frac{1}{2}} = \frac{p''(0)}{2!} & & \\
\hline
b_3^{[4]}: & \mathbf{\frac{5}{6}} = \frac{p'''(0)}{3!} & & &
\end{array}
$$

Thus, $p(t) = \mathbf{1} - \mathbf{\frac{1}{3}}\,t + \mathbf{\frac{1}{2}}\,t^2 + \mathbf{\frac{5}{6}}\,t^3$

Representation of Polynomial Curves

Monomial , (Taylor) , Lagrange , Newton representation of **P**

$\mathbf{a}_i = \frac{\mathbf{P}^{(i)}(0)}{i!}$, ($\mathbf{a}_i = \frac{\mathbf{P}^{(i)}(\xi)}{i!}$) , $\mathbf{P}(t_i)$, $\mathbf{b}_i = \Delta(t_0, \ldots, t_i)$

Shortcomings (computer graphics):

Monomial , (Taylor) , Lagrange , Newton representation of **P**

$\mathbf{a}_i = \frac{\mathbf{P}^{(i)}(0)}{i!}$, ($\mathbf{a}_i = \frac{\mathbf{P}^{(i)}(\xi)}{i!}$) , $\mathbf{P}(t_i)$, $\mathbf{b}_i = \Delta(t_0, \ldots, t_i)$

Shortcomings (computer graphics):

- difficult geometric interpretation • •
- not well-aligned with parameter interval $[a, b]$, asymmetrical • • •
- (occasionally) unstable • • •
- sub-optimal evaluation •

Representation of Polynomial Curves

Monomial , (Taylor) , Lagrange , Newton representation of **P**

$\mathbf{a}_i = \frac{\mathbf{P}^{(i)}(0)}{i!}$, ($\mathbf{a}_i = \frac{\mathbf{P}^{(i)}(\xi)}{i!}$) , $\mathbf{P}(t_i)$, $\mathbf{b}_i = \Delta(t_0, \ldots, t_i)$

Shortcomings (computer graphics):

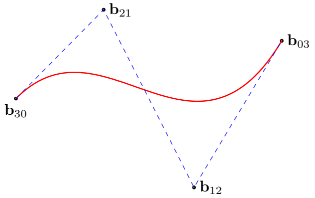
- difficult geometric interpretation ● ●
- not well-aligned with parameter interval $[a, b]$, asymmetrical ● ● ●
- (occasionally) unstable ● ● ●
- sub-optimal evaluation ●
- . . . ● ● ●

Consequently: Bernstein-Bézier representation of polynomial curve **P**

Bernstein-Bézier Representation (B-Form)

$$\mathbf{P}(t) = \sum_{i+j=q} B_{ij}(t)\,\mathbf{b}_{ij},\ t \in [a,b]$$

$\mathbf{b}_{ij} \in \mathbb{R}^d$ **Bernstein coefficients**: control points



a cubic Bézier curve

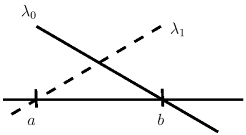
$\mathbf{b}_{q0}, \mathbf{b}_{q-1\,1}, \ldots, \mathbf{b}_{0q}$ form a path: control polygon

$B_{ij},\ i+j=q$: **Bernstein polynomials**

$$B_{ij}(t) = \tfrac{q!}{i!j!}\,\lambda_0^i(t)\lambda_1^j(t)$$

λ_0, λ_1: **barycentric coordinates**

Barycentric coordinates

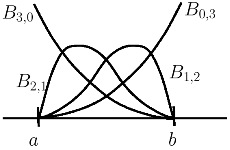


$$\lambda_0(t) = \frac{b-t}{b-a}, \ \lambda_1(t) = \frac{t-a}{b-a}, \ t \in \mathbb{R}$$

Mathematical properties:

- $t \in [a, b]$ if and only if $\lambda_0(t) \geq 0$ and $\lambda_1(t) \geq 0$
- $\lambda_0(t) + \lambda_1(t) = 1$ and $a \, \lambda_0(t) + b \, \lambda_1(t) = t$
- For all polynomial curves $\mathbf{P}$ of degree 1 (linear, straight lines): $\lambda_0(t) \, \mathbf{P}(a) + \lambda_1(t) \, \mathbf{P}(b) = \mathbf{P}(t)$

Bernstein Polynomials



$$B_{ij}(t) = \frac{q!}{i!j!}\,\lambda_0^i(t)\lambda_1^j(t) = \frac{q!}{i!j!}\,(\frac{b-t}{b-a})^i(\frac{t-a}{b-a})^j$$

$q+1$ distinct curves ($i+j=q$) – basis of polynomials (degree q)

Mathematical properties:

- $B_{ij}(t) > 0,\ t \in (a,b)$ (positivity)
- $B_{ij}(t) = 0$ if and only if ($t = a$ and $i \neq q$) or ($t = b$ and $i \neq 0$) (roots)
- $B_{ij}(t) = B_{ji}(a+b-t)$ (symmetry)

Bernstein Polynomials – additional properties

TECHNISCHE UNIVERSITÄT DARMSTADT

Important for geometry of B-form (. . . in a few minutes)

$$\sum_{i+j=q} B_{ij}(t) = 1 \text{ and } \sum_{i+j=q} (\tfrac{i}{q}\, a + \tfrac{j}{q}\, b)\, B_{ij}(t) = t$$

For all polynomial curves **P** of degree 1 (linear, straight lines): $\sum_{i+j=q} \mathbf{P}(\tfrac{i}{q}\, a + \tfrac{j}{q}\, b)\, B_{ij}(t) = \mathbf{P}(t)$

Recursive definition: $B_{ij}(t) = \lambda_0(t) \cdot B_{i-1\,j}(t) + \lambda_1(t) \cdot B_{i\,j-1}(t)$

Derivative: $B'_{ij}(t) = \frac{q}{b-a}\,(\, B_{i\,j-1}(t) - B_{i-1\,j}(t)\,)$

Maxima: (in the *domain points*) $\max\{B_{ij}(t) :\ t \in [a,b]\} = B_{ij}(\tfrac{i}{q}\, a + \tfrac{j}{q}\, b)$

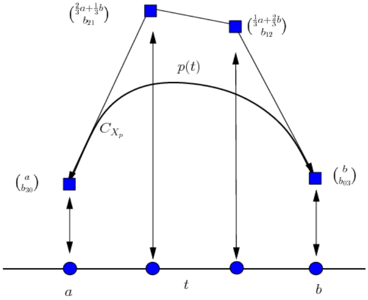
Functional Bernstein-Bézier Representation

$$p(t) = \sum_{i+j=q} b_{ij}\, B_{ij}(t),\ t \in [a,b],\ b_{ij} \in \mathbb{R}$$

Graph $C_{\mathbf{X}_p}$ of p is given by

$$\mathbf{X}_p(t) = \begin{pmatrix} t \\ p(t) \end{pmatrix}$$

$$= \sum_{i+j=q} B_{ij}(t) \begin{pmatrix} \frac{i}{q}\, a + \frac{j}{q}\, b \\ b_{ij} \end{pmatrix}$$



Note: domain points $\xi_{ij} = \frac{i}{q}\, a + \frac{j}{q}\, b,\ i+j=q$, are uniquely associated with b_{ij}.

B-Form: Control Points & Polygon

$$\mathbf{P}(t) = \sum_{i+j=q} B_{ij}(t)\,\mathbf{b}_{ij},\ t \in [a,b]$$

$\mathbf{b}_{ij} \in \mathbb{R}^d$ “control” the geometry of $\mathbf{P}$.

End point interpolation

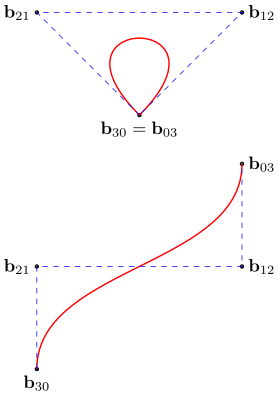
- $\mathbf{P}(a) = \mathbf{b}_{q0}$ and $\mathbf{P}(b) = \mathbf{b}_{0q}$.

Affine invariance

- is a consequence of the partition of unity $\sum B_{ij} = 1$.

Convex hull property

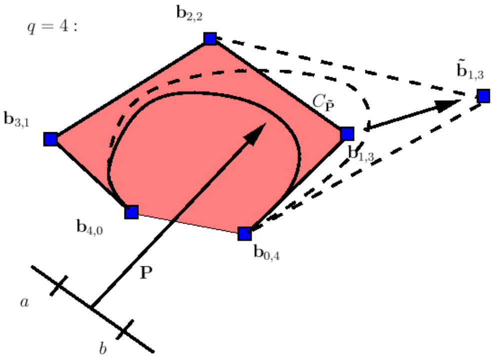
- due to $B_{ij} \geq 0$ for $t \in [a,b]$, the curve is either inside or intersects the control polygon



Examples of cubic Bézier curves

Convex Hull and Pseudo Locality

Relocation of a single control point
changes the geometry of the curve primarily at the local parts near the control point



quartic Bézier curve

Derivative of B-Form

TECHNISCHE UNIVERSITÄT DARMSTADT

Algorithm: Determine ℓ-th derivative of P

Set $\widehat{\mathbf{b}}_{ij}^{[0]} = \mathbf{b}_{ij},\ i+j=q$ and compute for $\ell = 1,\ldots,q$ successively

$$\widehat{\mathbf{b}}_{ij}^{[\ell]} = \widehat{\mathbf{b}}_{i\,j+1}^{[\ell-1]} - \widehat{\mathbf{b}}_{i+1\,j}^{[\ell-1]},\ i+j=q-\ell\,.$$

Applies for $\ell \in \{0,\ldots,q\}$: $\mathbf{P}^{(\ell)} = \frac{q!}{(q-\ell)!}\,\frac{1}{(b-a)^\ell} \sum\limits_{i+j=q-\ell} B_{ij}\,\widehat{\mathbf{b}}_{ij}^{[\ell]}$

In particular, it holds (derivative at $\mathbf{P}(a)$):

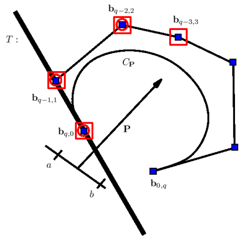
$$\mathbf{P}^{(\ell)}(a) = \frac{q!}{(q-\ell)!}\,\frac{1}{(b-a)^\ell}\,\widehat{\mathbf{b}}_{q-\ell\,0}^{[\ell]} = \frac{q!}{(q-\ell)!}\,\frac{1}{(b-a)^\ell}\sum_{k=0}^{\ell}(-1)^{\ell-k}\binom{\ell}{k}\,\mathbf{b}_{q-k,k}$$

Derivative of B-Form

Example ($q = 3$):

$$\begin{aligned}
\mathbf{P}(t) &= B_{30}(t)\,\mathbf{b}_{30} + B_{21}(t)\,\mathbf{b}_{21} + B_{12}(t)\,\mathbf{b}_{12} + B_{03}(t)\,\mathbf{b}_{03} \\
\mathbf{P}'(t) &= \frac{3}{b-a}\,\big(\,B_{20}(t)\,(\mathbf{b}_{21} - \mathbf{b}_{30}) + B_{11}(t)\,(\mathbf{b}_{12} - \mathbf{b}_{21}) + \\
&\qquad B_{02}(t)\,(\mathbf{b}_{03} - \mathbf{b}_{12})\,\big) \\
\mathbf{P}''(t) &= \frac{6}{(b-a)^2}\,\big(\,B_{10}(t)\,(\mathbf{b}_{12} - 2\,\mathbf{b}_{21} + \mathbf{b}_{30}) + \\
&\qquad B_{01}(t)\,(\mathbf{b}_{03} - 2\,\mathbf{b}_{12} + \mathbf{b}_{21})\,\big) \\
\mathbf{P}'''(t) &= \frac{6}{(b-a)^3}\,\big(\,\mathbf{b}_{03} - 3\,\mathbf{b}_{12} + 3\,\mathbf{b}_{21} - \mathbf{b}_{30}\,\big)
\end{aligned}$$

Derivative of B-Form



Control points $\mathbf{b}_{q,0},\ \mathbf{b}_{q-1,1},\ldots,\mathbf{b}_{q-\ell,\ell}$ determine $\mathbf{P}^{(\ell)}(a)$.

Derivatives $\mathbf{P}(a),\mathbf{P}'(a),\ldots,\mathbf{P}^{(\ell)}(a)$ determine $\mathbf{b}_{q-\ell,\ell}$ (**Hermite**!)

Tangents in end points of the curve (,e.g., a,):

$$T=\{\mathbf{z}\in\mathbb{R}^d:\ \mathbf{z}=\underbrace{\mathbf{P}(a)}_{\mathbf{b}_{q,0}}+\tau\underbrace{\mathbf{P}'(a)}_{\frac{q}{b-a}(\mathbf{b}_{q-1,1}-\mathbf{b}_{q,0})},\ \tau\in\mathbb{R}\}$$

Evaluation of P in B-Form

We seek to compute: $\mathbf{P}(\xi)$ for a certain $\xi \in \mathbb{R}$

de Casteljau algorithm

Set $\mathbf{b}_{ij}^{[0]} = \mathbf{b}_{ij},\ i+j=q$, and compute for $\ell = 1,\dots,q$, successively

$$\mathbf{b}_{ij}^{[\ell]} = \lambda_0(\xi)\,\mathbf{b}_{i+1,j}^{[\ell-1]} + \lambda_1(\xi)\,\mathbf{b}_{i,j+1}^{[\ell-1]},\ i+j=q-\ell\,.$$

$$\mathbf{P}(\xi) = \mathbf{b}_{00}^{[q]}$$

Idea of proof: (Inductive contention, Recursion of B-Polynomials, index-shift)

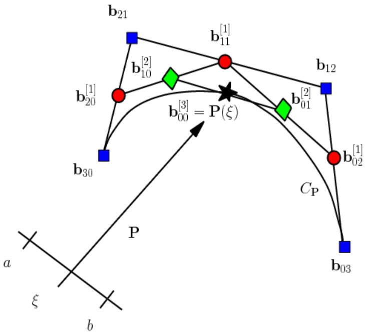
$$\begin{aligned}
\mathbf{P}(\xi) &= \sum_{i+j=q} B_{ij}(\xi)\,\mathbf{b}_{ij} = \sum_{i+j=q} (\lambda_0(\xi)\,B_{i-1,j}(\xi) + \lambda_1(\xi)\,B_{i,j-1}(\xi))\,\mathbf{b}_{ij} \\
&= \sum_{i+j=q-1} B_{ij}(\xi)\,(\lambda_0(\xi)\,\mathbf{b}_{i+1,j}^{[0]} + \lambda_1(\xi)\,\mathbf{b}_{i,j+1}^{[0]}) = \sum_{i+j=q-1} B_{ij}(\xi)\,\mathbf{b}_{ij}^{[1]}\,.
\end{aligned}$$

For the polynomial curve of degree $q-1$, one can obtain $\widetilde{\mathbf{P}}(\xi) = \mathbf{b}_{00}^{[q]}$ using de Casteljau and it holds that $\mathbf{P}(\xi) = \widetilde{\mathbf{P}}(\xi)$.

de Casteljau Algorithm

Example: $q = 3$

$$\mathbf{P} \equiv B_{30}\, \mathbf{b}_{30} + B_{21}\, \mathbf{b}_{21} + B_{12}\, \mathbf{b}_{12} + B_{03}\, \mathbf{b}_{03}$$



de Casteljau Algorithm – Triangle Scheme

$$
\begin{array}{ccccccccc}
\mathbf{b}^{[0]}_{q0} & & & & & & & & \\
 & > & \mathbf{b}^{[1]}_{q-1\,0} & & & & & & \\
\mathbf{b}^{[0]}_{q-1\,1} & & & > & \mathbf{b}^{[2]}_{q-2\,0} & & & & \\
 & > & \mathbf{b}^{[1]}_{q-2\,1} & & & \ddots & & & \\
\mathbf{b}^{[0]}_{q-2\,2} & & & \ddots & & & \mathbf{b}^{[q-1]}_{10} & & \\
 & & & & & & & > & \mathbf{b}^{[q]}_{00} = \mathbf{P}(\xi) \\
\vdots & & \vdots & & & & \mathbf{b}^{[q-1]}_{01} & & \\
\mathbf{b}^{[0]}_{1\,q-1} & & & > & \mathbf{b}^{[2]}_{0\,q-2} & & & & \\
 & > & \mathbf{b}^{[1]}_{0\,q-1} & & & & & & \\
\mathbf{b}^{[0]}_{0\,q} & & & & & & & &
\end{array}
$$

- Cost is $\mathcal{O}(q^2)$.
- For $\xi \in [a,b]$: convex combinations, therefore numerically stable.
- In addition, it provides the derivatives $\mathbf{P}'(\xi),\ \mathbf{P}''(\xi), \ldots,\ \mathbf{P}^{(q)}(\xi)$.
  - i-th derivative (in general): $\frac{1}{(b-a)^i}\,\frac{q!}{(q-i)!}\sum_{k=0}^{i}\binom{i}{k}(-1)^{i-k}\cdot b^{[q-i]}_{i-k,k}$
  - ,e.g., $\mathbf{P}'(\xi) = \frac{q}{b-a}\cdot(b^{[q-1]}_{01} - b^{[q-1]}_{10})$, $\mathbf{P}''(\xi) = \frac{q(q-1)}{(b-a)^2}\cdot(b^{[q-2]}_{02} - 2b^{[q-2]}_{11} + b^{[q-2]}_{20})$.

de Casteljau Change of B-Basis

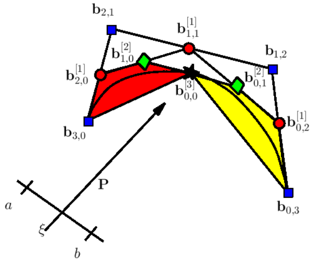
de Casteljau performs two changes of B-Bases !

$\mathbf{b}_{q,0}^{[0]}, \mathbf{b}_{q-1,0}^{[1]}, \ldots, \mathbf{b}_{1,0}^{[q-1]}, \mathbf{b}_{0,0}^{[q]}$ are control points of **P**'s B-Basis regarding $[a, \xi]$.

$\mathbf{b}_{00}^{[q]}, \mathbf{b}_{01}^{[q-1]}, \ldots, \mathbf{b}_{0\ q-1}^{[1]}, \mathbf{b}_{0q}^{[0]}$ are control points of **P**'s B-Basis regarding $[\xi, b]$

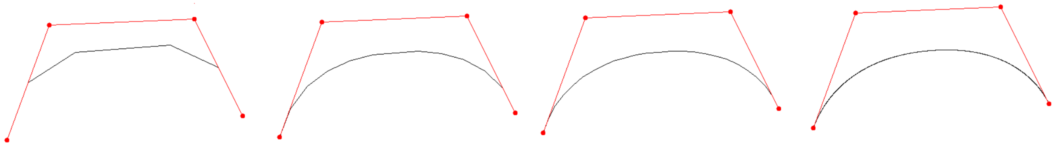
Example ($q = 3$):

$$\begin{aligned}
\mathbf{P} &= B_{30}\,\mathbf{b}_{30} + B_{21}\,\mathbf{b}_{21} + B_{12}\,\mathbf{b}_{12} + B_{03}\,\mathbf{b}_{03} \\
&= \widetilde{B}_{30}\,\mathbf{b}_{30} + \widetilde{B}_{21}\,\mathbf{b}_{20}^{[1]} + \widetilde{B}_{12}\,\mathbf{b}_{10}^{[2]} + \widetilde{B}_{03}\,\mathbf{b}_{00}^{[3]} \\
&= \widetilde{B}_{30}^{*}\,\mathbf{b}_{00}^{[3]} + \widetilde{B}_{21}^{*}\,\mathbf{b}_{01}^{[2]} + \widetilde{B}_{12}^{*}\,\mathbf{b}_{02}^{[1]} + \widetilde{B}_{03}^{*}\,\mathbf{b}_{03}
\end{aligned}$$



Application (examplatory) of B-Basis Change

Display of polynomial curve in B-Form (3 –4 recursion steps) through Subdivision using de Casteljau:



v.l.n.r.: 1,2,3,4 recursion steps; control polygon converge towards the curve

Lane & Riesenfeld 1980: quadratic convergence of control polygon $\mathcal{K}(\mathbf{P})$ towards the curve $\mathbf{P}$

Another application: **Bézier-Clipping**; approximate intersection point computation of polynomial curves through intersecting the (refined) control polygons.

Online Literature

TECHNISCHE UNIVERSITÄT DARMSTADT

- G. Farin: *Curves and Surfaces for CAGD*, https://books.google.com/books?id=5HYTP1dIAp4C&printsec=frontcover&hl=de&source=gbs_ViewAPI
- Prautzsch, Boehm & Paluszny: *Bézier and B-Spline techniques*, https://books.google.com/books?id=xP7A8F6NZGQC&printsec=frontcover&hl=de&source=gbs_ViewAPI
- Foley, van Dam, Feiner, Hughes: *Computer Graphics: Principles and Practice*, p. 478–491 https://books.google.com/books?id=-4ngT05gmAQC&printsec=frontcover&hl=de&source=gbs_ViewAPI
- de Boor, B-form basics, SIAM, 1987, ftp://ftp.cs.wisc.edu/Approx/BBform.pdf
- Farin *Triangular Bernstein-Bézier patches*, J. CAGD 3 (1986), https://www.ljll.math.upmc.fr/frey/ftp/M5105/Farin%20G.,%20Triangular%20Bernstein%20Bezier%20patches.pdf