Computer Graphics II
3: Freeform Curves and Surfaces

Computer Graphics and
Multimedia Systems Group

University of Siegen

Structure of this Chapter

Structure of Chapter

- *Subsection 1: Bézier-curves (single polynomial curves)*
- *Subsection 2: Bézier-splines (piece-wise polynomial curves)*
- *Subsection 3: B-Spline curves (intrinsic piece-wise polynomial curves*
- *Subsection 4: Tensor product surfaces*
- *Subsection 5: Differential geometry for curves*

Motivation (Criteria for Modeling Techniques)

Geometric Models: *Elementary for all graphic 3D applications*

Main question: *How to efficiently build a desired model / shape?*

Criteria from the application viewpoint:

- *Problem-oriented methods (e.g. accuracy & freedom)*
- *Intuitive building technique $\Longrightarrow$ what are "good" control parameters*
- *Sensible follow-up handling /reprocessing*

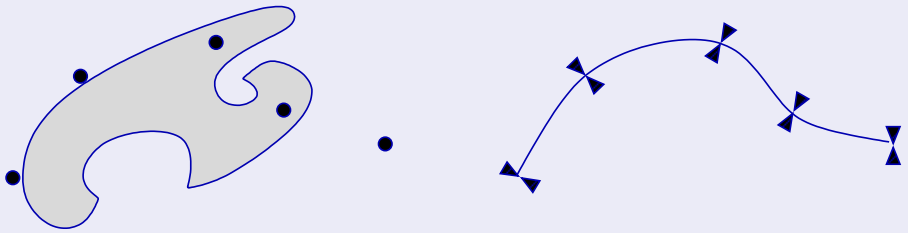
Criteria from an information technology viewpoint:

- *Geometry as a function of model parameters (internal representation)*
- *Geometry independance of **specific** control parameter change e.g. affine invariance)*
- *Representation of geometry for real-time-graphics or offline-rendering*

Remark

Origin of geometric modelling: *Industrial production in particular the manufacturing of automobiles*

Prior to 1960: *Various drawing methods as e.g. curve ruler or metal ruler.*



- *Inacurate transfer to real shapes for cast (dt: Guss), dies (dt: Stanzen) etc.*
- *Surfaces only manageable via curve arrays*

after 1960: *Drawing approaches transferred to computers*

- *Modelling of warped objects (curves, but also surfaces)*
- *Intuitive control parameters with high flexibility*
- *Efficiently computable on processors ($\longrightarrow$ **polynomials!**)*

Objective (Control of Polynomials)

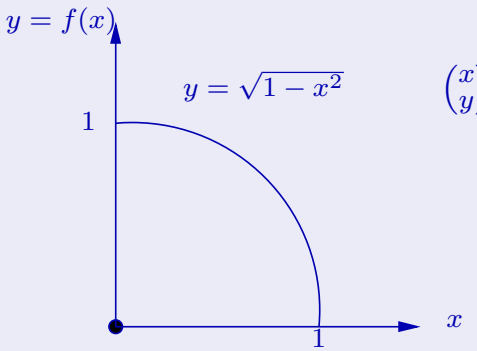
Arbitrary shape: *Function graphs are not enough!*

Intuitive form control: *Monomial-basis* $1, u, u^2, \ldots$ *does not work!*

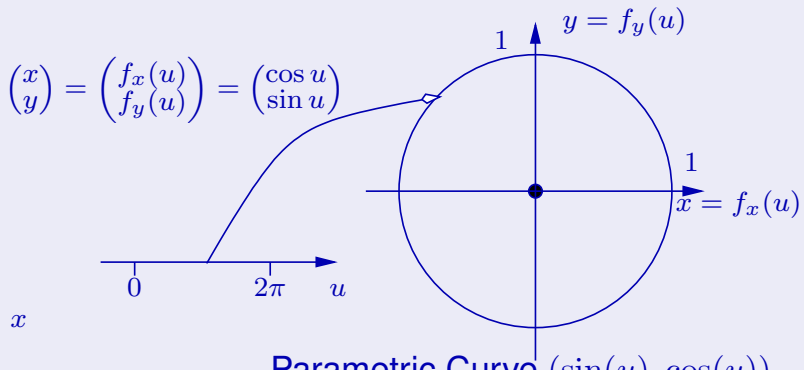
Remark (Function Graphs vs. Parametric Curves)

Graph: *Plot the dependant parameter* $y = f(x)$ *over the free parameter* x

Param. curve: *Representation of several dependant parameters* $x = f_x(u), y = f_y(u), \ldots$, *ignoring the free parameter* u



Graph $(x, \sqrt{1-x^2})$



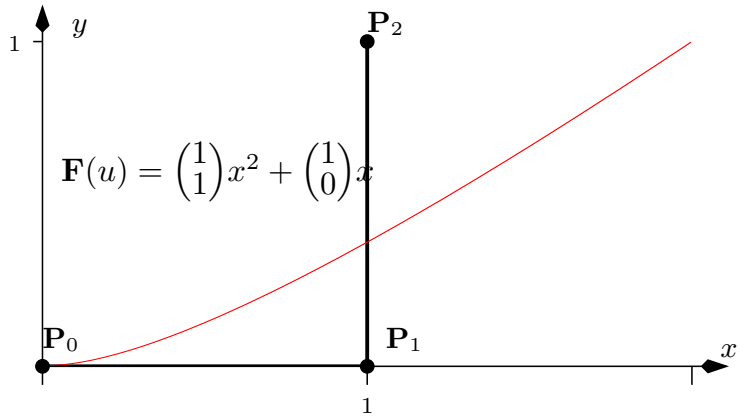
Parametric Curve $(\sin(u), \cos(u))$

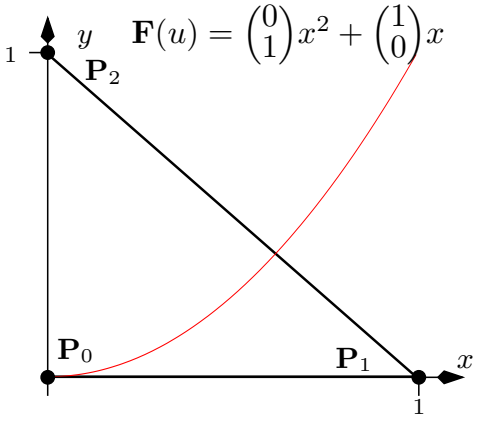
Control of Polynomials with Monomial-Basis

Direct utilization of the monomial-basis for 2D parametric curves of the degree n

$$\mathbf{F}(u) = \begin{pmatrix} f_x(u) \\ f_y(u) \end{pmatrix} = \sum_{i=0}^{n} \mathbf{P}_i u^i, \quad \text{mit } \mathbf{P}_i \in \mathbb{R}^2,\ i = 0, \ldots, n, u \in [0,1]$$

But no intuitive connection between the "parameter points" $\mathbf{P}_i$ and the curve shape:



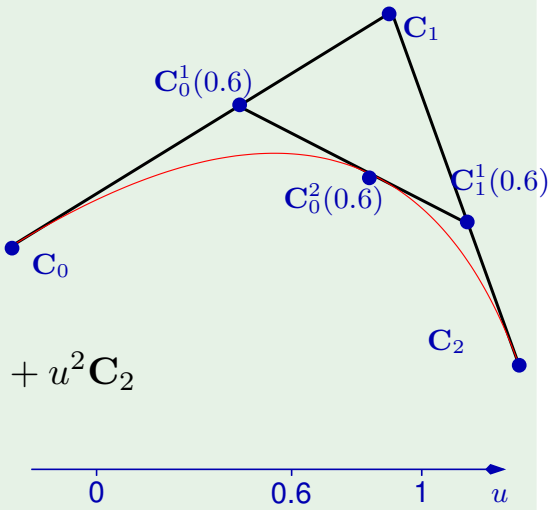


Example

Given: Three points: $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^2$

Construction: be $u \in [0,1]$, then bild

$$\begin{aligned} \mathbf{C}_0^1(u) &= (1-u)\mathbf{C}_0 + u\mathbf{C}_1 \\ \mathbf{C}_1^1(u) &= (1-u)\mathbf{C}_1 + u\mathbf{C}_2 \\ \mathbf{C}(u) = \mathbf{C}_0^2(u) &= (1-u)\mathbf{C}_0^1 + u\mathbf{C}_1^1 \\ &= (1-u)^2\mathbf{C}_0 + 2(1-u)u\mathbf{C}_1 + u^2\mathbf{C}_2 \end{aligned}$$



Properties of this construction:

1. $\mathbf{C}(u)$ is *affine combination* of $\mathbf{C}_i$, since $(1-u)^2 + 2(1-u)u + u^2 = ((1-u)+u)^2 = 1$
2. *Endpoint interpolation:* $\mathbf{C}_0^2(0) = \mathbf{C}_0,\ \mathbf{C}_0^2(1) = \mathbf{C}_2$
3. $\{\mathbf{C}(u) \ : \ u \in [0,1]\} \subset \Delta(\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2)$, i.e. curve lies in *convex hull* of the control points, as the weights are ≥ 0

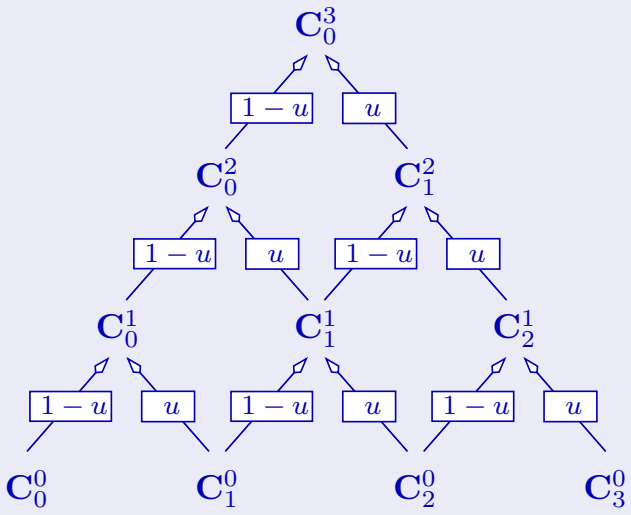
Algorithm (de Casteljau)

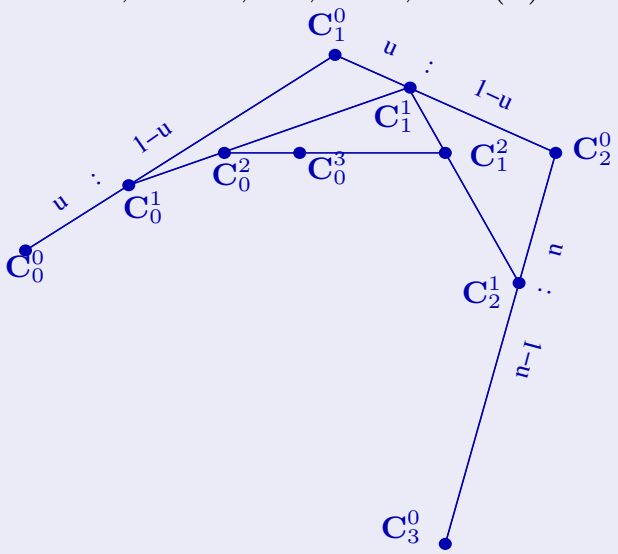
Given: *Control points* $\mathbf{C}_i,\ i = 0, \ldots, n$ *for* $n \in \mathbb{N}$, *and parameter* $u \in [0,1]$

Initialization: $\mathbf{C}_i^0 = \mathbf{C}_i, \quad i = 0, \ldots, n$

Recursion: *Affine-combinations of neighboring control points*

$$\mathbf{C}_i^{k+1} = (1-u)\mathbf{C}_i^k + u\mathbf{C}_{i+1}^k,\ i = 0, \ldots, n-k-1,\ k = 0, \ldots, n-1, \quad \mathbf{C}(u) = \mathbf{C}_0^n$$



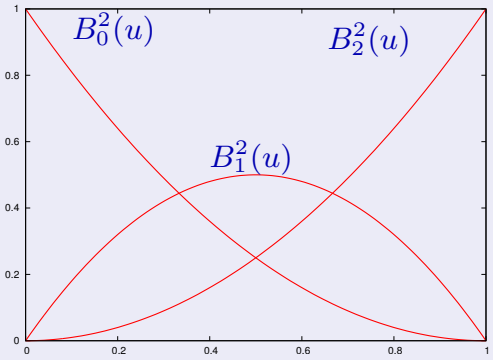


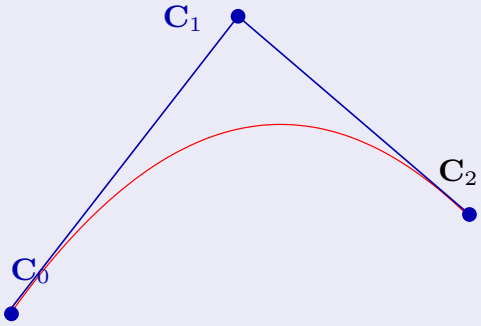
Bézier-Curves ($n = 2$)

Notation

Parabola: $\mathbf{C}(u) = \underbrace{(1-u)^2}_{=B_0^2(u)}\mathbf{C}_0 + \underbrace{2(1-u)u}_{=B_1^2(u)}\mathbf{C}_1 + \underbrace{u^2}_{=B_2^2(u)}\mathbf{C}_2$

Bernstein-Bézier-polynomials $B_i^2(u)$ *are the weights of the affine-combination of the* $\mathbf{C}_i$





Note: $B_i^2(u)$ *correspond to variable weights of an affine-combination as a function of* u

Bézier-Curves (General Case)

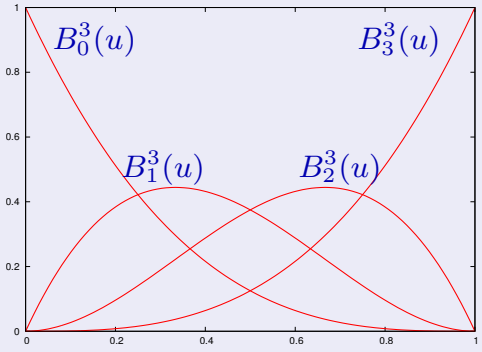
Notation

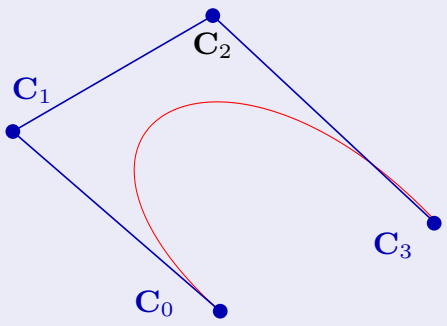
Bernstein-BézierPolynomials: *For arbitrary* $n \in \mathbb{N}$ *we define:*

$$\mathbf{C}(u) = \sum_{i=0}^{n} \mathbf{C}_i \; B_i^n(u), \text{ mit } B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i} \text{ und } \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Bézier-Curves: *For control points* $\mathbf{C}_0, \ldots, \mathbf{C}_n$ *we define:* $\mathbf{C}(u) = \sum_{i=0}^{n} B_i^n(u)\mathbf{C}_i$

Cubic Case $n = 3$**:** *Often Bézier-curves of degree* 3 *are used:*





Property (Bernstein-Bézier-Polynomials)

Polynomial-Basis: *Each polynomial* $\mathbf{P} \in \mathcal{P}^n$ *(*$\mathcal{P}^n$ *is space of the polynomials of degree* $\leq n$*) can be Bézier-curve represented: Let* $M_k(u) = u^k$*, we have*

$$\forall \mathbf{P} = \sum_{i=0}^{n} \mathbf{A}_i m_i : \ \exists_1 \ \mathbf{C}_i, \ i = 0, \ldots, n : \mathbf{P} = \sum_{i=0}^{n} \mathbf{C}_i B_i^n$$

Positivity: *For* $u \in [0,1]$ *always applies:* $B_i^n(u) \geq 0$

Endpoint interpolation: $B_i^n(0) = \begin{cases} 1 & \text{if } i = 0 \\ 0 & \text{else} \end{cases} \qquad B_i^n(1) = \begin{cases} 1 & \text{if } i = n \\ 0 & \text{else} \end{cases}$

Partition of unity: $1 = [u + (1-u)]^n = \sum_{i=0}^{n} \binom{n}{i} u^i (1-u)^{n-i} = \sum_{i=0}^{n} B_i^n(u)$

Derivation:

$$\left(B_i^n\right)'(u) = \begin{cases} -nB_0^{n-1}(u) & \text{if } i = 0 \\ nB_{n-1}^{n-1}(u) & \text{if } i = n \\ n\left(B_{i-1}^{n-1}(u) - B_i^{n-1}(u)\right) & \text{if } 0 < i < n \end{cases}$$

Property (Bézier-Curves)

Endpoint interpolation: $\mathbf{C}(0) = \mathbf{C}_0, \ \mathbf{C}(1) = \mathbf{C}_n$*, since* $B_0^n(0) = B_n^n(1) = 1$*, otherwise* $B_i^n(0) = B_i^n(1) = 0$

Global impact of $\mathbf{C}_i$**:** $\mathbf{C}_i$ *influences* $\mathbf{C}(u), \ u \in]0,1[$*, since here* $B_i^n(u) \neq 0, \ \forall i$

Affine Invariance: *The curve* $\mathbf{C}$ *is affine invariant with respect to the control points:*

$$T \text{ affine mapping} \quad \Rightarrow \quad T(\mathbf{C}(u)) = \sum_{i=0}^{n} T(\mathbf{C}_i) B_i^n(u) \quad \text{since} \quad \sum B_i^n \equiv 1$$

Convexe Hull: *Due to the positivity of* B_i^n *and the affine invariance* $\mathbf{C}(u), \ u \in [0,1]$ *lies within the convex hull of the control points*

Variation Diminishing Property (without proof): *The Bézier-curve has a maximum of inflections equal to it's control polygon:*

Case $n = 2$:$\forall$ *line* g *we have:* $\#\{g \cap$ *control polygon*$\} \geq \#\{g \cap$ *curve*$\}$

Case $n = 3$:$\forall$ *plane* E *we have:* $\#\{E \cap$ *control polygon*$\} \geq \#\{E \cap$ *curve*$\}$

Property (Bézier-Curves (cont.))

Derivation: *Bézier representation of the first derivative:*

$$\begin{aligned}\mathbf{C}'(u) &= n\left(-\mathbf{C}_0 B_0^{n-1}(u) + \mathbf{C}_n B_{n-1}^{n-1}(u) + \sum_{i=1}^{n-1} \mathbf{C}_i \left[B_{i-1}^{n-1}(u) - B_i^{n-1}(u)\right]\right) \\ &= n\sum_{i=0}^{n-1} \left(\mathbf{C}_{i+1} - \mathbf{C}_i\right) B_i^{n-1}(u)\end{aligned}$$

End-tangents (special case): *For $u \in \{0,1\}$ applies:*

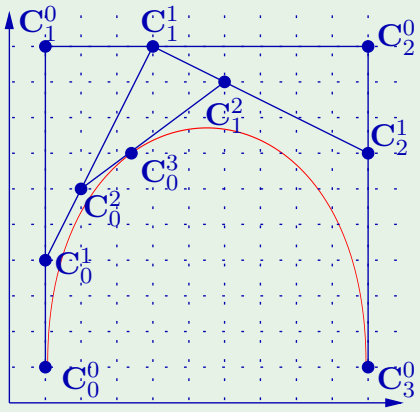
$$\mathbf{C}'(0) = n\left(\mathbf{C}_1 - \mathbf{C}_0\right) \qquad \mathbf{C}'(1) = n\left(\mathbf{C}_n - \mathbf{C}_{n-1}\right)$$

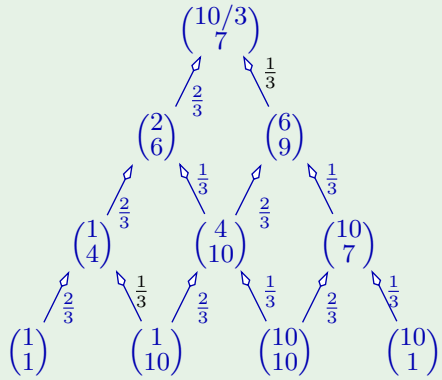
Reference to de Casteljau: *The last edge of the de Casteljau-scheme corresponds to the tangent direction of the curve:*

$$\mathbf{C}'(u) = n\left(\mathbf{C}_1^{n-1} - \mathbf{C}_0^{n-1}\right)$$

Example

Given: Points $\mathbf{C}_0 = \binom{1}{1}, \mathbf{C}_1 = \binom{1}{10}, \mathbf{C}_2 = \binom{10}{10}, \mathbf{C}_3 = \binom{10}{1}$. wanted: $\mathbf{C}(\frac{1}{3})$





Endpoints: $\mathbf{C}(0) = \mathbf{C}_0 = (1,1)^T, \quad \mathbf{C}(1) = \mathbf{C}_3 = (10,1)^T$

Derivative: $\mathbf{C}'(\frac{1}{3}) = 3\left((6,9)^T - (2,6)^T\right) = (12,9)^T$ and

$$\mathbf{C}'(u) = 3\sum_{i=0}^{2}\left(\mathbf{C}_{i+1} - \mathbf{C}_i\right) B_i^2(u) = 3\left(\binom{0}{9} B_0^2(u) + \binom{9}{0} B_1^2(u) + \binom{0}{-9} B_2^2(u)\right)$$

Remark

Evaluation of de Casteljau:

- *Very stable (only interpolation, i.e. weights between* 0 *and* 1*)*
- *Complexity:* $\mathcal{O}(n^2)$ *per evaluation, more precisely*

$$\text{Vector additions: } \frac{(n+1)n}{2} \qquad \text{Scalar multiplications: } (n+1)n$$

Horner-Scheme: *Based on the polynomial curve* $\mathbf{F}(u) = \sum_{i=0}^{n} \mathbf{A}_i u^i \in \mathcal{P}^n$

$$\mathbf{F}(u) = \sum_{i=0}^{n} \mathbf{A}_i u^i = \mathbf{A}_0 + u(\mathbf{A}_1 + u(\mathbf{A}_2 + u(\mathbf{A}_3 \ldots + u(\mathbf{A}_{n-1} + u\mathbf{A}_n))) \ldots)$$

- *Less stable*
- *Complexity:* $\mathcal{O}(n)$*, more precisely:* n *scalar multiplications,* n *vector additions*

3.1.1: Rational Bézier-Curves

Motivation

So far: *Only polynomial curves are representable*

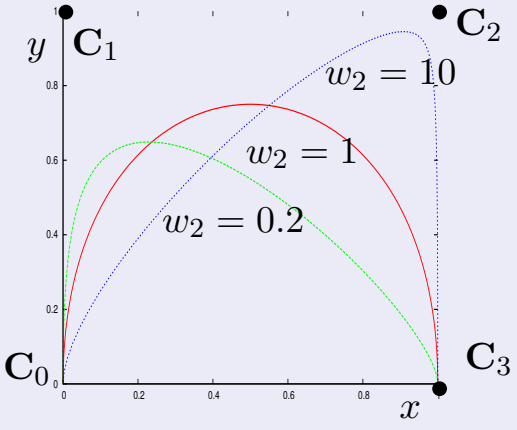
Aim: *Better curve control through the introduction of weights* w_i *for* $B_i^n(u)$

Important: *Affine invariance must be preserved* $\Longrightarrow$ *normalization of weights*

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n} \mathbf{C}_i \cdot w_i B_i^n(u)}{\sum_{i=0}^{n} w_i B_i^n(u)} = \sum_{i=0}^{n} \mathbf{C}_i \tilde{B}_i^n(u) \text{ with } \tilde{B}_i^n(u) = \frac{w_i B_i^n(u)}{\sum_{i=0}^{n} w_i B_i^n(u)}$$

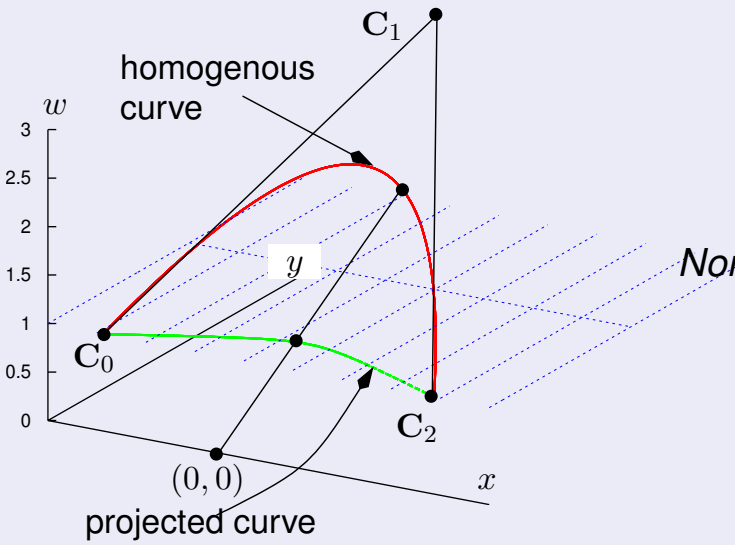
yields $\sum \tilde{B}_i^n(u) = 1$

Example: *Weight change at* $\mathbf{C}_2$



Remark (Alternative Interpretation)

Draw curve in homogenous coordinates *and project the curve onto* $w = 1$*:*



Homogenous description:

$$\mathbf{C}_i^h = \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix}$$

$$\mathbf{C}^h(u) = \sum_{i=0}^{n} \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix} B_i^n(u)$$

Normalization:

$$\mathbf{C}(u) = \begin{pmatrix} \mathbf{C}_x^h(u)/\mathbf{C}_w^h(u) \\ \mathbf{C}_y^h(u)/\mathbf{C}_w^h(u) \\ \mathbf{C}_z^h(u)/\mathbf{C}_w^h(u) \end{pmatrix} = \frac{\sum_{i=0}^{n} w_i \mathbf{C}_i B_i^n(u)}{\sum_{i=0}^{n} w_i B_i^n(u)}$$

Remark (Influence of Weights w_i)

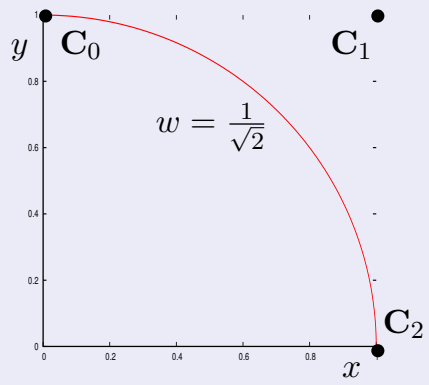
General: *The larger* w_i *and for unchanged* w_j, $j \neq i$ *the curve* $\mathbf{C}(u)$ *runs closer by* $\mathbf{C}_i$

Special case: *Conic sections* (dt: Kegelschnitte) *for* $n = 2$ *and* $w_0 = w_2 = 1$ *results by variation of* w_1 *in:*

$$\mathbf{C} \text{ is a } \left\{ \begin{array}{c} \text{ellipse segment} \\ \text{parabola segment} \\ \text{hyperbola segment} \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} w_1 < 1 \\ w_1 = 1 \\ w_1 > 1 \end{array} \right.$$

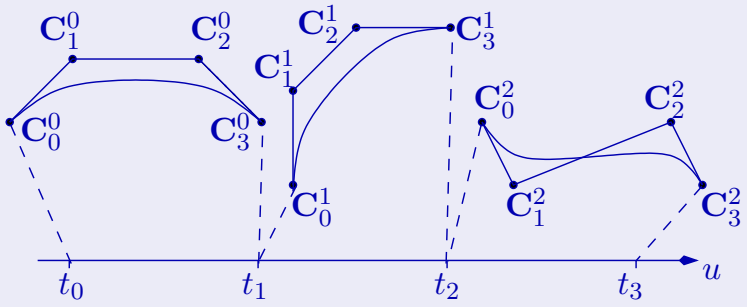
Special case: *Circle segment for* $n = 2$*:*

$w_0 = w_2 = 1$ *and* $w_1 = 1/\sqrt{2}$



Remark (Reparameterization using a Linear Function)

Goal: *Use parameter interval* $[a, b]$ *instead of* $[0, 1]$ *e.g. for piecewise curves*



Reparametrization: *Linear transformation* $\phi : I = [t_j, t_{j+1}] \to [0, 1]$ *does not change polynomial degree. Using* $\phi(u) = \frac{u-t_j}{\Delta_j}$, $\Delta_j = t_{j+1} - t_j$ *we get*

$$B_i^{I,n}(u) := B_i^n(\phi(u)) = \binom{n}{i} \left(\frac{u-t_j}{\Delta_j}\right)^i \left(1 - \frac{u-t_j}{\Delta_j}\right)^{n-i} = \binom{n}{i} \left(\frac{u-t_j}{\Delta_j}\right)^i \left(\frac{t_{j+1}-u}{\Delta_j}\right)^{n-i}$$

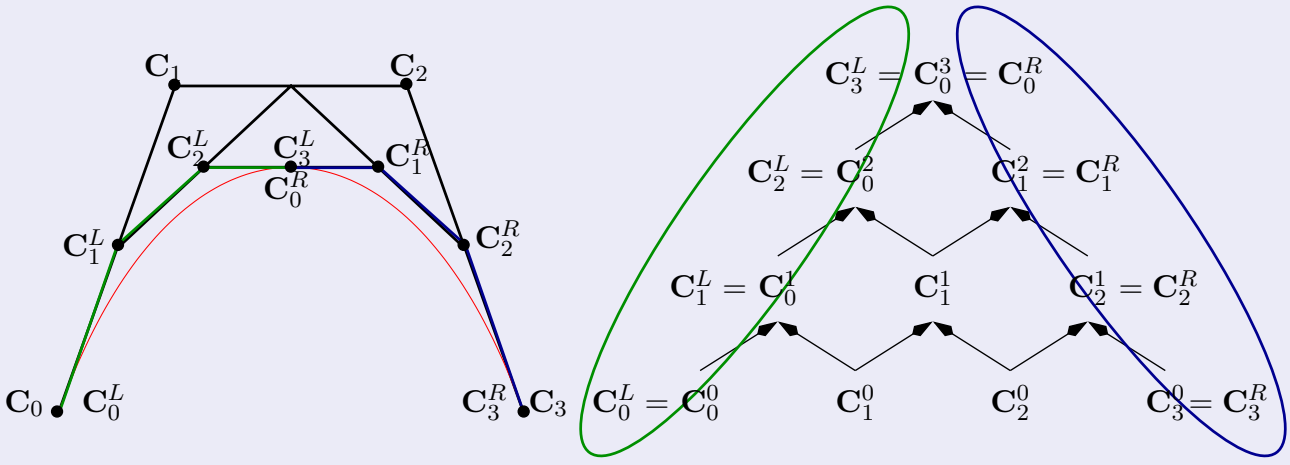
and $\left(B_i^{I,n}\right)'(u) = \phi'(u) \cdot \left(B_i^n\right)'(\phi(u)) = \frac{1}{\Delta_j} \cdot \left(B_i^n\right)'(\phi(u))$

Note: *The geometry of the curve is unchanged, but the derivative is scaled.*

Subdivision of Bézier-Curves

Remark (Subdivision of the Bézier-Curve $\mathbf{C}(u)$)

Evaluation *of* $\mathbf{C}$ *with* $a \in [0, 1]$ *delivers Bézier representation of the partial curves on* $[0, a]$ *and* $[a, 1]$*:*



Note: *The new Bézier curves with control points set* $\{\mathbf{C}_*^R\}, \{\mathbf{C}_*^L\}$ *describe sections of the same polynomial curve. However, changing e.g.* $\mathbf{C}_3^R$ *does not have any influence on the left segment.*

Summary

Essential Advantages:

- *Intuitive control of the shape of the curve*
- *Stabile calculation (de Casteljau)*
- *Affine invariance and convex hull properties*
- *Representation of all polynomials and conic sections*

Essential Disadvantages:

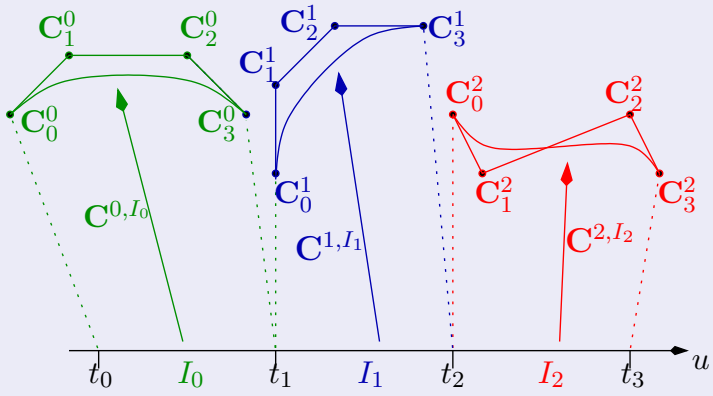
- *Global influence of the control points on the curve*
- *Only polynomial curves and conic sections, resp. (rational Bézier-curves)*
- *Number of degrees of freedom = number of control points = polynomial degree + 1*
  $\Longrightarrow$ more degrees of freedon requires a higher polynomial degree and thus more computation time.

Notation (Bézier-Spline Curves)

Approach: *More degrees of freedom by joining several Bézier-curves*

- *Knot vector* $T = \{t_0 < t_1 < \ldots < t_{k+1}\}$ *forming* $k+1$ *intervals* $I_j,\ j = 0, \ldots, k$
- $k+1$ *Bézier-segments* $\mathbf{C}^{j,I_j}(u),\ j = 0, \ldots, k$ *of the degree* n

Bézier Spline Definition: *For* $u \in [t_j, t_{j+1}]$ *the spline curve* $\mathbf{C}(u)$ *is defined via* $\mathbf{C}^{j,I_j}(u)$



Question: *Which conditions must be met for* $\mathbf{C}$ *to be a continuous and/or smooth curve at the joints?*

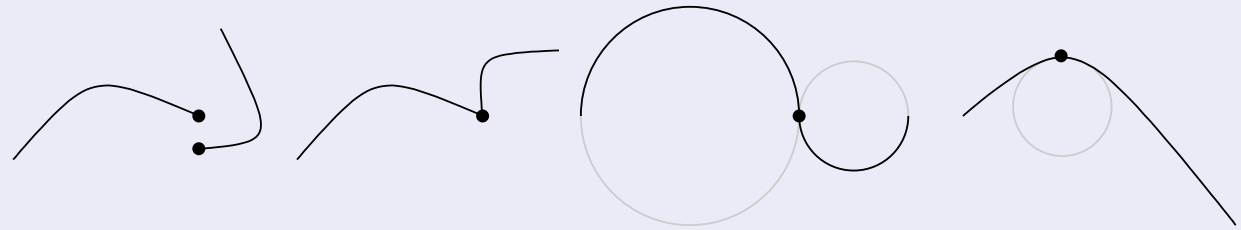
Definition (Continuous Differentiability / C^k Continuity)

Formal: A function $f(u)$ is in u_0 *C^k-continuous*, if it can be

1. in u_0 k-times differentiable and
2. the k-th derivation in u_0 is continuous

Specifically:

- C^{-1}: The curve is unsteady (step)
- C^0: The curve is continuous but cannot be differentiated (“cusp”)
- C^1: The curve is once differentiable with a continuous tangent
- C^2: The curve is twice differentiable with continuous 2nd derivation (curvature)



C^{-1} continuous | C^0 continuous | C^1 continuous but not C^2 | C^2 continuous

Property

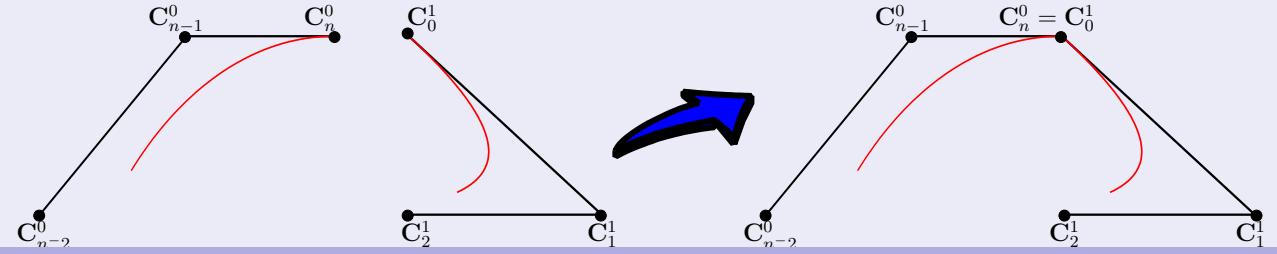
In the following *we focus on Bézier segments $\mathbf{C}^{0,I_o}, \mathbf{C}^{1,I_1}$ of degree n over the knot vector $T = \{t_0, t_1, t_2\},\ t_i < t_{i+1}$ and interval lenghts $\Delta_i = t_{i+1} - t_i = |I_i|$*

Furthermore: *We drop the interval when denoting the curve segments, i.e. $\mathbf{C}^i = \mathbf{C}^{i,I_i}$*

Given: *Curve $\mathbf{C}^0$ with control points $\mathbf{C}^0_0, \ldots, \mathbf{C}^0_n$*

Wanted: *Control points $\mathbf{C}^1_i$ of $\mathbf{C}^1$, so that $\mathbf{C}^1$ connects continuously to $\mathbf{C}^0$ in $\mathbf{C}^0(t_1)$*

C^0-continuity: $\mathbf{C}^0(t_1) = \mathbf{C}^1(t_1)$ *has to be valid, thus $\mathbf{C}^0_n = \mathbf{C}^1_0$ (endpoint interpolation)*



C^1 Continuity Condition for Bézier Splines

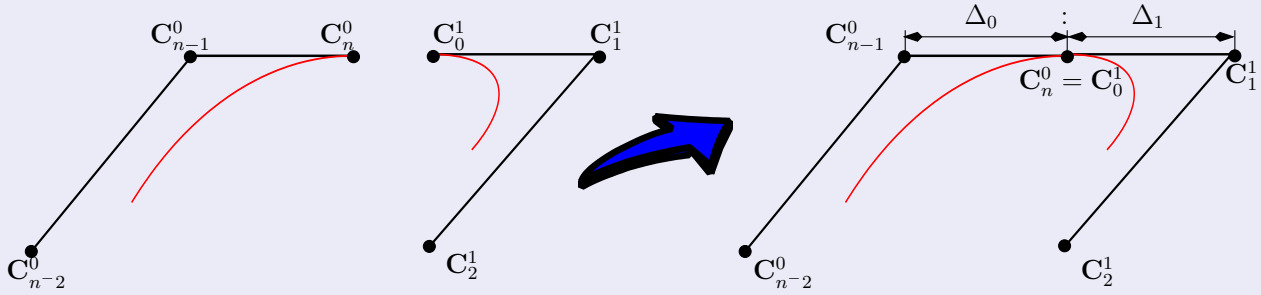
Property

Precondition: C^0-*continuity, thus* $\mathbf{C}_n^0 = \mathbf{C}_0^1$

Additionally must be given: $(\mathbf{C}^0)'(t_1) = (\mathbf{C}^1)'(t_1)$ *(interpolation of the end tangent)*

$$(\mathbf{C}^0)'(t_1) = \frac{n}{\Delta_0}\left(\mathbf{C}_n^0 - \mathbf{C}_{n-1}^0\right) \stackrel{!}{=} \frac{n}{\Delta_1}\left(\mathbf{C}_1^1 - \mathbf{C}_0^1\right) = (\mathbf{C}^1)'(t_1)$$

$$\stackrel{C^0-\text{continuous}}{\Leftrightarrow} \mathbf{C}_1^1 = \mathbf{C}_0^1 + \frac{\Delta_1}{\Delta_0}(\underbrace{\mathbf{C}_n^0}_{=\mathbf{C}_0^1} - \mathbf{C}_{n-1}^0)$$



C^2 Continuity Condition for Bézier Splines

Property

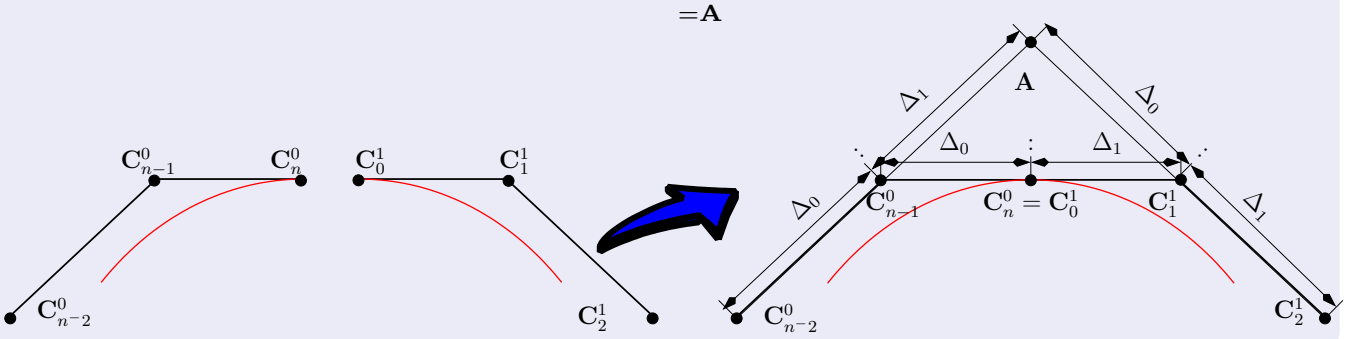
Precondition: C^1-*continuity, thus* $\mathbf{C}_n^0 = \mathbf{C}_0^1 = \frac{\Delta_1}{\Delta_0+\Delta_1}\mathbf{C}_{n-1}^0 + \frac{\Delta_0}{\Delta_0+\Delta_1}\mathbf{C}_1^1$

Additionally must be given: $(\mathbf{C}^0)''(t_1) = (\mathbf{C}^1)''(t_1)$, *more precisely*

$$(\mathbf{C}^I)''(u) = \frac{n(n-1)}{\Delta^2}\sum_{i=0}^{n-2}(\mathbf{C}_{i+2} - 2\mathbf{C}_{i+1} + \mathbf{C}_i)B_i^{n-2}(u) \text{ with } \Delta = |I|, \text{ thus:}$$

$$(\mathbf{C}^0)''(t_1) = (\mathbf{C}^1)''(t_1) \Leftrightarrow \frac{1}{(\Delta_0)^2}\left(\mathbf{C}_n^0 - 2\mathbf{C}_{n-1}^0 + \mathbf{C}_{n-2}^0\right) = \frac{1}{(\Delta_1)^2}\left(\mathbf{C}_2^1 - 2\mathbf{C}_1^1 + \mathbf{C}_0^1\right)$$

$$\stackrel{C^1-\text{cont.}}{\Leftrightarrow} \mathbf{C}_2^1 = \mathbf{C}_1^1 + \frac{\Delta_1}{\Delta_0}(\mathbf{C}_1^1 - \underbrace{(\mathbf{C}_{n-1}^0 + \frac{\Delta_1}{\Delta_0}(\mathbf{C}_{n-1}^0 - \mathbf{C}_{n-2}^0))}_{=\mathbf{A}})$$



$$
\begin{aligned}
\mathbf{C}_2^1 &= 2\mathbf{C}_1^1 - \mathbf{C}_0^1 + \frac{(\Delta_1)^2}{(\Delta_0)^2}\left(\mathbf{C}_n^0 - 2\mathbf{C}_{n-1}^0 + \mathbf{C}_{n-2}^0\right) \\
&= 2\mathbf{C}_1^1 - \left(\frac{\Delta_1}{\Delta_0+\Delta_1}\mathbf{C}_{n-1}^0 + \frac{\Delta_0}{\Delta_0+\Delta_1}\mathbf{C}_1^1\right) + \frac{(\Delta_1)^2}{(\Delta_0)^2}\left(\left(\frac{\Delta_1}{\Delta_0+\Delta_1}\mathbf{C}_{n-1}^0 + \frac{\Delta_0}{\Delta_0+\Delta_1}\mathbf{C}_1^1\right) - 2\mathbf{C}_{n-1}^0 + \mathbf{C}_{n-2}^0\right) \\
&= \left(2 - \frac{\Delta_0}{\Delta_0+\Delta_1} + \frac{\Delta_0(\Delta_1)^2}{(\Delta_0)^2(\Delta_0+\Delta_1)}\right)\mathbf{C}_1^1 + \left(-\frac{\Delta_1}{\Delta_0+\Delta_1} + \frac{(\Delta_1)^3}{(\Delta_0)^2(\Delta_0+\Delta_1)} - 2\frac{(\Delta_1)^2}{(\Delta_0)^2}\right)\mathbf{C}_{n-1}^0 + \frac{(\Delta_1)^2}{(\Delta_0)^2}\mathbf{C}_{n-1}^0 \\
&= \frac{2\Delta_0(\Delta_0+\Delta_1)-\Delta_0^2+(\Delta_1)^2}{\Delta_0(\Delta_0+\Delta_1)}\mathbf{C}_1^1 + \left(-\frac{\Delta_1}{\Delta_0+\Delta_1} + \frac{(\Delta_1)^3}{(\Delta_0)^2(\Delta_0+\Delta_1)} - 2\frac{(\Delta_1)^2}{(\Delta_0)^2}\right)\mathbf{C}_{n-1}^0 + \frac{(\Delta_1)^2}{(\Delta_0)^2}\mathbf{C}_{n-1}^0 \\
&= \frac{(\Delta_0+\Delta_1)^2}{\Delta_0(\Delta_0+\Delta_1)}\mathbf{C}_1^1 + \left(\frac{\Delta_1((\Delta_1)^2-(\Delta_0)^2)}{(\Delta_0)^2(\Delta_0+\Delta_1)} - 2\frac{(\Delta_1)^2}{(\Delta_0)^2}\right)\mathbf{C}_{n-1}^0 + \frac{(\Delta_1)^2}{(\Delta_0)^2}\mathbf{C}_{n-1}^0 \\
&= \left(1 + \frac{\Delta_1}{\Delta_0}\right)\mathbf{C}_1^1 + \left(\frac{\Delta_1(\Delta_1-\Delta_0)}{(\Delta_0)^2} - 2\frac{(\Delta_1)^2}{(\Delta_0)^2}\right)\mathbf{C}_{n-1}^0 + \frac{(\Delta_1)^2}{(\Delta_0)^2}\mathbf{C}_{n-1}^0 \\
&= \left(1 + \frac{\Delta_1}{\Delta_0}\right)\mathbf{C}_1^1 + \left(-\frac{\Delta_1}{\Delta_0} - \frac{(\Delta_1)^2}{(\Delta_0)^2}\right)\mathbf{C}_{n-1}^0 + \frac{(\Delta_1)^2}{(\Delta_0)^2}\mathbf{C}_{n-1}^0 \\
&= \mathbf{C}_1^1 + \frac{\Delta_1}{\Delta_0}(\mathbf{C}_1^1 - (\mathbf{C}_{n-1}^0 + \frac{\Delta_1}{\Delta_0}(\mathbf{C}_{n-1}^0 - \mathbf{C}_{n-2}^0)))
\end{aligned}
$$

Remark

- *Wherefore C^2: the human eye perceives C^1 as a step, e.g. picture frame:*

Approach (Interpolating Cubic, C^1-Continuous Bézier Splines)

Control Parameters: *Given a uniform knot vector* $t_j = j \cdot \Delta + t_0$ *with* $\Delta = 1$, *we interpolate points* $\mathbf{P}_j$ *and tangents* $\vec{\mathbf{t}}_j$ *at parameters* $t_j,\ j = 0, \ldots, k+1$

Idea: *Use cubic Bézier curves* $\mathbf{C}^j$ *over* $I_j = [t_j, t_{j+1}],\ j = 0, \ldots, k$

Problem: *Two different control types (points, tangents)*

Algorithm (Catmull-Rom-Splines)

Approach: *Use only control points as parameters and estimate tangents*

Calculation of Bézier points: *For the* j*-th Bézier-segment we get*

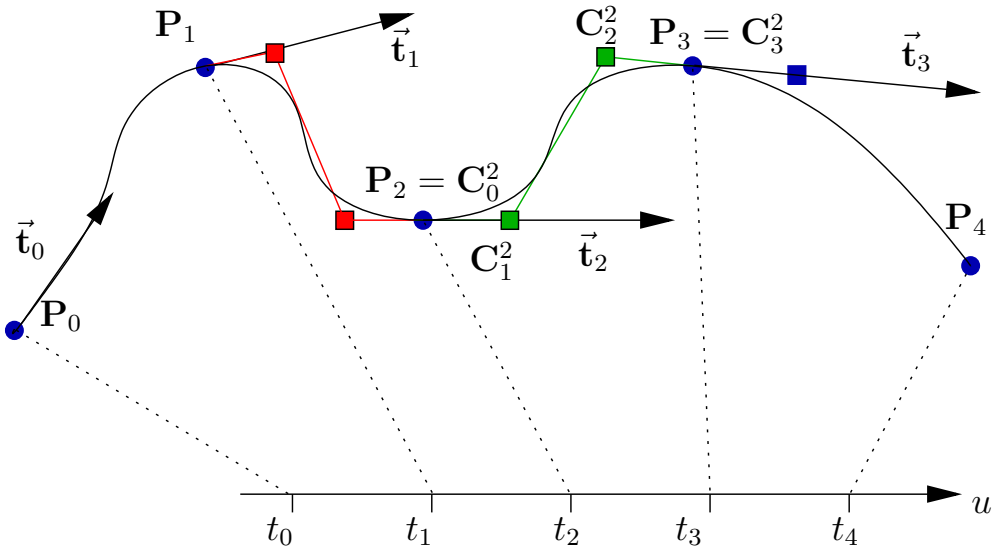
$$\text{End point: } \mathbf{C}_0^j = \mathbf{P}_j, \mathbf{C}_3^j = \mathbf{P}_{j+1}, \qquad j = 0, \ldots, k$$

$$\text{Tangent at } \mathbf{P}_j : \vec{\mathbf{t}}_j = \tfrac{1}{2}\left(\mathbf{P}_{j+1} - \mathbf{P}_{j-1}\right),\ j = 1, \ldots, k-1$$

$$\text{End tangent, e.g.: } \vec{\mathbf{t}}_0 = \tfrac{1}{2}\left(\mathbf{P}_1 - \mathbf{P}_0\right) \text{ and } \vec{\mathbf{t}}_{k+2} = \tfrac{1}{2}\left(\mathbf{P}_{k+2} - \mathbf{P}_{k+1}\right)$$

$$\text{Inner control points: } \vec{\mathbf{t}}_j = 3(\mathbf{C}_1^j - \mathbf{C}_0^j) \Rightarrow \mathbf{C}_1^j = \mathbf{C}_0^j + \tfrac{1}{3}\vec{\mathbf{t}}_j,$$

$$\text{analog: } \mathbf{C}_2^j = \mathbf{C}_3^j - \tfrac{1}{3}\vec{\mathbf{t}}_{j+1}$$



$$\text{Tangente at } \mathbf{P}_j : \quad \vec{\mathbf{t}}_j = \frac{1}{2}\left(\mathbf{P}_{j+1} - \mathbf{P}_{j-1}\right)$$

$$\text{Interpolation of endpoints: } \quad \mathbf{C}_0^j = \mathbf{P}_j,\ \mathbf{C}_3^j = \mathbf{P}_{j+1}$$

$$\text{Interpolation of tangent: } \quad \left(\mathbf{C}_1^j - \mathbf{C}_0^j\right) = \left(\mathbf{C}_3^{j-1} - \mathbf{C}_2^{j-1}\right) = \frac{1}{3}\vec{\mathbf{t}}_j$$

Summary

Essential advantages:

- *All advantages of the Bézier-curves*
- *Add. control points with fix polynom. degree*

Essential disadvantages:

- *Compliance with specific continuity conditions* **or**
- *Rather specific curve type (Catmull-Rom)*

We want to have: *"Built-in" continuity with*

Main Properties of Bézier-Splines 49

Summary

Essential advantages:

- *All advantages of the Bézier-curves*
- *Add. control points with fix polynom. degree*

Essential disadvantages:

- *Compliance with specific continuity conditions* **or**
- *Rather specific curve type (Catmull-Rom)*

We want to have: *"Built-in" continuity with*

- *Maximum continuity, i.e.* C^{n-1} *for degree* n
  - *Polynomial* $\mathbf{F} \in \mathcal{P}^n$ *of degree* n *is completely defined by* $\mathbf{F}(u_0), \ldots, \mathbf{F}^{(n)}(u_0)$ *(for any given* u_0*)*
  - *Enforcing* C^{n-1} *gives one additional degree of freedom (⇒ max. continuity)*

Summary

Essential advantages:

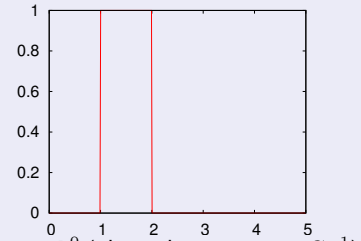
- *All advantages of the Bézier-curves*
- *Add. control points with fix polynom. degree*

Essential disadvantages:

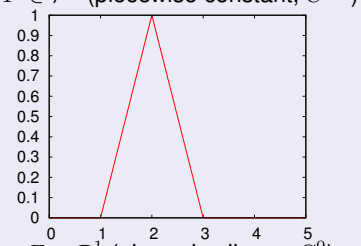
- *Compliance with specific continuity conditions* **or**
- *Rather specific curve type (Catmull-Rom)*

We want to have: *"Built-in" continuity with*

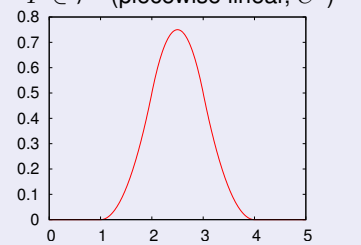
- *Maximum continuity, i.e.* C^{n-1} *for degree* n
  - *Polynomial* $\mathbf{F} \in \mathcal{P}^n$ *of degree* n *is completely defined by* $\mathbf{F}(u_0), \ldots, \mathbf{F}^{(n)}(u_0)$ *(for any given* u_0*)*
  - *Enforcing* C^{n-1} *gives one additional degree of freedom (⇒ max. continuity)*
- *Minimal influence of control points: Piecewise polynomials of degree* n *with continuity* C^{n-1} *span at least* $n+1$ *intervals*



$F \in \mathcal{P}^0$ (piecewise constant, C^{-1})



$F \in \mathcal{P}^1$ (piecewise linear, C^0)



$F \in \mathcal{P}^2$ (piecewise quadratic, C^1)

Approach (Construction of Quadratic B-Spline Curves)

Idea: *Use inner control points* $\mathbf{C}_1^i$ *of a* C^1 *continuous quadratic Bézier-spline as control points*

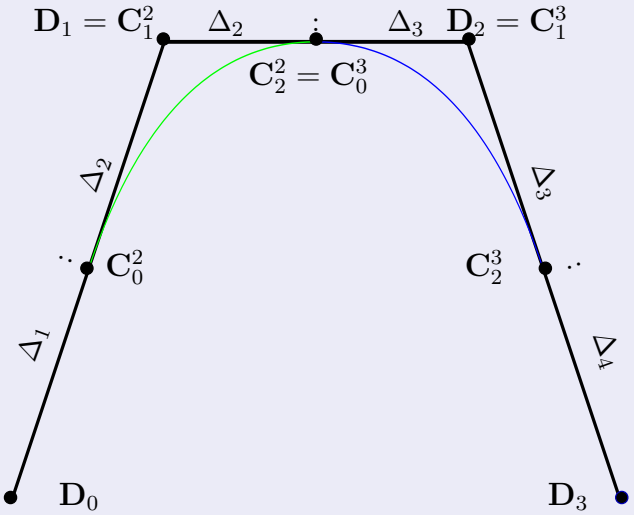
Given: *Knot vector* $T = \{t_0, \ldots, t_{m+3}\}$ *and de Boor points* $\mathbf{D}_0, \ldots, \mathbf{D}_m$

Construction: *Sequence* $\mathbf{D}_{j-2}, \ldots, \mathbf{D}_j$ *defines the j-th quadratic Bézier segment* $\mathbf{C}^j$, $j = 2, \ldots, m$ *over* $[t_j, t_{j+1}]$*:*

$$\mathbf{C}_0^j = \frac{\Delta_j}{\Delta_{j-1}+\Delta_j}\mathbf{D}_{j-2} + \frac{\Delta_{j-1}}{\Delta_{j-1}+\Delta_j}\mathbf{D}_{j-1}$$

$$\mathbf{C}_1^j = \mathbf{D}_{j-1}$$

$$\mathbf{C}_2^j = \frac{\Delta_{j+1}}{\Delta_j+\Delta_{j+1}}\mathbf{D}_{j-1} + \frac{\Delta_j}{\Delta_j+\Delta_{j+1}}\mathbf{D}_j$$



Note: *Automatic* C^1*-cont.*

Approach (Construction of Cubic B-Spline Curves)

Idea: *Use A-frame points of C^2 cubic Bézier-Splines as control points*

Given: *Knot vector $T = \{t_0, \ldots, t_{m+4}\}$ and de Boor points $\mathbf{D}_0, \ldots, \mathbf{D}_m$*

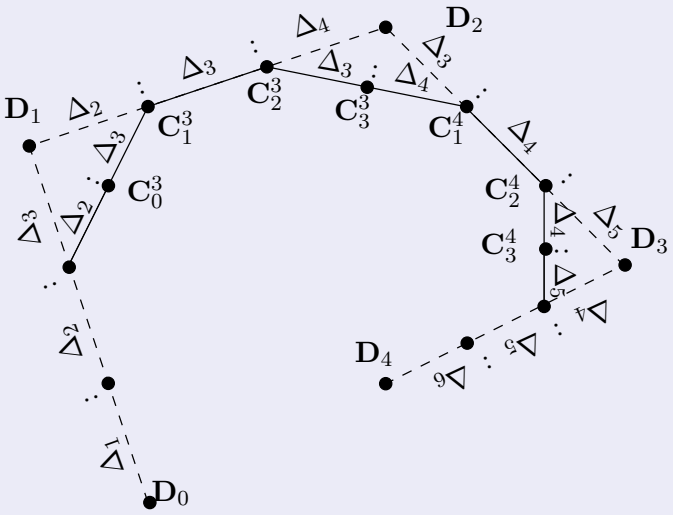
Construction: *Sequence $\mathbf{D}_{j-3}, \ldots, \mathbf{D}_j$ defines the j-th cubic Bézier segment $\mathbf{C}^j$ over $[t_j, t_{j+1}]$:*

$$\mathbf{C}_1^j = \frac{\Delta_j+\Delta_{j+1}}{\Delta_{j-1}+\Delta_j+\Delta_{j+1}}\mathbf{D}_{j-2} + \frac{\Delta_{j-1}}{\Delta_{j-1}+\Delta_j+\Delta_{j+1}}\mathbf{D}_{j-1}$$

$$\mathbf{C}_2^j = \frac{\Delta_{j+1}}{\Delta_{j-1}+\Delta_j+\Delta_{j+1}}\mathbf{D}_{j-2} + \frac{\Delta_{j-1}+\Delta_j}{\Delta_{j-1}+\Delta_j+\Delta_{j+1}}\mathbf{D}_{j-1}$$

$$\mathbf{C}_0^j = \frac{\Delta_j}{\Delta_{j-1}+\Delta_j}\mathbf{C}_2^{j-1} + \frac{\Delta_{j-1}}{\Delta_{j-1}+\Delta_j}\mathbf{C}_1^j$$

$$\mathbf{C}_3^j = \frac{\Delta_{j+1}}{\Delta_j+\Delta_{j+1}}\mathbf{C}_2^j + \frac{\Delta_j}{\Delta_j+\Delta_{j+1}}\mathbf{C}_1^{j+1}$$



Note: *Automatic C^2-continuity*

Objective (What is missing?)

- *Evaluation without detour via Bézier representation, and*
- *B-splines for arbitrary degrees n*

Observation (Direct Evaluation of Quadratic B-Splines)

Express *split ratio (dt: Teilungsverhältnis) with respect to neighboring intervals*

Evaluate *Bézier curve at $u \in [t_r, t_{r+1}$, denote points in the de Casteljau algorithm (segment r, level k) as $(\mathbf{C}_i^r)^k$*

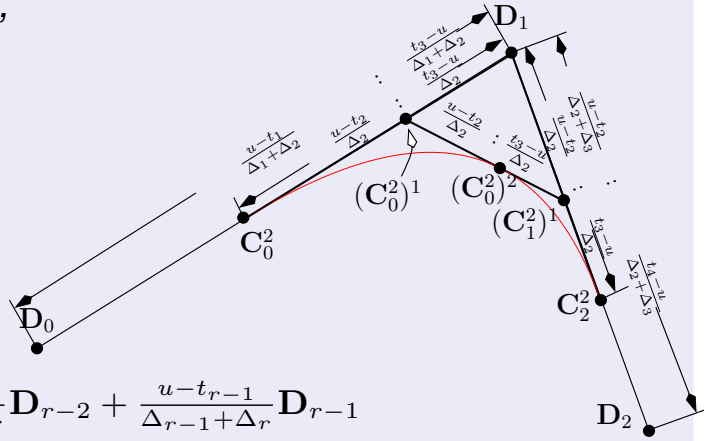
Since *we have $\mathbf{C}_1^r = \mathbf{D}_{r-1}$ and*

$$\mathbf{C}_0^r = \frac{\Delta_r}{\Delta_{r-1}+\Delta_r}\mathbf{D}_{r-2} + \frac{\Delta_{r-1}}{\Delta_{r-1}+\Delta_r}\mathbf{D}_{r-1}$$

we get

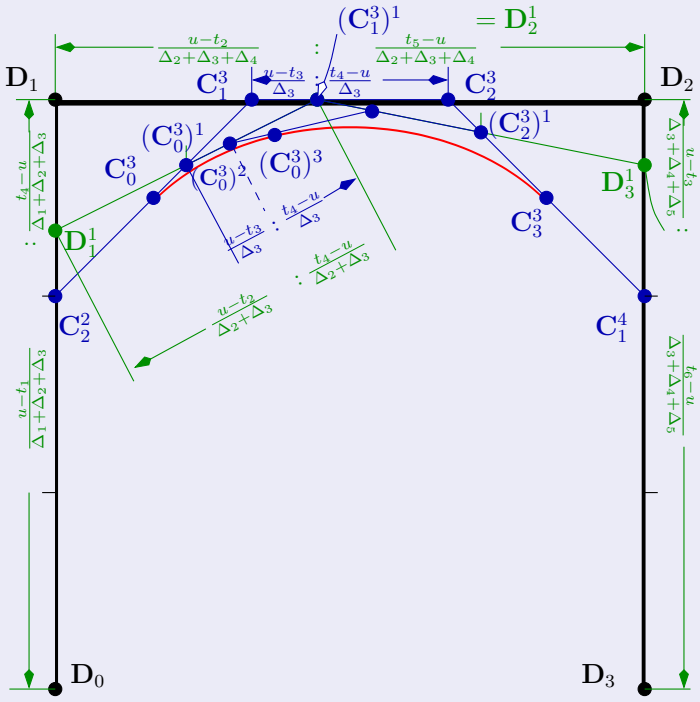
$$(\mathbf{C}_0^r)^1 = \frac{t_{r+1}-u}{\Delta_r}\mathbf{C}_0^r + \frac{u-t_r}{\Delta_r}\mathbf{C}_1^r = \frac{t_{r+1}-u}{\Delta_{r-1}+\Delta_r}\mathbf{D}_{r-2} + \frac{u-t_{r-1}}{\Delta_{r-1}+\Delta_r}\mathbf{D}_{r-1}$$

$$(\mathbf{C}_1^r)^1 = \frac{t_{r+1}-u}{\Delta_r}\mathbf{C}_1^r + \frac{u-t_r}{\Delta_r}\mathbf{C}_2^r = \frac{t_{r+2}-u}{\Delta_r+\Delta_{r+1}}\mathbf{D}_{r-1} + \frac{u-t_r}{\Delta_r+\Delta_{r+1}}\mathbf{D}_r$$



Observation

Given *parameter* $u \in [t_r, t_{r+1}]$ *and the reconstructed Bézier-points* $\mathbf{C}_0^r, \ldots, \mathbf{C}_3^r$ *and the de Casteljau points* $(\mathbf{C}_i^r)^k$ *(all in blue)*



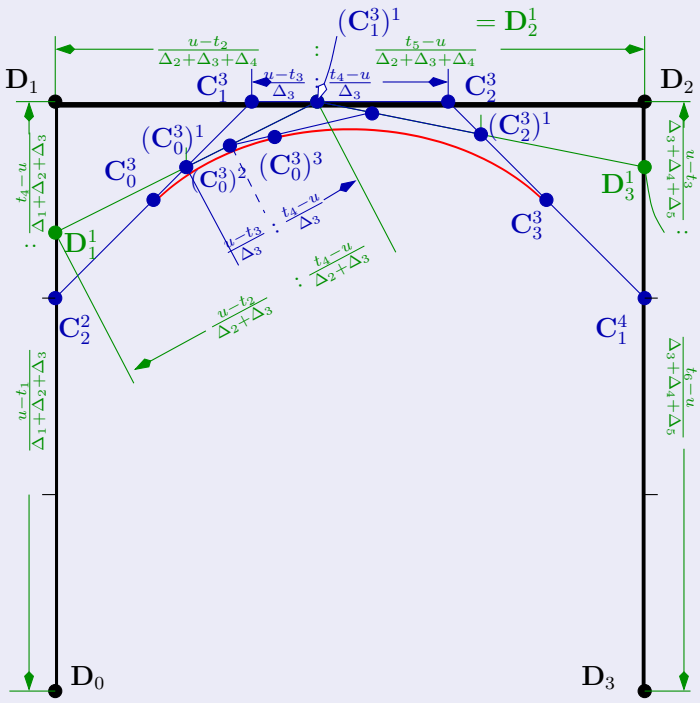
Direct Evaluation of Cubic B-Splines 53

Observation

Given *parameter* $u \in [t_r, t_{r+1}]$ *and the reconstructed Bézier-points* $\mathbf{C}_0^r, \ldots, \mathbf{C}_3^r$ *and the de Casteljau points* $(\mathbf{C}_i^r)^k$ *(all in blue)*

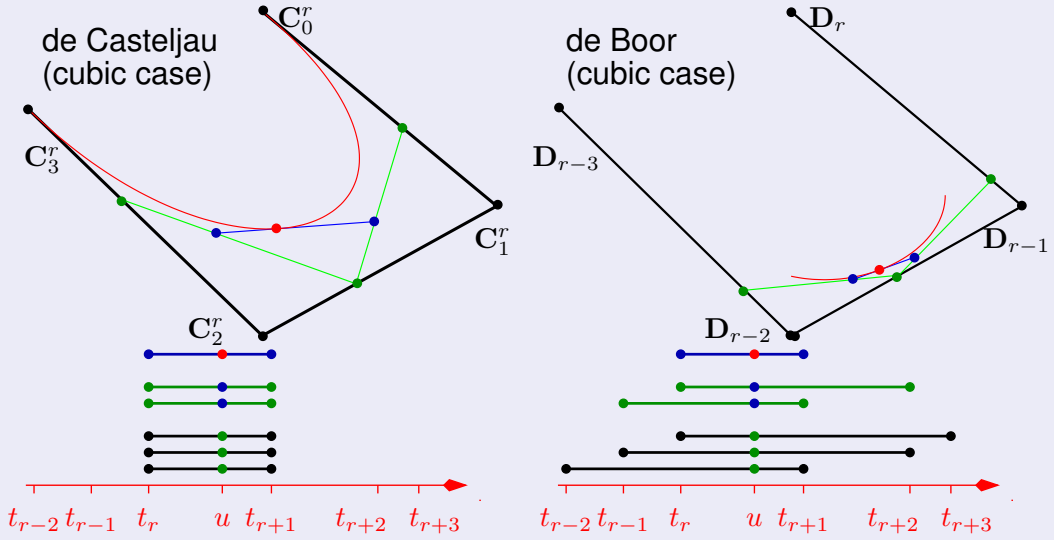
Elongation *of the* 1^{st} *de Casteljau layer* $(\mathbf{C}_0^r)^1, (\mathbf{C}_1^r)^1, (\mathbf{C}_2^r)^1$ *yields*

1. *new points* $\mathbf{D}_{r-2}^1, \mathbf{D}_{r-1}^1, \mathbf{D}_r^1$ *(green points), which*
2. *can be computed using split ratio over* 3 *segments*



Remark

The major difference between Bézier- and B-spline curves is the weighting of the control points. The de Casteljau algorithm uses constant weights with respect to the respective segment $[t_r, t_{r+1}]$. The de Boor algorithm for B-splines uses varying weights with respect to neighboring segments.



3.3.1: The de Boor Algorithm

Algorithm (De Boor's Algorithm, the general case)

The general case *is the natural expansion of spanning more segments for higher degrees.*

Given: *de Boor points* $\mathbf{D}_0, \ldots, \mathbf{D}_m$, *a knot vector* $\{t_0, \ldots, t_{n+m+1}\}$, $t_i < t_{i+1}$ *and a parameter* $u \in [t_n, t_{m+1}]$

Relevant *for* $u \in [t_r, t_{r+1}[$ *are the* $n+1$ *de Boor points* $\mathbf{D}_i^0 = \mathbf{D}_i,\ i = r-n, \ldots, r$

Recursion: *Affine combinations of neighboring control points:*

$$\mathbf{D}_i^k = (1 - \alpha_i^k(u))\mathbf{D}_{i-1}^{k-1} + \alpha_i^k(u)\mathbf{D}_i^{k-1},\ i = r-n+k, \ldots, r,\ k = 1, \ldots, n$$

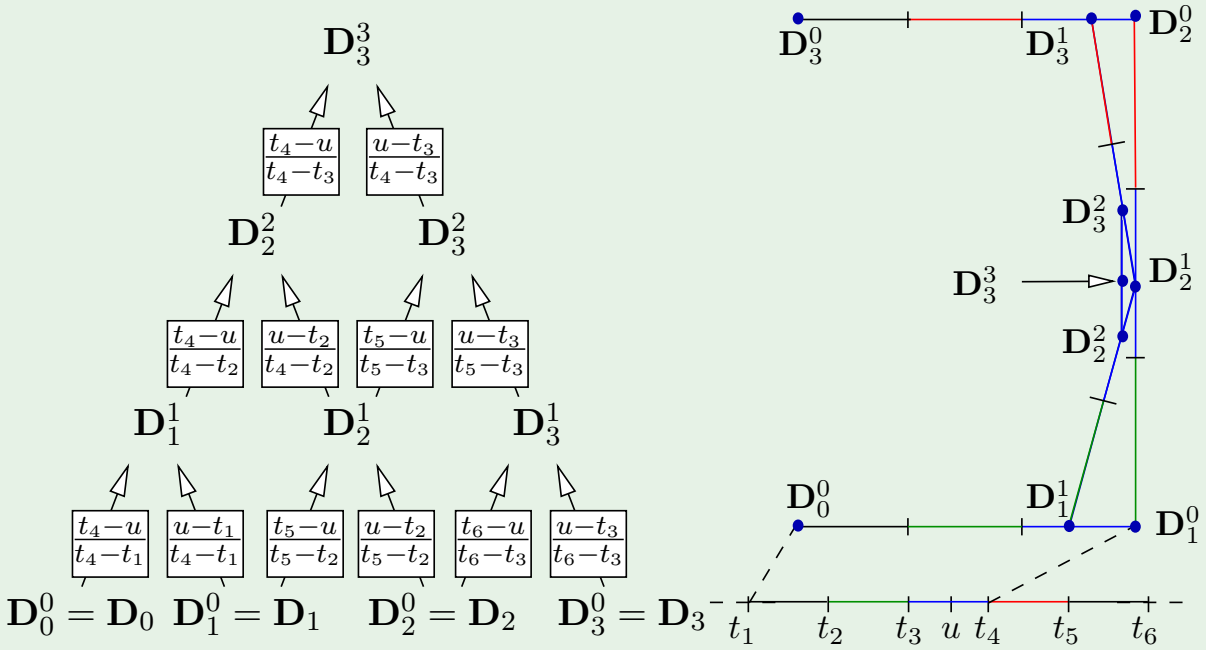
with

$$\alpha_i^k(u) = \frac{u - t_i}{t_{i+(n-k)+1} - t_i} = \frac{u - t_i}{\Delta_i + \ldots + \Delta_{i+(n-k)}},\ i = r-(n-k), \ldots, r,\ k = 1, \ldots, n$$

Result: $\mathbf{D}(u) = \mathbf{D}_r^n$

Example

Given: Cubic B-spline with de Boor-points $\mathbf{D}_0, \ldots, \mathbf{D}_m$ and $u \in [t_3, t_4[$

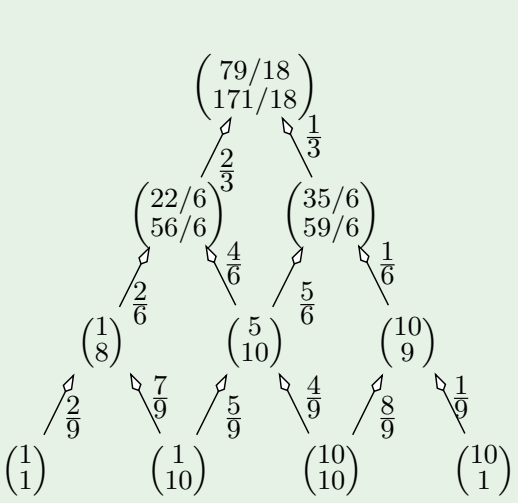


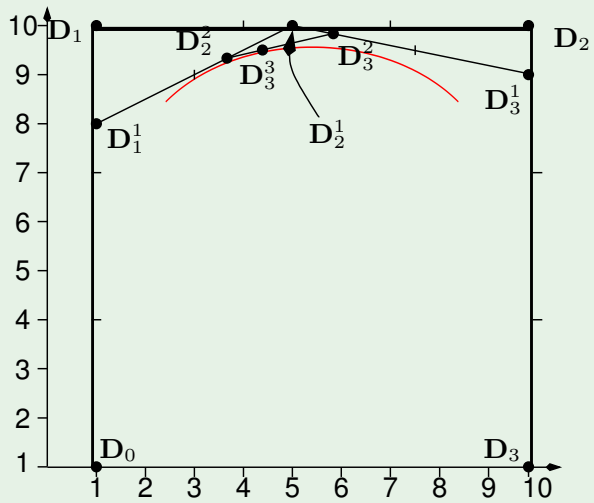
Example

Given: Uniform knot vector $t_i = i,\ i = 0, \ldots, 5$ and
$\mathbf{D}_0 = (1,1)$, $\mathbf{D}_1 = (1,10)$, $\mathbf{D}_2 = (10,10)$, $\mathbf{D}_3 = (10,1)$, $u = 10/3$, $r = 3$

Weights: Determination of weights $\alpha_i^k = \alpha_i^k(10/3) = \frac{u-i}{4-k} = \frac{10-3i}{12-3k}$:

$$\alpha_1^1 = 7/9, \quad \alpha_2^1 = 4/9, \quad \alpha_3^1 = 1/9, \quad \alpha_2^2 = 4/6, \quad \alpha_3^2 = 1/6, \quad \alpha_3^3 = 1/3$$





Approach (Quadratic B-Splines Basis Functions)

Goal: *A B-spline representation using basis function $N_i^n(u)$, i.e.*

$$\mathbf{D}(u) := \sum_{i=0}^{m} \mathbf{D}_i N_i^n(u),\ u \in [t_n, t_{m+1}]$$

Quadratic Basis Function: *For a fix i_0 we deduce $N_{i_0}^2$ via setting $\mathbf{D}_i = 0,\ i \neq i_0$ and $\mathbf{D}_{i_0} = 1$.*

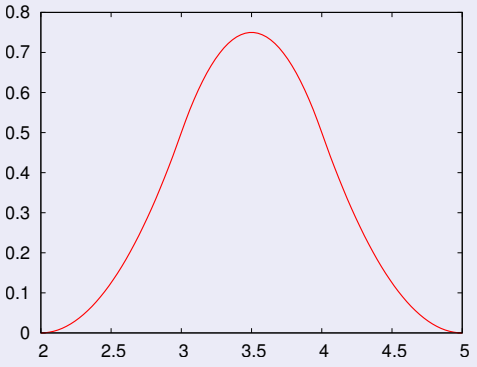
Example: *Determine $N_2^2(u)$ as a piecewise quadratic polynom in the uniform case ($t_i = i$). Converting to Bézier representation, all $\mathbf{C}_i^j = 0$ but*

$\mathbf{C}_2^2 = \mathbf{C}_0^3 = \mathbf{C}_2^3 = \mathbf{C}_0^4 = \frac{1}{2}, \mathbf{C}_1^3 = 1,$ *thus we get*

$$u \in [2,3[\text{:}\quad N_2^2(u) = \mathbf{C}^2(u) = \tfrac{1}{2} B_2^2(u)$$

$$u \in [3,4[\text{:}\quad N_2^2(u) = \mathbf{C}^3(u) = \tfrac{1}{2}\left(B_0^2(u) + B_2^2(u)\right) + B_1^2(u)$$

$$u \in [4,5[\text{:}\quad N_2^2(u) = \mathbf{C}^4(u) = \tfrac{1}{2} B_0^2(u)$$



General B-Spline Curves

Objective

Goal: *Basic function as piecewise polynom $N_i^n(u)$ with a given degree $n \in \mathbb{N}$, so that*

1. *Maximum continuity between the curve segments, i.e. C^{n-1}-continuous for degree n*
2. *Keeping the influence of the control points as small as possible (locality). This means to keep $\operatorname{supp}(N_i^n)$ as small as possible*

$$\operatorname{supp}(N_i^n) := \{u\ :\ N_i^n(u) \neq 0\}$$ *(support, (dt: Träger))*

3. *Partition of unity: $\sum N_i^n(u) \equiv 1,\ N_i^n(u) \geq 0$ (affine invariance, convex hull)*

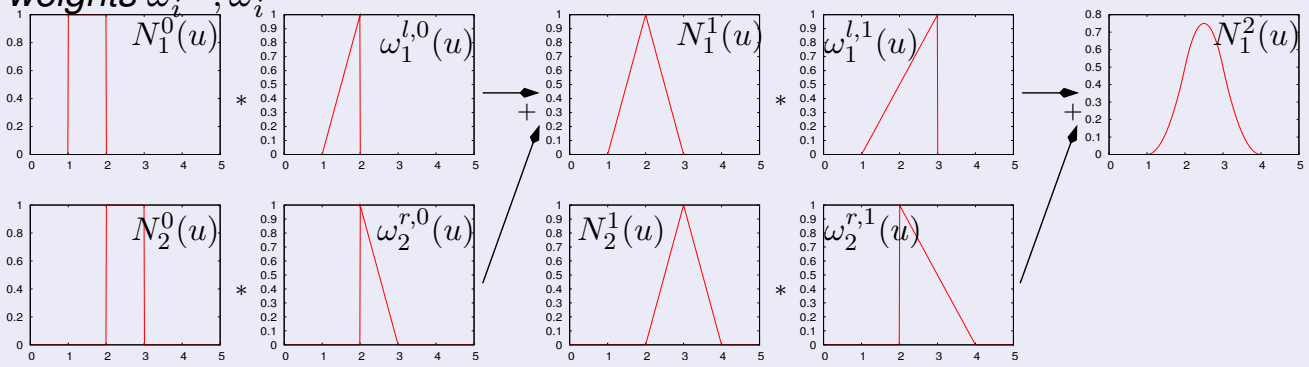
Knot vector: $T = \{t_0, \ldots, t_{n+m+1}\},\ t_i \leq t_{i+1},\ t_i \in \mathbb{R}$ *with:* $N_i^n\big|_{[t_i, t_{i+1}[} \in \mathcal{P}^n$

Case n=0: *Obviously a sensible definition is $N_i^0 = \chi([t_i, t_{i+1}[)$, with*

$$\chi(M)(u) = \begin{cases} 1 & u \in M \\ 0 & \text{else} \end{cases}$$ *characteristic function*

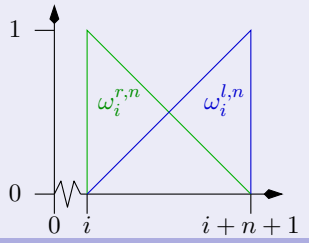
Approach (Recursive B-Spline Basis Functions)

$N_i^{n+1}(u)$ *results from the weighted sum of* $N_i^n(u)$ *and* $N_{i+1}^n(u)$ *with linear weights* $\omega_i^{l,n}, \omega_i^{r,n}$



Remark

- *On* $\operatorname{supp}(N_i^n)$ $\omega_i^{l,n}, \omega_i^{r,n}$ *rises/falls from* 0 *to* 1, *resp. from* 1 *to* 0
- *Valid is* $\sum N_i^n \equiv 1$ *(partition of unity, see below) and* $N_i^n(u) \geq 0$



Definition (Normalized B-spline Basis Functions)

Given: Knot vector $T = \{\ldots, t_{i-1}, t_i, t_{i+1}, \ldots\}$, $t_i \leq t_{i+1}$, $t_i \in \mathbb{R}$
The knot vector T is called *uniform*, if $t_{i+1} - t_i = \text{const}$, e.g. $t_i = i$

Case $n = 0$**:** Set $N_i^0 := \chi([t_i, t_{i+1}[)$

Recursion $n \to n+1$**:** $N_i^{n+1}(u) := \omega_i^{l,n}(u) \cdot N_i^n(u) + \omega_{i+1}^{r,n}(u) \cdot N_{i+1}^n(u)$ with

$$\omega_i^{l,n}(u) = \frac{u - t_i}{t_{i+n+1} - t_i}, \quad \omega_i^{r,n}(u) = \frac{t_{i+n+1} - u}{t_{i+n+1} - t_i}$$

Remark (Partition of Unity)

N_i^n *forms a partition of unity within the knot vector, since* $N_i^n(u) \geq 0$ *and by induction:* $\sum N_i^0 \equiv 1$ *and*

$$\begin{aligned}\sum_i N_i^{n+1}(u) &= \sum_i \left(\omega_i^{l,n}(u) N_i^n(u) + \omega_{i+1}^{r,n} N_{i+1}^n(u)\right) \\ &= \sum_i \underbrace{\left(\omega_i^{l,n}(u) + \omega_i^{r,n}(u)\right)}_{=1} N_i^n(u) = 1\end{aligned}$$

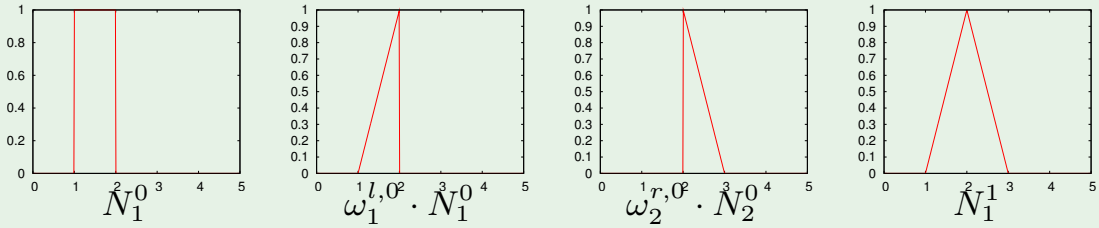
Example

Given: Uniform knot vector $T = \{t_0, \ldots, t_{n+m+1}\},\ t_i = i$

$n = 0$: $N_i^0 = \chi[i, i+1[= \begin{cases} 1 & \text{if } u \in [i, i+1[\\ 0 & \text{if } u \notin [i, i+1[\end{cases}$

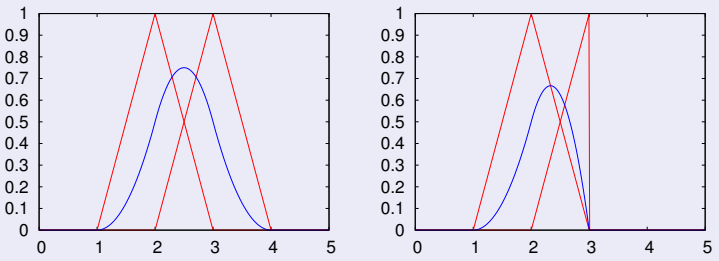
$n = 1$: $\omega_i^{l,0}(u) = u - i,\ \omega_i^{r,0}(u) = i + 1 - u$ and $N_i^1 = \omega_i^{l,0} \cdot N_i^0 + \omega_{i+1}^{r,0} \cdot N_{i+1}^0$

$$N_i^1(u) = (u-i) \cdot \chi[i, i+1[+ (i+2-u) \cdot \chi[i+1, i+2[$$
$$= \begin{cases} u - i & \text{if } u \in [i, i+1[\\ i + 2 - u & \text{if } u \in [i+1, i+2[\\ 0 & \text{else} \end{cases}$$



Property (Influence of the Knots t_i)

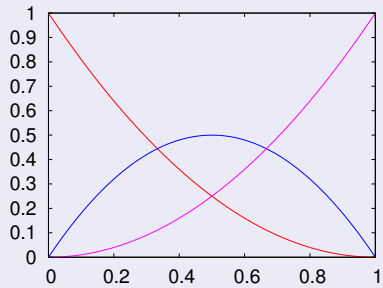
1. k-folded knot $t_{i-1} < t_i = t_{i+1} = \ldots = t_{i+k-1} < t_{i+k}$ *reduces the continuity by* k*; example* N_1^1, N_2^1, N_1^2 *uniform (left) and* $T = \{1, 2, 3, 3, 4\}$ *(right)*



2. $T = \{ \underbrace{0, \ldots, 0}_{(n+1) \text{ fold}}, \underbrace{1, \ldots, 1}_{(n+1) \text{ fold}} \}$ *delivers:*

$$N_i^n = B_i^n,\ \forall i = 0, \ldots, n$$

3. *On* $[t_n, t_{m+1}]$ $N_0^n, \ldots, N_m^n,\ m > n$ *forms a basis of all piecewise polynomials of the* n *th degree with the continuity given by* T



N_0^2, N_1^2, N_2^2 *for* $T = \{0, 0, 0, 1, 1, 1\}$

Remark (Properties of B-spline curves)

Polynomial curve: $\mathbf{D}$ *is a piecewise polynomial curve of degree* n *with* C^{n-1}*-transitions for single knots, and* C^{n-k} *for* k*-fold knots, respectively*

Locality: $\mathbf{D}_i$ *influences the curve only locally, since*

$$\operatorname{supp}(N_i^n) = \{u \,:\, N_i^n(u) \neq 0\} =]t_i, t_{i+n+1}[\quad \textit{(local support)}$$

Affine invariance & convex hull *since* $\sum_{i=0}^{m} N_i^n = 1$ *and* $N_i^n \geq 0$ *on* $[t_n, t_{m+1}]$

Local convex hull: $\mathbf{D}\big|_{[t_i,t_{i+1}[}$ *proceeds within the hull of* $\mathbf{D}_{i-n}, \ldots, \mathbf{D}_i$

Derivative *of a (piecewise) polynomial reduces the degree from* n *to* $n-1$

Derivative of B-splines *with knot vector* $T = \{t_0 \leq t_1 \leq \ldots \leq t_{m+n+1}\}$*:*

$$\mathbf{D}(u) = \sum_{i=0}^{m} \mathbf{D}_i N_i^n(u) \quad \Longrightarrow \quad \mathbf{D}'(u) = n \sum_{i=0}^{m} \frac{\mathbf{D}_i - \mathbf{D}_{i-1}}{t_{i+n} - t_i} N_i^{n-1}(u)$$

Knot Insertion (Böhm's Algorithm)

Algorithm

Goal: *Additional degrees of freedom by adding new control points*

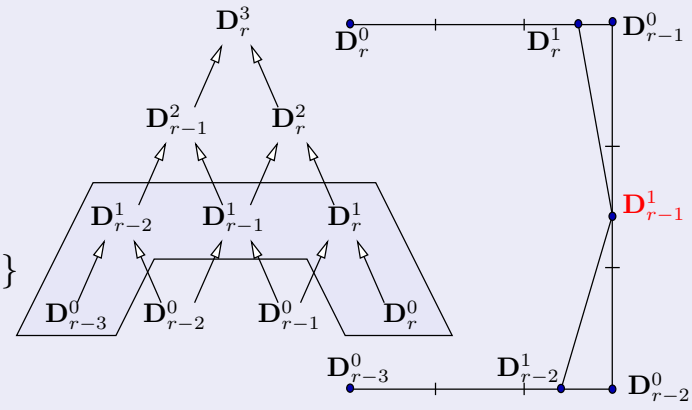
Piecewise polynomial *over*

$$T = \{t_0, t_1, \ldots, t_{m+n+1}\}$$

is also a piecewise polyn. over

$$T^* = \{t_0, \ldots, t_r, t^*, t_{r+1} \ldots, t_{m+n+1}\}$$

Control points are calculated via de Boor-algorithm for $u = t^*$



Curve split *requires* $(n+1)$*-fold knot-insertion, i.e. a complete de Boor:*

Knot vector: $T^* = \{t_0, \ldots, t_r, \underbrace{t^*, \ldots, t^*}_{(n+1)\text{-folded}}, t_{r+1} \ldots, t_{m+n+1}\}$

Left curve: $T^L = \{t_0, \ldots, t_r, t^*, \ldots, t^*\},\ KP : \mathbf{D}_0, \ldots, \mathbf{D}_{r-n}, \mathbf{D}_{r-n+1}^1, \ldots, \mathbf{D}_r^n$

Right curve: $T^R = \{t^*, \ldots, t^*, t_{r+1} \ldots, t_{m+n+1}\}, KP : \mathbf{D}_r^n, \ldots, \mathbf{D}_r^1, \mathbf{D}_r, \ldots, \mathbf{D}_m$

Approach

Given: *Curve scheme* $\mathbf{P}(u) = \sum_{i=0}^{n} \mathbf{P}_i F_i(u)$

Variation of the control points $\mathbf{P}_i$ *along another curve*

$$\mathbf{P}_i = \mathbf{P}_i(v) = \sum_{j=0}^{m} \mathbf{P}_{ij} F_j(v) \text{ results in } \mathbf{P}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{P}_{ij} F_i(u) F_j(v)$$

TP-Bézier surface: *Usually, polynomial curves of the same degree are used:*

$$\mathbf{C}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{n} \mathbf{C}_{ij} B_i^n(u) B_j^n(v)$$

TP-B-spline surfaces *generally have a different number of segments in* u*- and* v*-direction:*

$$\mathbf{D}(u,v) = \sum_{i=0}^{m_u}\sum_{j=0}^{m_v} \mathbf{D}_{ij} N_i^n(u) N_j^n(v)$$

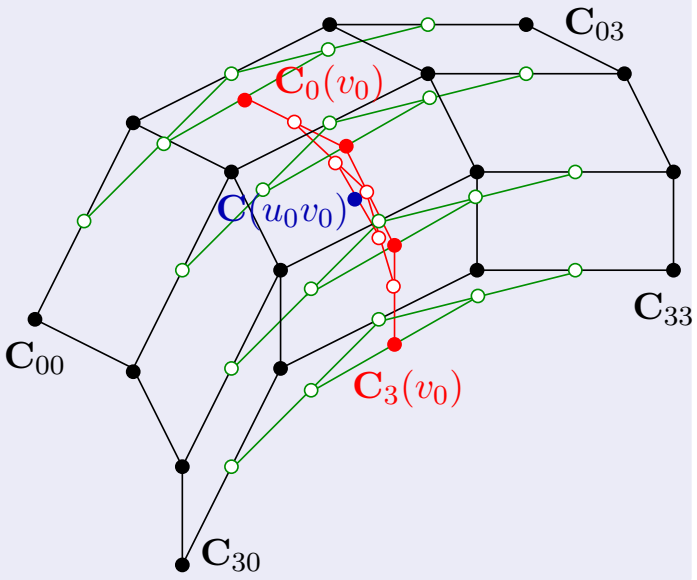
Evaluation of Bézier-Surfaces

Algorithm

Dual-step de Casteljau: *Separate analysis for parameter* (u_0, v_0)*:*

1. $\forall i = 0, \ldots, n : \mathbf{C}_i(v_0) = \sum_{j=0}^{n} \mathbf{C}_{ij} B_j^n(v_0)$
2. $\mathbf{C}(u_0, v_0) = \sum_{i=0}^{n} \mathbf{C}_i(v_0) B_i^n(u_0)$

Notice: *The roles of* u *and* v *can be switched*



Remark

Derivation of function $f : \mathbb{R} \to \mathbb{R} : f'(u) = \frac{\partial f}{\partial u}(u) = \lim_{\delta \to 0} \frac{f(u+\delta)-f(u)}{\delta}$.

Derivation of a curve $\mathbf{F} : \mathbb{R} \to \mathbb{R}^3 : \mathbf{F}'(u) = \frac{\partial \mathbf{F}}{\partial u}(u) = \lim_{\delta \to 0} \frac{\mathbf{F}(u+\delta)-\mathbf{F}(u)}{\delta}$.

Partial derivative of a surface $\mathbf{F} : \mathbb{R}^2 \to \mathbb{R}^3$*: Derive one variable keeping the others fixed*

$$\frac{\partial \mathbf{F}}{\partial u}(u,v) := \lim_{\delta \to 0} \frac{\mathbf{F}(u+\delta,v) - \mathbf{F}(u,v)}{\delta}$$
$$\frac{\partial \mathbf{F}}{\partial v}(u,v) := \lim_{\delta \to 0} \frac{\mathbf{F}(u,v+\delta) - \mathbf{F}(u,v)}{\delta}$$

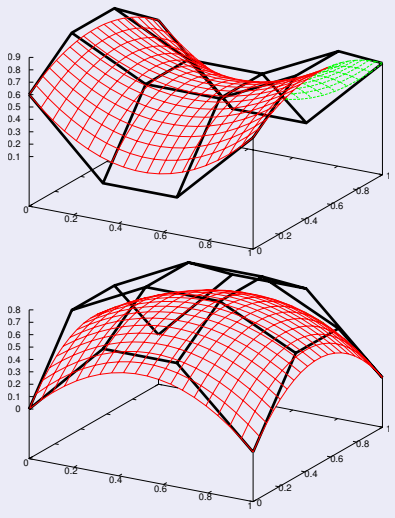
So: *Use derivation rules as usual, once for* u*, once for* v*.*

Property

Interpolation of vertices: $\mathbf{C}(0,0) = \mathbf{C}_{00}, \mathbf{C}(1,0) = \mathbf{C}_{n0}, \ldots$

Derivative:

$$\frac{\partial \mathbf{C}}{\partial u}(u,v) = \frac{\partial}{\partial u}\left(\sum_{i=0}^{n}\left(\sum_{j=0}^{n}\mathbf{C}_{i,j}B_j^n(v)\right)B_i^n(u)\right)$$
$$= n \sum_{(i,j)=(0,0)}^{(n-1,n)} \left(\mathbf{C}_{i+1,j} - \mathbf{C}_{i,j}\right) B_i^{n-1}(u) B_j^n(v)$$
$$\frac{\partial \mathbf{C}}{\partial u}(0,v) = n \sum_{j=0}^{n} \left(\mathbf{C}_{1,j} - \mathbf{C}_{0,j}\right) B_j^n(v)$$
$$\frac{\partial \mathbf{C}}{\partial u}(0,0) = n\left(\mathbf{C}_{1,0} - \mathbf{C}_{0,0}\right)$$



Global influence of $\mathbf{C}_{ij}$**,** *since* $B_i^n(u) \cdot B_j^n(v) \neq 0$ *on* $(u,v) \in]0,1[^2$

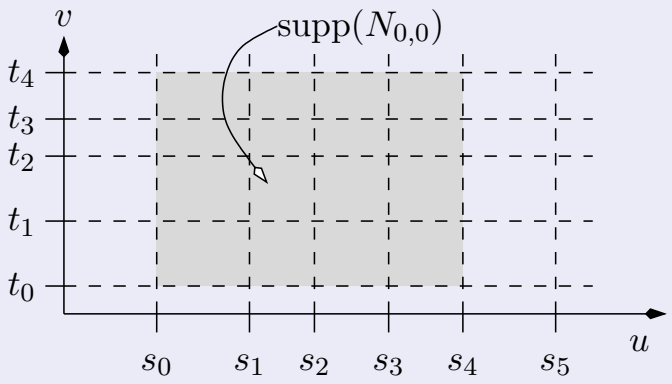
Affine invariance & convexe hulls *, since*
$\sum_{(i,j)} B_i^n(u)B_j^n(v) = (\sum_i B_i^n(u))(\sum_j B_j^n(v)) = 1$

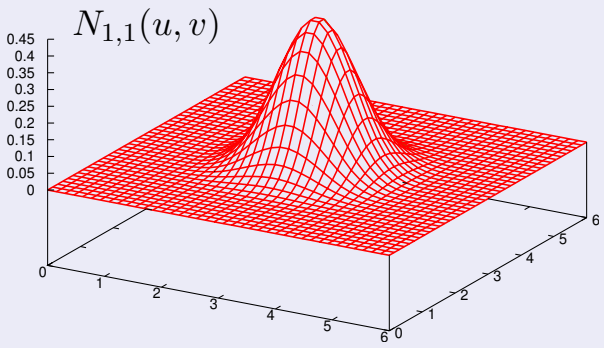
Property

Knot vectors $S = \{s_0, s_1, \ldots, s_{m_u+n+1}\}$, $T = \{t_0, t_1, \ldots, t_{m_v+n+1}\}$ *in the direction of* u *and* v*, respectively*

Local influence of $\mathbf{D}_{ij}$: $\mathbf{D}_{ij}$ *namipulates* $\mathbf{D}(]s_i, s_{i+n+1}[\times]t_j, t_{j+n+1}[)$*, since*

$$N_{ij}(u,v) := N_i^n(u) \cdot N_j^n(v) = 0 \; \forall (u,v) \notin]s_i, s_{i+n+1}[\times]t_j, t_{j+n+1}[$$





Affine invariance and convex hull: *are equally valid, since* $\sum_{(i,j)} N_i^n(u) N_j^n(v) = 1$ *auf* $[s_n, s_{n+m_u+1}] \times [t_n, t_{n+m_v+1}]$

Remark (Geometric Continuity)

So far: *Piecewise curves introduced via* C^k*-continuity*

Ambiguity: *Geometric form (curve) can be represented by various functions: For* $\mathbf{C}(u), u \in [a,b]$ *and bijektive* $g : [c,d] \to [a,b]$ *results in*

$$\mathbf{D}(u) := \mathbf{C}(g(u)), \; u \in [c,d]$$

the same geometry, but the derivatives are different:

$$\textit{Chain rule: } \mathbf{D}'(u) = \mathbf{C}'(g(u)) \cdot g'(u) \neq \mathbf{C}'(g(u)) \textit{ if } g'(u) \neq 1$$

Geometric continuity: *Curves* $\mathbf{C}$ *and* $\mathbf{D}$ *are in* $\mathbf{D}(u_0)$ G^k*-continuous, if:*

$$\exists\, g \in C^k \textit{ bijectiv } : \mathbf{D}^{(i)}(u_0) = \frac{d^i}{d\,u^i}\mathbf{C}(g(u_0)), \; i = 0, \ldots, k$$

G^1*-continuity:* $\mathbf{D}'(u_0) = \mathbf{C}'(g(u_0)) \cdot \gamma_1, \; \gamma_1 > 0$

G^2*-continuity:* $\mathbf{D}''(u_0) = \mathbf{C}''(g(u_0)) \cdot (\gamma_1)^2 + \mathbf{C}'(g(u_0)) \cdot \gamma_2, \; \gamma_1, \gamma_2 > 0$

Definition

Objective: Determine a local coordination system to a curve (camera path).

Given: Curve $\mathbf{C}$, C^2-continuous with $\mathbf{C}'(u), \mathbf{C}''(u) \neq \vec{\mathbf{0}}, \mathbf{C}'(u) \not\parallel \mathbf{C}''(u),\ \forall u$

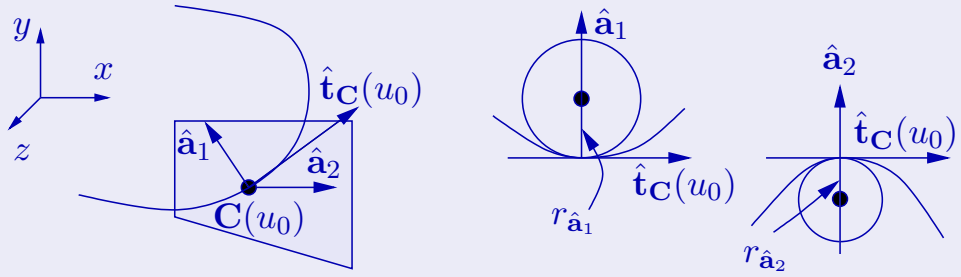
Geometric quantities of $\mathbf{C}$ in u_0:

Tangent $\hat{\mathbf{t}}_{\mathbf{C}}(u_0)$: points in the motion direction: $\hat{\mathbf{t}}_{\mathbf{C}}(u_0) = \mathbf{C}'(u_0) / \|\mathbf{C}'(u_0)\|$

Curvature $\kappa_{\mathbf{C}}(u_0)$: Curvature is defined in a plane

$$\mathrm{span}\{\hat{\mathbf{t}}_{\mathbf{C}}(u_0), \hat{\mathbf{a}}\} \text{ with arbitrary } \hat{\mathbf{a}} \perp \hat{\mathbf{t}}_{\mathbf{C}}(u_0)$$

as: $\kappa_{\mathbf{C},\hat{\mathbf{a}}}(u_0) = \dfrac{1}{r_{\hat{\mathbf{a}}}}$, $r_{\hat{\mathbf{a}}}$ radius of the best matched circle

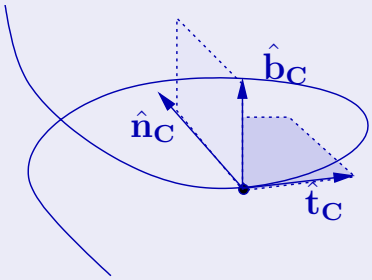


Property (Geometric quantities of $\mathbf{C}$ in u_0)

Normal $\hat{\mathbf{n}}_{\mathbf{C}}(u_0)$: *corresponds to $\hat{\mathbf{a}}$ with a maximum curvature of $\kappa_{\mathbf{C},\hat{\mathbf{a}}}(u_0)$*
It is necessary that $\hat{\mathbf{n}}_{\mathbf{C}}(u_0) \in \mathrm{span}\{\mathbf{C}'(u_0), \mathbf{C}''(u_0)\}$.

Curvature: $\kappa_{\mathbf{C}}(u_0) = \kappa_{\mathbf{C},\hat{\mathbf{n}}_{\mathbf{C}}(u_0)}(u_0)$

Bi-normal $\hat{\mathbf{b}}_{\mathbf{C}}(u_0)$: *Third vector perpendicular to the tangent and normal:*



$$\hat{\mathbf{b}}_{\mathbf{C}}(u_0) := \hat{\mathbf{t}}_{\mathbf{C}}(u_0) \times \hat{\mathbf{n}}_{\mathbf{C}}(u_0)$$

Frenet-Serret frame: *The coordinate system $\{\hat{\mathbf{t}}_{\mathbf{C}}(u_0), \hat{\mathbf{n}}_{\mathbf{C}}(u_0), \hat{\mathbf{b}}_{\mathbf{C}}(u_0)\}$ in point $\mathbf{C}(u_0)$.*

Curvature vector: $\kappa_{\mathbf{C}}(u_0) \cdot \hat{\mathbf{n}}_{\mathbf{C}}(u_0)$

Osculating plane: $\mathrm{span}\{\hat{\mathbf{t}}_{\mathbf{C}}(u_0), \hat{\mathbf{n}}_{\mathbf{C}}(u_0)\}$, *the locally best fitting plane*

Normal plain: *Plain perpendicular to the curve progression, that is* $\mathrm{span}\{\hat{\mathbf{n}}_{\mathbf{C}}(u_0), \hat{\mathbf{b}}_{\mathbf{C}}(u_0)\}$

Remark

Calculation of the Frenet-Serret Frame The coordination system $\{\hat{\mathbf{t}}_\mathbf{C}(u_0), \hat{\mathbf{n}}_\mathbf{C}(u_0), \hat{\mathbf{b}}_\mathbf{C}(u_0)\}$ *is calculated as follows:*

Tangent: $$\hat{\mathbf{t}}_\mathbf{C}(u_0) = \frac{\mathbf{C}'(u_0)}{\|\mathbf{C}'(u_0)\|}$$

Binormale: $$\hat{\mathbf{b}}_\mathbf{C}(u_0) = \frac{\mathbf{C}'(u_0) \times \mathbf{C}''(u_0)}{\|\mathbf{C}'(u_0) \times \mathbf{C}''(u_0)\|}$$

since $\mathbf{C}'(u_0)$ *and* $\mathbf{C}''(u_0)$ *span the Schmiegeebene and* $\{\hat{\mathbf{t}}_\mathbf{C}(u_0), \hat{\mathbf{n}}_\mathbf{C}(u_0), \hat{\mathbf{b}}_\mathbf{C}(u_0)\}$ *and therewith also* $\{\mathbf{C}'(u_0), \mathbf{C}''(u_0), \hat{\mathbf{b}}_\mathbf{C}(u_0)\}$ *are right-handed*

Normal: $$\hat{\mathbf{n}}_\mathbf{C}(u_0) = \hat{\mathbf{b}}_\mathbf{C}(u_0) \times \hat{\mathbf{t}}_\mathbf{C}(u_0)$$

(no normalization needed, since $\hat{\mathbf{b}}_\mathbf{C}, \hat{\mathbf{t}}_\mathbf{C}$ *are orthonormal)*

Curvature: $$\kappa_\mathbf{C}(u_0) = \frac{\|\mathbf{C}'(u_0) \times \mathbf{C}''(u_0)\|}{\|\mathbf{C}'(u_0)\|^3}$$

Example (Frenet–Serret Frame of a Spiral)

Spiral: $\mathbf{C}(u) = \begin{pmatrix} \sin u \\ \cos u \\ u \end{pmatrix}$; derivatives: $\mathbf{C}'(u) = \begin{pmatrix} \cos u \\ -\sin u \\ 1 \end{pmatrix}$; $\mathbf{C}''(u) = \begin{pmatrix} -\sin u \\ -\cos u \\ 0 \end{pmatrix}$

with $\|\mathbf{C}'(u)\| = \sqrt{2}$ and $\mathbf{C}'(u) \times \mathbf{C}''(u) = \begin{pmatrix} \cos u \\ -\sin u \\ -1 \end{pmatrix}$ folgt: $\hat{\mathbf{t}}_\mathbf{C}(u) = \frac{1}{\sqrt{2}} \begin{pmatrix} \cos u \\ -\sin u \\ 1 \end{pmatrix}$

$$\hat{\mathbf{b}}_\mathbf{C}(u) = \frac{\mathbf{C}'(u) \times \mathbf{C}''(u)}{\|\mathbf{C}'(u) \times \mathbf{C}''(u)\|} = \frac{1}{\sqrt{2}} \begin{pmatrix} \cos u \\ -\sin u \\ -1 \end{pmatrix}, \quad \hat{\mathbf{n}}_\mathbf{C}(u) = \hat{\mathbf{b}}_\mathbf{C}(u) \times \hat{\mathbf{t}}_\mathbf{C}(u) = \begin{pmatrix} -\sin u \\ -\cos u \\ 0 \end{pmatrix}$$

For $u = \frac{\pi}{2}$: $\hat{\mathbf{t}}_\mathbf{C}(\frac{\pi}{2}) = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$

$$\hat{\mathbf{n}}_\mathbf{C}(\tfrac{\pi}{2}) = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}; \quad \hat{\mathbf{b}}_\mathbf{C}(\tfrac{\pi}{2}) = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -1 \\ -1 \end{pmatrix}$$