Computergraphik II

Prof. Dr. Andreas Kolb

Fachgruppe Computergraphik und Multimediasysteme

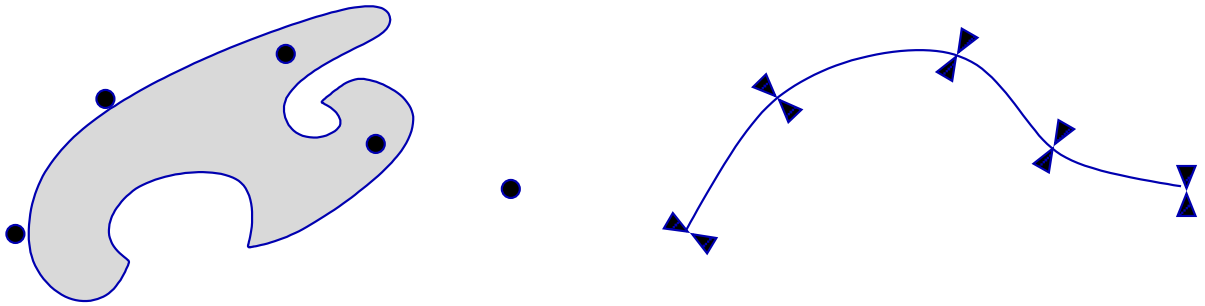
Universität Siegen – Fachbereich 12

Version: 10. November 2004

Bemerkung: Historie

Ursprung der Geometrischen Modellierung: Industrielle Fertigung i.w. Automobilbau

< 1960: Verschiedene zeichentechnische Methoden wie z.B. Kurvenlineal und Stahllineal.



- Ungenauigkeit bei Übertragung auf reale Formen für Guß, Stanzen etc.
- Flächen nur über Kurvenscharen handhabbar

nach 1960: Übertragung zeichentechnischer Ansätze auf Computer (NC-Steuerung)

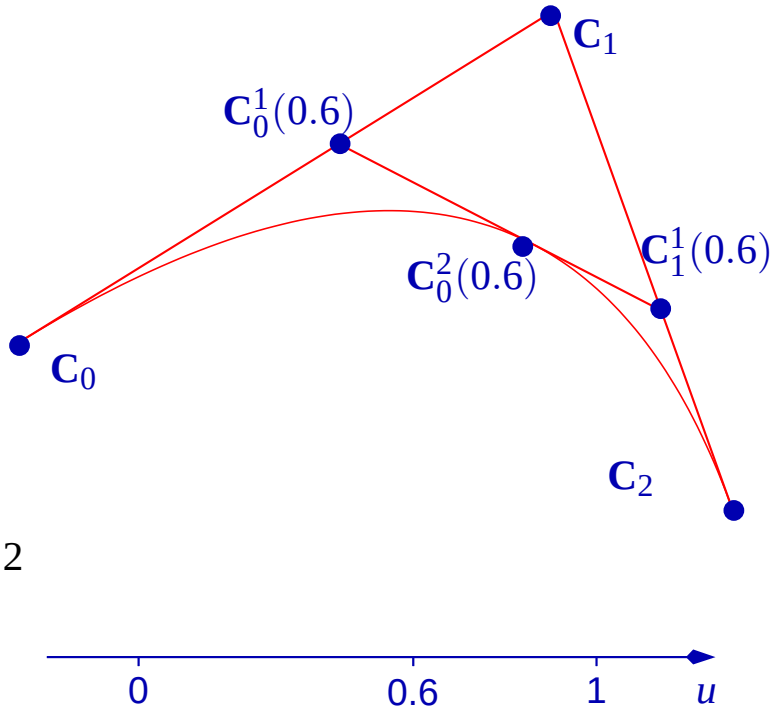
- Modellierung gekrümmte Objekte (Kurven aber auch Flächen)
- intuitiven Kontrollparametern mit hoher Flexibilität
- effizient auf Rechnern handhabbar ($\longrightarrow$ **Polynome!**)

Beispiel: Erzeugung von parametrischen Parabel-Kurven

Gegeben: Drei Punkte eine Parabel: $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^2$

Konstruktion: Sei $u \in [0,1]$, dann bilde

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



Diese Konstruktion hat folgende Eigenschaften:

1. $\mathbf{C}(u)$ ist **affine Kombination** der $\mathbf{C}_i$, da $(1-u)^2 + 2(1-u)u + u^2 = (1-u+u)^2 = 1$
2. **Endpunktinterpolation:** $\mathbf{C}_0^2(0) = \mathbf{C}_0,\ \mathbf{C}_0^2(1) = \mathbf{C}_2$
3. $\{\mathbf{C}(u) : u \in [0,1]\} \subset \Delta(\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2)$, also Kurve verläuft in **konvexer Hülle** der Kontrollpunkte, da Gewichte ≥ 0

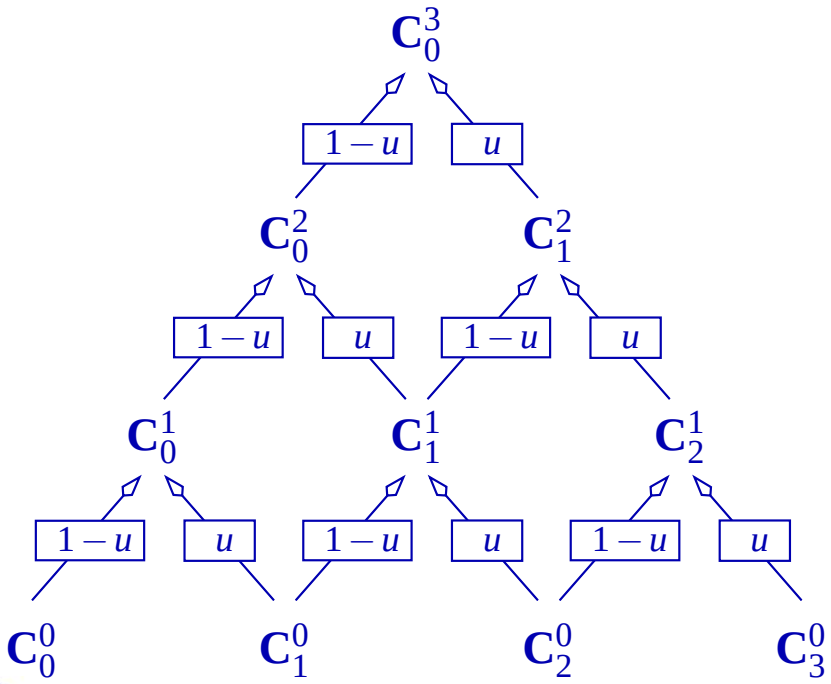
Algorithmus: de Casteljau

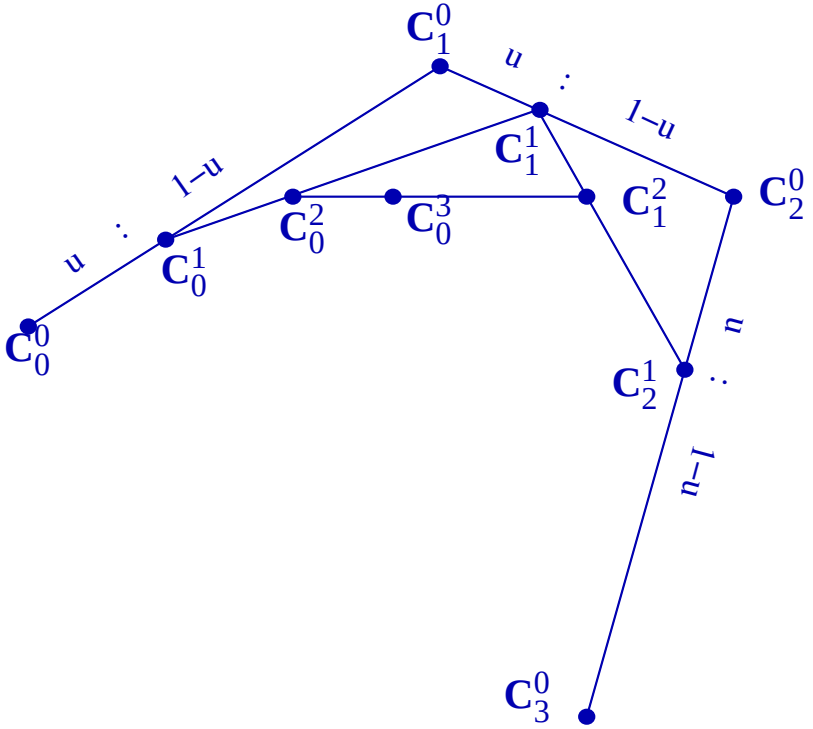
Gegeben: Kontrollpunkte $\mathbf{C}_i,\ i=0,\ldots,n$ für ein $n \in \mathbb{N}$, sowie der Parameter $u \in [0,1]$

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

Rekursion: Affin-Kombinationen benachbarter Kontrollpunkte:

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

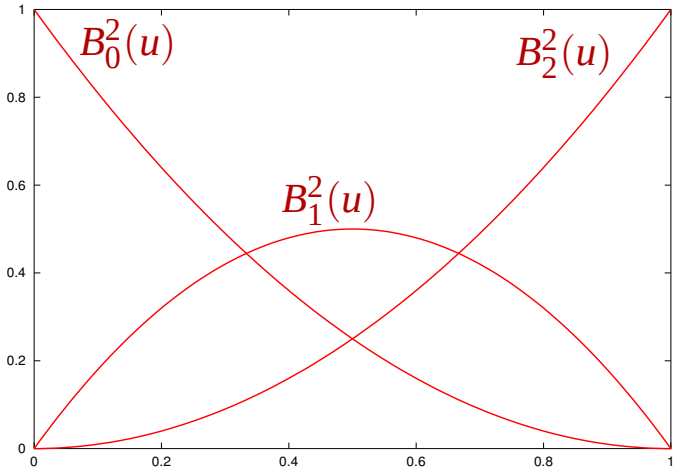


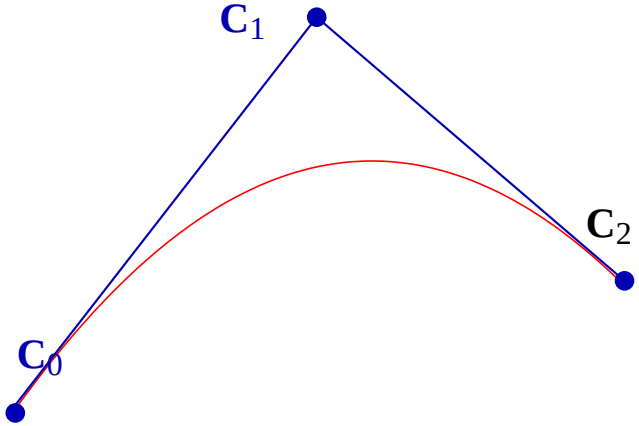


Bezeichnung: Bézier-Kurven ($n = 2$)

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

Bernstein-Bézier-Polynome: Gewichte $B_i^2(u)$ der Affin-Kombination der $\mathbf{C}_i$





Beachte: $B_i^2(u)$ entsprechen variablen Gewichten einer Affin-Kombination

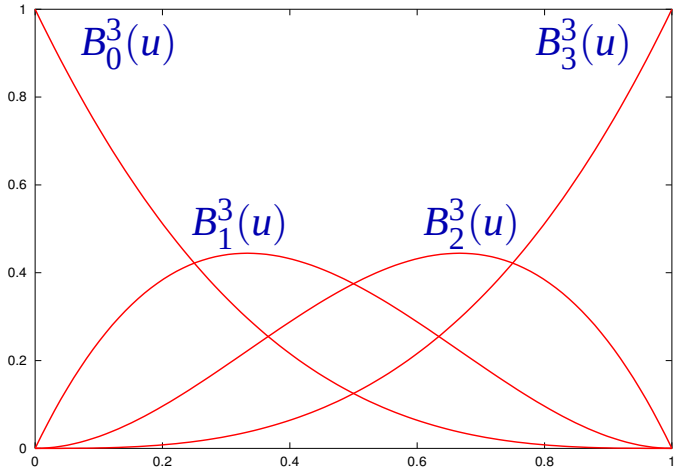
Bezeichnung: Bézier-Kurven (allgemeiner Fall)

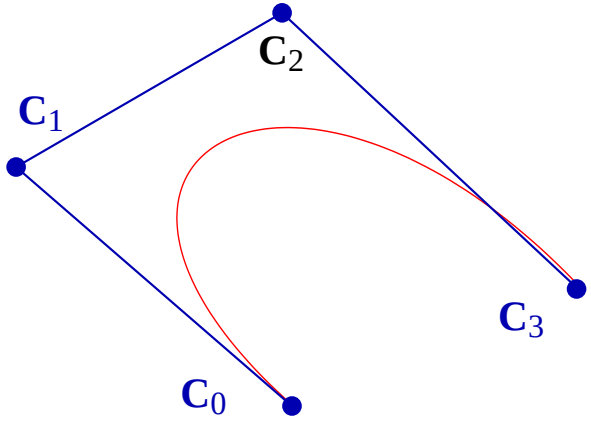
Bernstein-Bézier-Polynome: Für beliebiges $n \in \mathbb{N}$ wird definiert:

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

Bézier-Kurve: Für Kontrollpunkte $\mathbf{C}_0, \ldots, \mathbf{C}_n$ ist: $\mathbf{C}(u) = \sum_{i=0}^{n} B_i^n(u) \mathbf{C}_i$

Kubischer Fall $n = 3$: Häufig kommen Bézier-Kurven vom Grad 3 zum Einsatz:





Eigenschaften der Bernstein-Bézier-Polynome

Polynom-Basis: Polynome $\mathbf{P} \in \mathcal{P}^n$ vom Grad n können als Bézier-Kurve dargestellt werden:

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

Positivität: Für $u \in [0,1]$ gilt stets: $B_i^n(u) \geq 0$

Endpunktverhalten: $B_i^n(0) = \begin{cases} 1 & \text{falls } i = 0 \\ 0 & \text{sonst} \end{cases}$ $\quad B_i^n(1) = \begin{cases} 1 & \text{falls } i = n \\ 0 & \text{sonst} \end{cases}$

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

Ableitung:

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

Eigenschaften der Bézier-Kurven

Endpunktinterpolation: $\mathbf{C}(0) = \mathbf{C}_0$, $\mathbf{C}(1) = \mathbf{C}_n$, da $B_0^n(0) = B_n^n(1) = 1$, sonst $B_i^n(0) = B_i^n(1) = 0$

Globaler Einfluß der $\mathbf{C}_i$: $\mathbf{C}_i$ beeinflußt $\mathbf{C}(u)$, $u \in]0,1[$, da hier $B_i^n(u) \neq 0$, $\forall i$

Affine Invarianz: Die Kurve $\mathbf{C}$ ist affin invariant bzgl. seiner Kontrollpunkte:

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

Konvexe Hülle: Aufgrund der Positivität der B_i^n und der affinen Invarianz verläuft $\mathbf{C}(u)$, $u \in [0,1]$ innerhalb der konvexen Hülle der Kontrollpunkte

Variation Diminishing Property (ohne Beweis): Die Bézier-Kurve hat max. soviele Wendepunkte wie ihr Kontrollpolygon:

$\forall$ Gerade g gilt: $\#\{g \cap \text{ Kontrollpolygon}\} \geq \#\{g \cap \text{Kurve}\}$

Eigenschaften der Bézier-Kurven (Forts.)

Ableitung: Bézier-Darstellung der ersten Ableitung:

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

End-Tangenten (Spezialfall): Für $u \in \{0,1\}$ gilt:

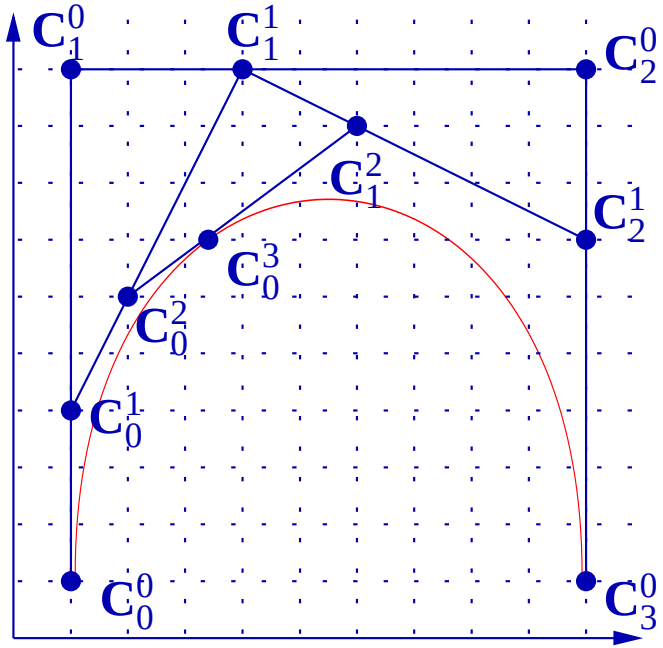
$$\mathbf{C}'(0) = n(\mathbf{C}_1 - \mathbf{C}_0) \qquad \mathbf{C}'(1) = n(\mathbf{C}_n - \mathbf{C}_{n-1})$$

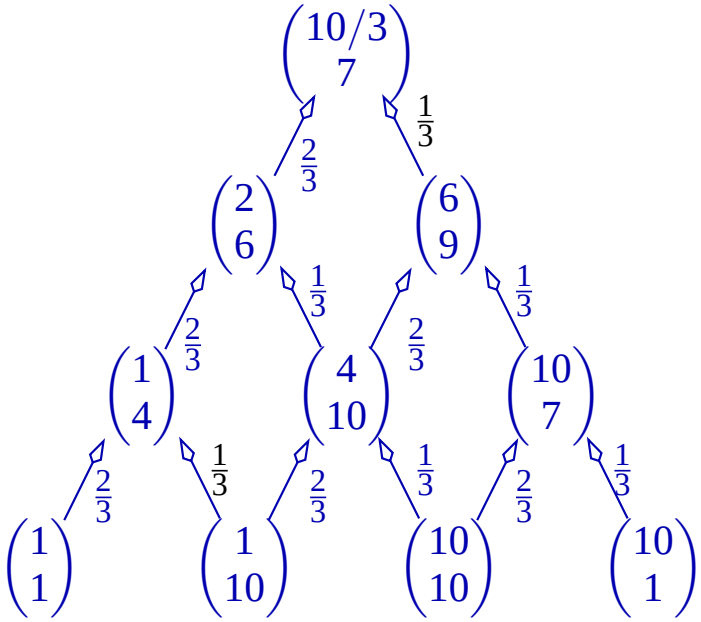
Bezug zu de Casteljau: Die letzte Kante des de Casteljau-Schemas entspricht der Tangentenrichtung der Kurve:

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

Beispiel:

Gegeben: Punkte $\mathbf{C}_0 = \begin{pmatrix}1\\1\end{pmatrix}, \mathbf{C}_1 = \begin{pmatrix}1\\10\end{pmatrix}, \mathbf{C}_2 = \begin{pmatrix}10\\10\end{pmatrix}, \mathbf{C}_3 = \begin{pmatrix}10\\1\end{pmatrix}$. Gesucht: $\mathbf{C}(\frac{1}{3})$





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

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

Auswertung von Polynomkurven

Bewertung de Casteljau:
- sehr stabil (nur Interpolation)
- Komplexität: $O(n^2)$ pro Auswertung, genauer

Vektor-Additionen: $\frac{(n+1)n}{2}$ Skalare Multiplikationen: $(n+1)n$

Horner-Schema: Ausgehend von der Polynomkurve $\mathbf{F}(u) = \sum_{i=0}^{n} \mathbf{A}_i u^i \in \mathcal{P}^n$ ($\mathcal{P}^n$: Raum der Polynome vom Grad $\leq n$)

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

- weniger stabil
- Komplexität: $O(n)$, genauer: n skalare Multiplikationen, n Vektor-Additionen

Vorwärts-Differenzen

Ziel: Auswertung von $\mathbf{F}(u) = \sum_{i=0}^{n} \mathbf{A}_i u^i$ an *äquidistanten Parametern*

$$u_j = u_0 + j \cdot \Delta,\ j = 0, \ldots, k \text{ für ein } k \in \mathbb{N} \text{ und Schrittweite } \Delta > 0$$

Erste Vorwärtdifferenz: Betrachte Polynom $\delta^1(u) := \mathbf{F}(u+\delta) - \mathbf{F}(u)$; es gilt:

$$\begin{aligned}\delta^1(u) &= \mathbf{F}(u+\Delta) - \mathbf{F}(u) = \mathbf{A}_n\left((u+\Delta)^n - u^n\right) + \mathbf{G}(u), \quad \mathbf{G} \in \mathcal{P}^{n-1} \\ &= \mathbf{A}_n\left(\left(\sum_{i=0}^{n}\binom{n}{i}u^i\Delta^{n-1}\right) - u^n\right) + \mathbf{G}(u) \quad \boxed{\Longrightarrow \delta^1 \in \mathcal{P}^{n-1}}\end{aligned}$$

***i*-te Vorwärtdifferenz:** Rekursive Differenzbildung

$$\delta^i(u) := \delta^{i-1}(u+\Delta) - \delta^{i-1}(u) \in \mathcal{P}^{n-i} \quad \text{insbesondere:} \quad \delta^n(u) \in \mathcal{P}^0 \text{ konstant}$$

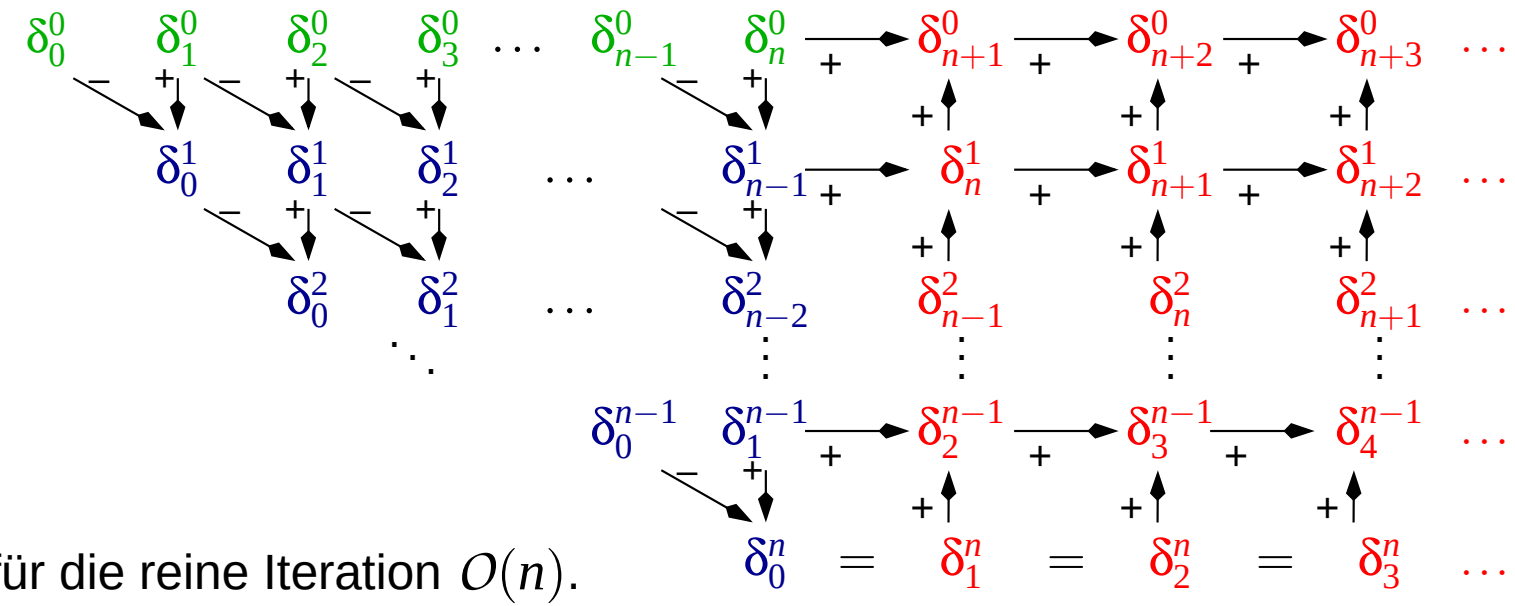
Algorithmus: Vorwärts-Differenzen

Initialisierung: Berechne ausgehend von u_0 und festem $\Delta > 0$:

1. Punkte auf der Kurve: $\delta_j^0 := \mathbf{F}(u_j) = \mathbf{F}(u_0 + j\Delta),\ j = 0, \ldots, n$
2. Differenzen $\delta_j^i := \delta^i(u_j) = \delta^{i-1}(u_{j+1}) - \delta^{i-1}(u_j),\ i = 1, \ldots, n, ; j = 0, \ldots, n-i$

Iteration: Sukzessive Berechnung der $(m+1)$-ten Spalte ($m \geq n$):

$$\delta^n_{(m+1)-n} = \delta^n_{m-n} = \text{const}, \quad \delta^r_{(m+1)-r},\ r = 0, \ldots, n-1$$



Komplexität für die reine Iteration $O(n)$.

Beispiel: Beispiel Vorwärts-Differenzen

Gegeben: $f(u) = u^3 - 4u^2 - 2u,\ u_0 = 0, \Delta = 1$

Lösung:

u :	0	1	2	3	4	5	6	7	8
δ_j^0 :	0	-5	-12	-15	-8	15	60	133	240
δ_j^1 :		-5	-7	-3	7	23	45	73	107
δ_j^2 :			-2	4	10	16	22	28	34
δ_j^3 :				6	6	6	6	6	6

Umwandlung Bézier nach Polynome

Erinnerung: Jedes Polynom $\in \mathcal{P}^n$ läßt sich in der Bernstein-Bézier-Basis darstellen:

$$\text{span}\{1, u, u^2, \ldots, u^n\} = \text{span}\{B_0^n(u), \ldots, B_n^n(u)\}$$

Basis-Transformation: Die Umwandlung der Darstellung erfolgt über eine $(n+1) \times (n+1)$-Matrix.

Beispiel $n = 3$: $\mathbf{C}(u) = \sum_{i=0}^{3} \mathbf{C}_i B_i^3(u) = \sum_{i=0}^{3} \mathbf{A}_i u^i$

$$\begin{pmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \end{pmatrix} \text{ da } \begin{aligned} B_0^3(u) &= (1-u)^3 &&= -u^3 + 3u^2 - 3u + 1 \\ B_1^3(u) &= 3(1-u)^2 u &&= 3u^3 - 6u^2 + 3u \\ B_2^3(u) &= 3(1-u)u^2 &&= -3u^3 + 3u^2 \\ B_3^3(u) &= u^3 &&= u^3 \end{aligned}$$

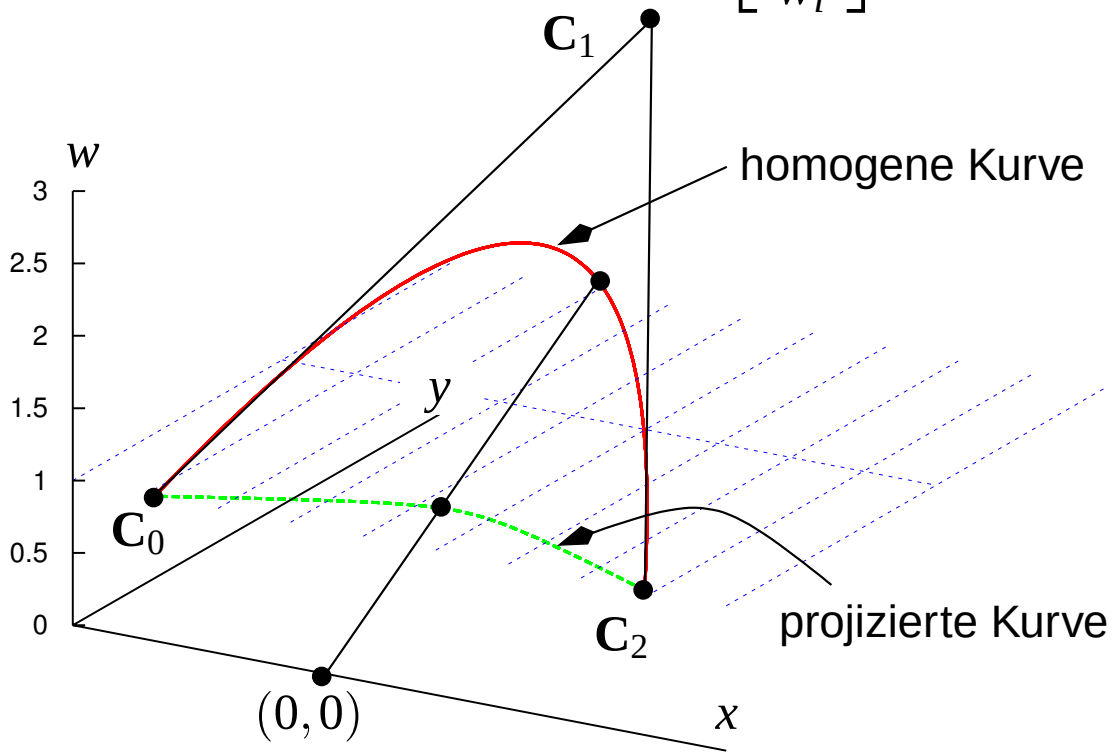
Rationale Bézier-Kurven

Bislang: Nur Polynomkurven darstellbar

Ziel: Exakte Kreissegmente und Kegelschnitte

Ansatz: Zeichne Kurve in *Homogenen Koordinaten* und projiziere Kurve auf $w = 1$:

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



Einfluß der Gewichte

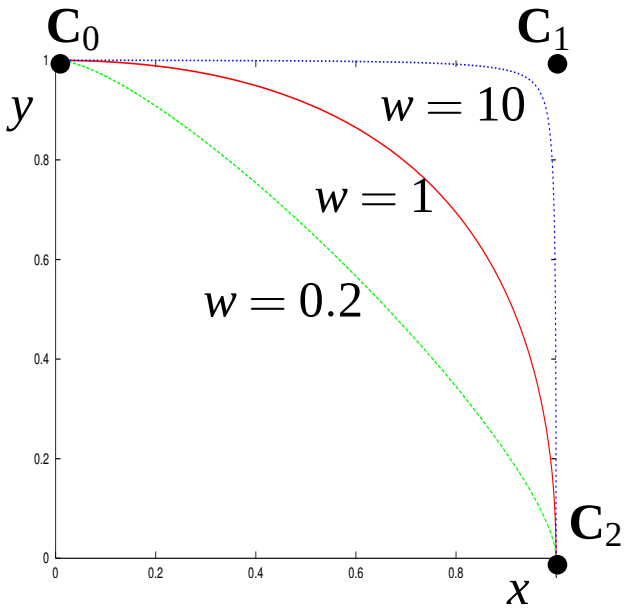
Allgemein: Je größer w_i (bei konstanten w_j, $j \neq i$), desto näher verläuft $\mathbf{C}(u)$ bei $\mathbf{C}_i$

Spezialfall Kegelschnitte für $n = 2$ und $w_0 = w_2 = 1$ ergibt sich durch Variation von w_1:

$$\mathbf{C} \text{ ist Segment einer } \left\{ \begin{array}{c} \text{Ellipse} \\ \text{Parabel} \\ \text{Hyperbel} \end{array} \right\} \Longleftrightarrow \left\{ \begin{array}{l} w_1 < 1 \\ w_1 = 1 \\ w_1 > 1 \end{array} \right.$$

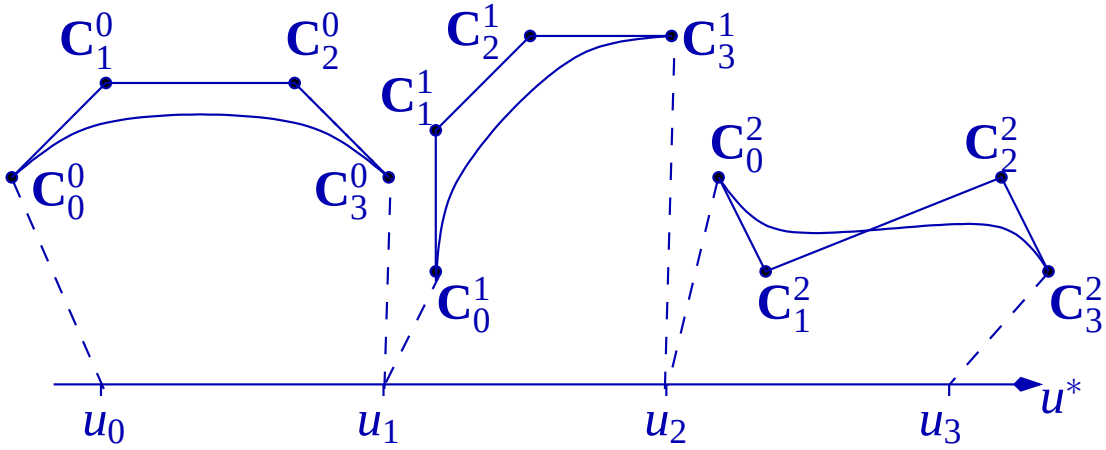
Spezialfall Kreissegment für $n = 2$:

$$w_0 = w_2 = 1 \text{ und } w_1 = 1/\sqrt{2}$$



Allgemeines Parameter-Intervall

Ziel: Verwendung beliebiger Parameter-Intervalle $[a,b]$ statt $[0,1]$, z.B. für stückweise Kurven



Umparametrisierung: Lineare Transformation $\phi : I = [a,b] \rightarrow [0,1]$ ändert Polynomgrad nicht:

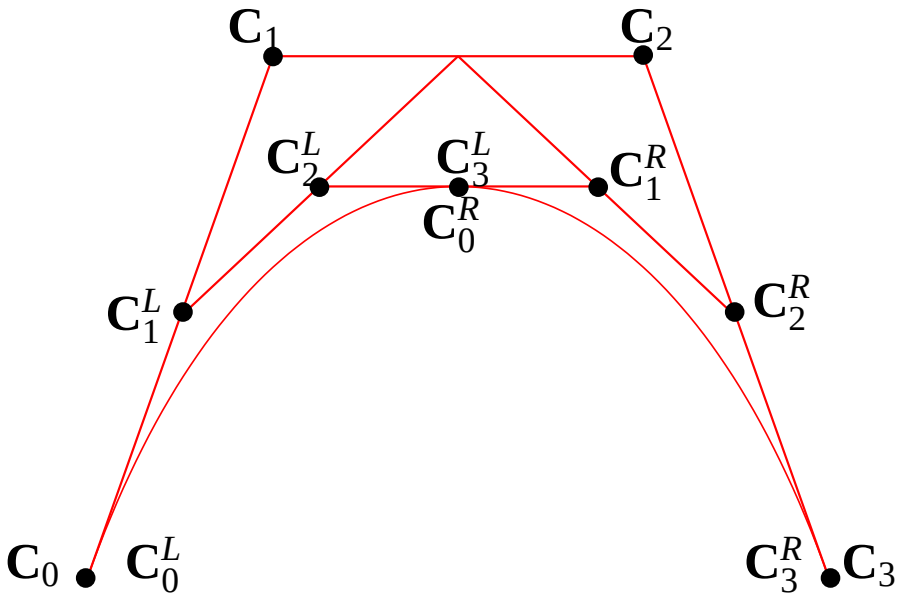
$$\phi(u) := \frac{u-a}{b-a} \in [0,1],\ \forall u \in I \text{ und}$$

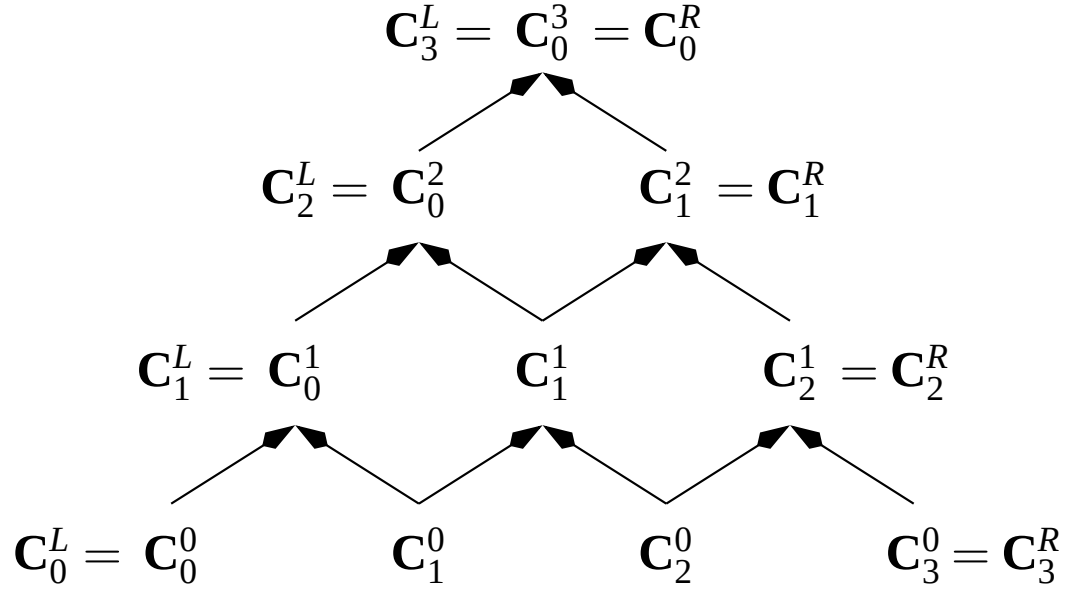
$$B_i^{I,n}(u) := B_i^n(\phi(u)) = \binom{n}{i} \left(\frac{u-a}{b-a}\right)^i \left(\frac{b-u}{b-a}\right)^{n-i}$$

Beachte: Die Geometrie der Kurve bleibt unverändert!

Unterteilung der Bézier-Kurve $\mathbf{C}(u)$

Auswertung von $\mathbf{C}$ bei $a \in [0,1]$ liefert Bézier-Darstellung der Teilkurven auf $[0,a]$ bzw. $[a,1]$:





Beachte: Neue Bézier-Kurven beschreiben nur einen Teil der ursprünglichen Geometrie

Schnitt zwischen Bézier-Kurven

Konvexe-Hülle der Kontrollpunkte disjunkt ⇒ Bézier-Kurven schneiden sich nicht

⇒ Rekursive Unterteilung der Bézier-Kurven liefert Schnittpunkte

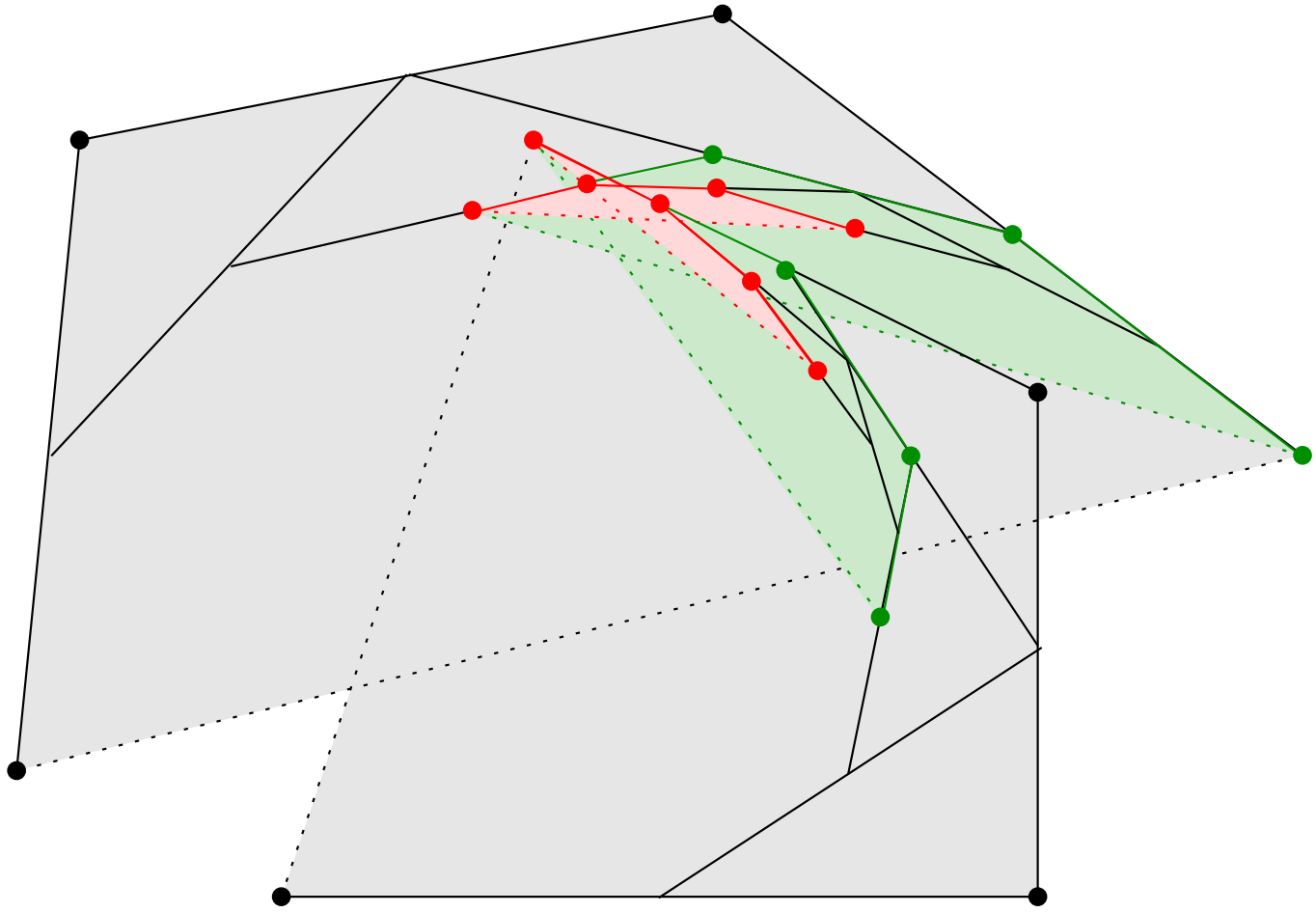
Algorithmus: Zum Schnitt von Bézier-Kurven vom Grad n

```
void intersectCurve(Point cpA[], cpB[], vector intersectPts) {
  Point cpALeft[n+1], cpARight[n+1], cpBLeft[n+1], cpBRight[n+1];

  if ( ! intersectConvexHull(cpA, cpB) ) return;

  if ( sizeOfBBox(cpA) > epsilon ) {
    subdivideCurve(cpA, cpALeft, cpARight);
    intersectCurve(cpALeft,  cpB, intersectPts);
    intersectCurve(cpARight, cpB, intersectPts);
  }
  else if ( sizeOfBBox(cpB) > epsilon ) {
    subdivideCurve(cpB, cpBLeft, cpBRight);
    intersectCurve(cpA, cpBLeft,  intersectPts);
    intersectCurve(cpA, cpBRight, intersectPts);
  }
  else {   // both boxes smaller than epsilon
    intersectPts.append(computeCenter(cpA));
  }
  return;
}
```

Beispiel: Schnitt zwischen Bézier-Kurven



● Original Kontrollpunkt ● Kontrollpunkt 1.Subd. ● Kontrollpunkt 2.Subd.

Bewertung Bézier-Kurven

Wesentliche Vorteile:

- Intuitive Kontrolle des Kurvenverlaufes
- stabile Berechnung (de Casteljau); alternativ schnelle, äquidistante Auswertung (Vorw.-Diff.)
- affine Invarianz und konvexe Hülle
- Darstellung aller Polynome und Kegelschnitte

Wesentliche Nachteile:

- Globaler Einfluß der Kontrollpunkte
- Polynomkurven (integrale Bézier-Kurven) oder Kegelschnitte (rationale Bézier-Kurven)
- Anzahl Freiheitsgrade = Anzahl Kontrollpunkte = Polynomgrad + 1
  $\Longrightarrow$ mehr Freiheitsgrade durch höheren Polynomgrad, aber mehr Rechenaufwand!

3.2 Hermite-Kurven

Bézier-Kurven *interpolieren* $\mathbf{C}_0, \mathbf{C}_n$ und *approximieren* $\mathbf{C}_i, i \notin \{0, n\}$

Hermite-Kurven *interpolieren* Position und Ableitungen in den Endpunkten

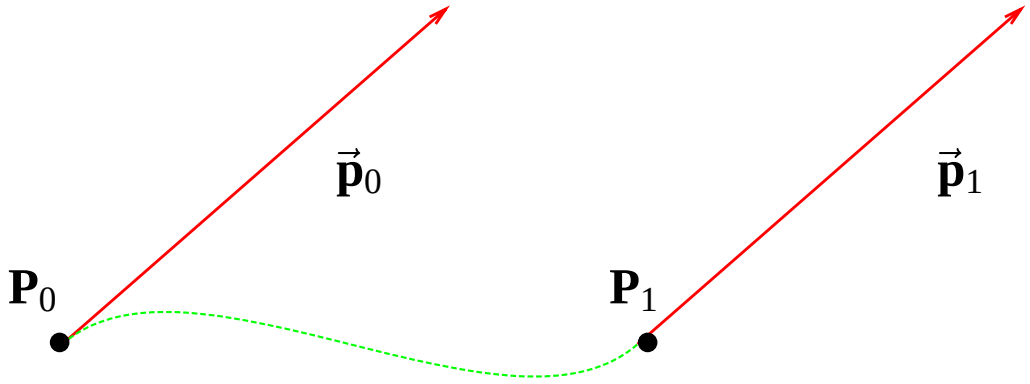
Zahl der Kontrollparameter legt den Grad der Kurve fest:

1. Ableitung $\longrightarrow$ 4 Parameter $\longrightarrow$ kubische Polynome

1. und 2. Ableitung $\longrightarrow$ 6 Parameter $\longrightarrow$ quinitische Polynome etc.

Kubische Hermite-Kurve $\mathbf{H}(u)$ interpolieren $\mathbf{P}_0, \mathbf{P}_1, \vec{\mathbf{p}}_0, \vec{\mathbf{p}}_1$:

$$\mathbf{H}(0) = \mathbf{P}_0, \quad \mathbf{H}(1) = \mathbf{P}_1, \quad \mathbf{H}'(0) = \vec{\mathbf{p}}_0, \quad \mathbf{H}'(1) = \vec{\mathbf{p}}_1$$



Kubische Hermite-Basisfunktionen

Kubische Hermite-Kurven definiert sich zu

$$\mathbf{H}(u) = \mathbf{P}_0 h_0(u) + \vec{\mathbf{p}}_0 h_1(u) + \vec{\mathbf{p}}_1 h_2(u) + \mathbf{P}_1 h_3(u)$$

Kubische Hermite-Basisfunktionen:

$$h_0(u) = 2u^3 - 3u^2 + 1$$

mit $h_0(0) = 1, h_0(1) = 0, h_0'(0) = 0, h_0'(1) = 0$

$$h_1(u) = u^3 - 2u^2 + u$$

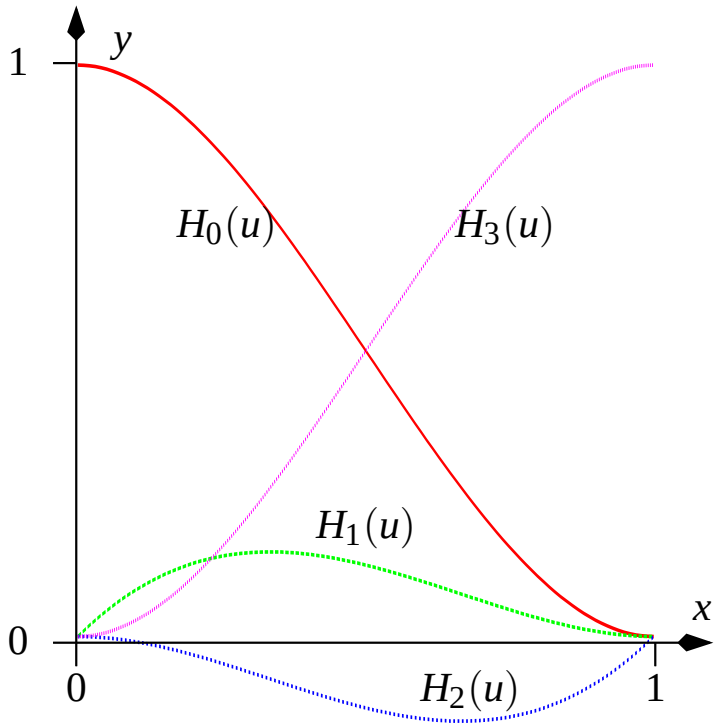
mit $h_1(0) = 0, h_1(1) = 0, h_1'(0) = 1, h_1'(1) = 0$

$$h_2(u) = u^3 - u^2$$

mit $h_2(0) = 0, h_2(1) = 0, h_2'(0) = 0, h_2'(1) = 1$

$$h_3(u) = -2u^3 + 3u^2$$

mit $h_3(0) = 0, h_3(1) = 1, h_3'(0) = 0, h_3'(1) = 0$



Bezeichnung: Bézier-Splinekurve

Bézier-Splinekurve: Mehr Freiheitsgrade durch *Aneinanderreihung einzelner Bézier-Kurven*

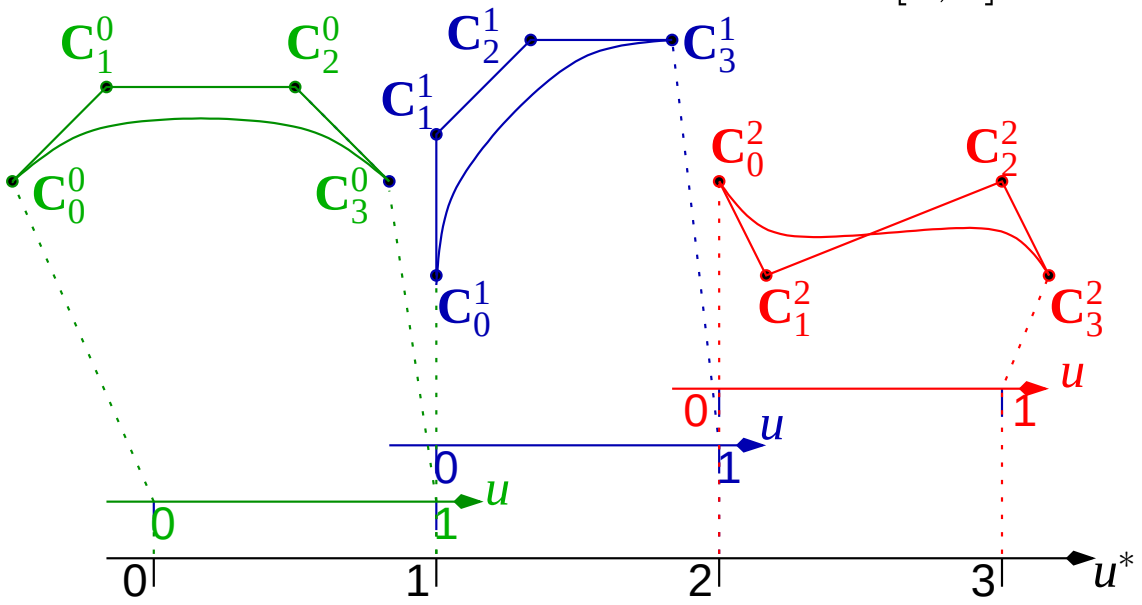
Konkret: Bézier-Splinekurve vom Grad n mit $k+1$ Segmenten $\mathbf{C}^j(u),\ j=0,\ldots,k$

Lokaler Parameter: Jedes Segment $\mathbf{C}^j$ hat als Bézier-Kurve den Parameter $u \in [0,1]$

Globaler Parameter: Parameter $u^* \in [0, k+1]$ der gesamten Splinekurve $\mathbf{C}$ mit

$$u^* \in [j, j+1[$$

$$\Longrightarrow \mathbf{C}(u^*) = \mathbf{C}^j(u) \text{ mit } u = u^* - j$$



Frage: Welche Bedingungen müssen gelten, damit $\mathbf{C}$ eine stetige und/oder glatte Kurve ist?

Anschlußbedingungen

Gegeben: Kurve $\mathbf{C}^0$ mit Kontrollpunkten $\mathbf{C}_0^0, \ldots, \mathbf{C}_n^0$

Gesucht: Kontrollpunkte $\mathbf{C}_i^1$ von $\mathbf{C}^1$, so dass $\mathbf{C}^1$ stetig/glatt an $\mathbf{C}^0$ in $\mathbf{C}^0(1)$ anschließt.

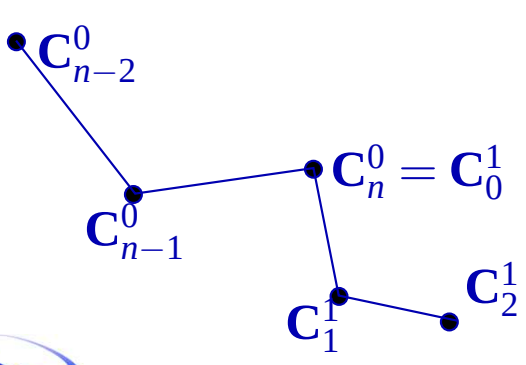
C^0-stetig: Es muß $\mathbf{C}^0(1) = \mathbf{C}^1(0)$ gelten, also $\mathbf{C}_n^0 = \mathbf{C}_0^1$ (Endpunktinterpolation)

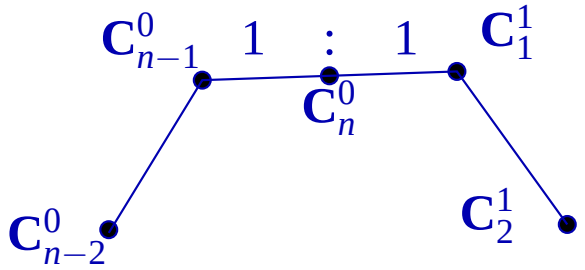
C^1-stetig: C^0-stetig und $(\mathbf{C}^0)'(1) = (\mathbf{C}^1)'(0)$ (Interpolation der Endtangenten)

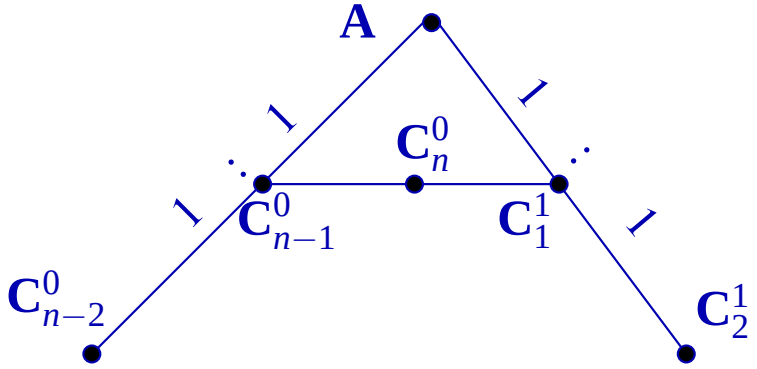
$$(\mathbf{C}^0)'(1) = n\left(\mathbf{C}_n^0 - \mathbf{C}_{n-1}^0\right) \stackrel{!}{=} n\left(\mathbf{C}_1^1 - \mathbf{C}_0^1\right) = (\mathbf{C}^1)'(0) \overset{C^0\text{-stetig}}{\Longleftrightarrow} \mathbf{C}_1^1 = \mathbf{C}_0^1 + (\underbrace{\mathbf{C}_n^0}_{=\mathbf{C}_0^1} - \mathbf{C}_{n-1}^0)$$

C^2-stetig: C^1-stetig und $(\mathbf{C}^0)''(1) = (\mathbf{C}^1)''(0)$ (*krümmungsstetig*)

A-Frame-Konstruktion: $\mathbf{C}_2^1 = \mathbf{C}_1^1 + (\mathbf{C}_1^1 - \underbrace{(\mathbf{C}_{n-1}^0 + (\mathbf{C}_{n-1}^0 - \mathbf{C}_{n-2}^0))}_{=\mathbf{A}})$







3.3.1 Catmull-Rom-Ansatz

Ansatz: Verwendung von *kubische* C^1-stetigen Bézier-Splines (vgl. Hermite-Kurven!)

Kontrollgrößen: Interpoliert werden

1. Kontrollpunkten $\mathbf{C}_0^j,\ j=0,\ldots,k$ und $\mathbf{C}_3^k$
2. Tangenten $\vec{\mathbf{t}}_j = (\mathbf{C}^j)'(0),\ j=0,\ldots,k$ und $\vec{\mathbf{t}}_{k+1} = (\mathbf{C}^k)'(1)$

Problem: Die Vorgabe von zwei Kontrolltypen (Punkte, Tangenten) ist unhandlich.

Catmull-Rom-Splines: Vorgabe der Kontrollpunkte & Abschätzung der Tangenten.

Gegeben: Interpolationspunkte $\mathbf{P}_i,\ i=0,\ldots,k+1$

Berechnung der Bézier-Punkte: Für das j-te Bézier-Segment ergibt sich

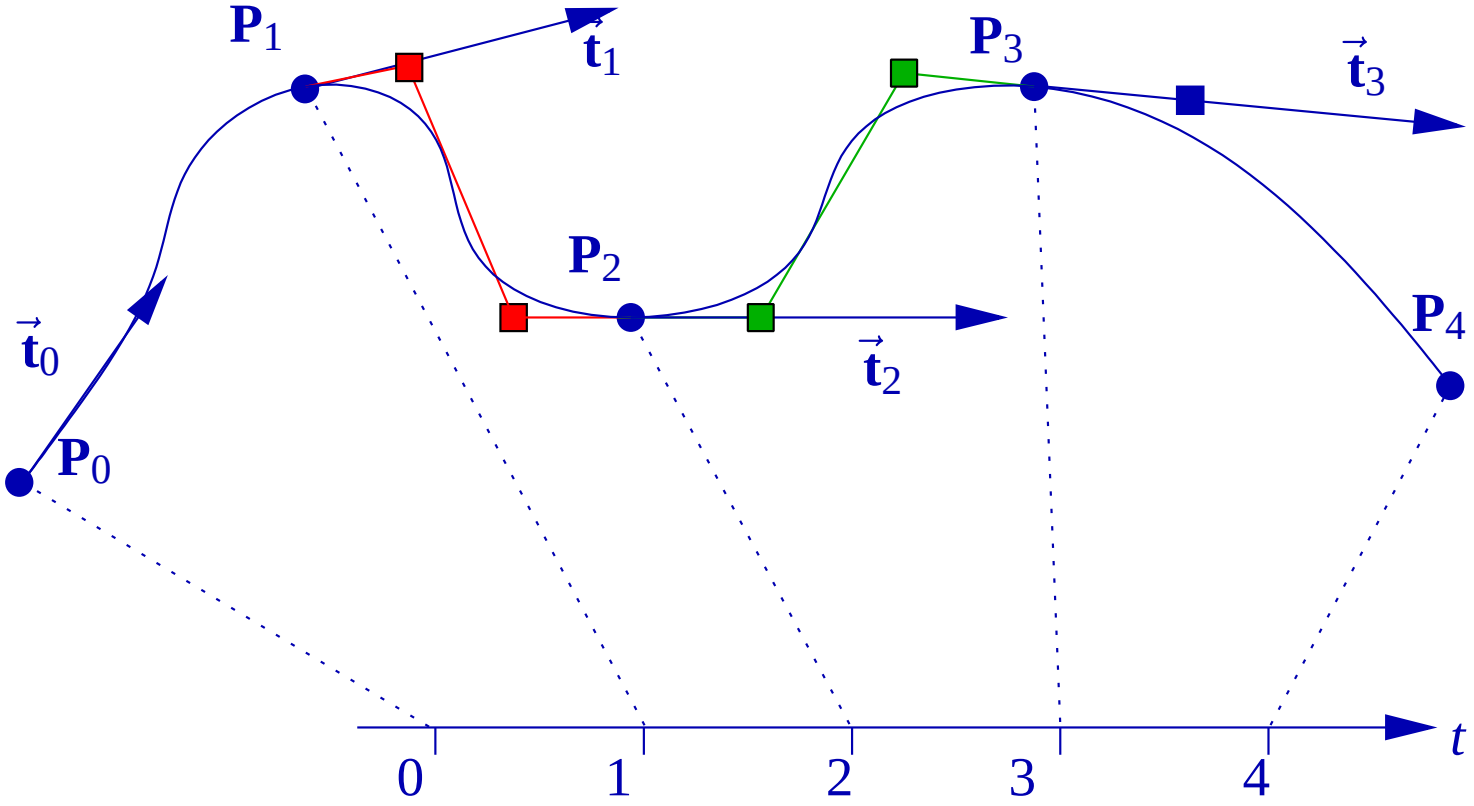
Endpunkte: $\mathbf{C}_0^j = \mathbf{P}_j, \mathbf{C}_3^j = \mathbf{P}_{j+1}, \qquad j=0,\ldots,k$

Tangente bei $\mathbf{P}_j$: $\vec{\mathbf{t}}_j = \frac{1}{2}\left(\mathbf{P}_{j+1} - \mathbf{P}_{j-1}\right),\ j=1,\ldots,k-1$

Tangenten am Rand: $\mathbf{P}_{-1}$ und $\mathbf{P}_{k+2}$ zusätzlich vorgeben, z.B. $\mathbf{P}_{-1} = \mathbf{P}_0$

Innere Kontrollpunkte: $\mathbf{C}_1^j - \mathbf{C}_0^j = \frac{1}{3}\vec{\mathbf{t}}_j \Rightarrow \mathbf{C}_1^j = \mathbf{C}_0^j + \frac{1}{3}\vec{\mathbf{t}}_j$, analog $\mathbf{C}_2^j = \mathbf{C}_3^j - \frac{1}{3}\vec{\mathbf{t}}_{j+1}$

3.3.1 Catmull-Rom-Ansatz



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

Interpolation der Endpunkte: $\mathbf{C}_0^j = \mathbf{P}_j,\ \mathbf{C}_3^j = \mathbf{P}_{j+1}$

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

3.3.1 Catmull-Rom-Ansatz

Bewertung Bézier-Splines

Wesentliche Vorteile:

- Alle Vorteile der Bézier-Kurven
- Zusätzliche Freiheitsgrade bei konstantem Grad

Wesentliche Nachteile:

- Erfüllung expliziter Übergangsbedingungen **oder**
- verschiedenartige Kontrollparameter (Catmull-Rom)

Wünschenswert:

- „Eingebaute“ Übergangsbedingung
- maximale Stetigkeit bei minimalem Einfluß der Kontrollpunkte

Ansatz: Quadratische B-Spline Kurven

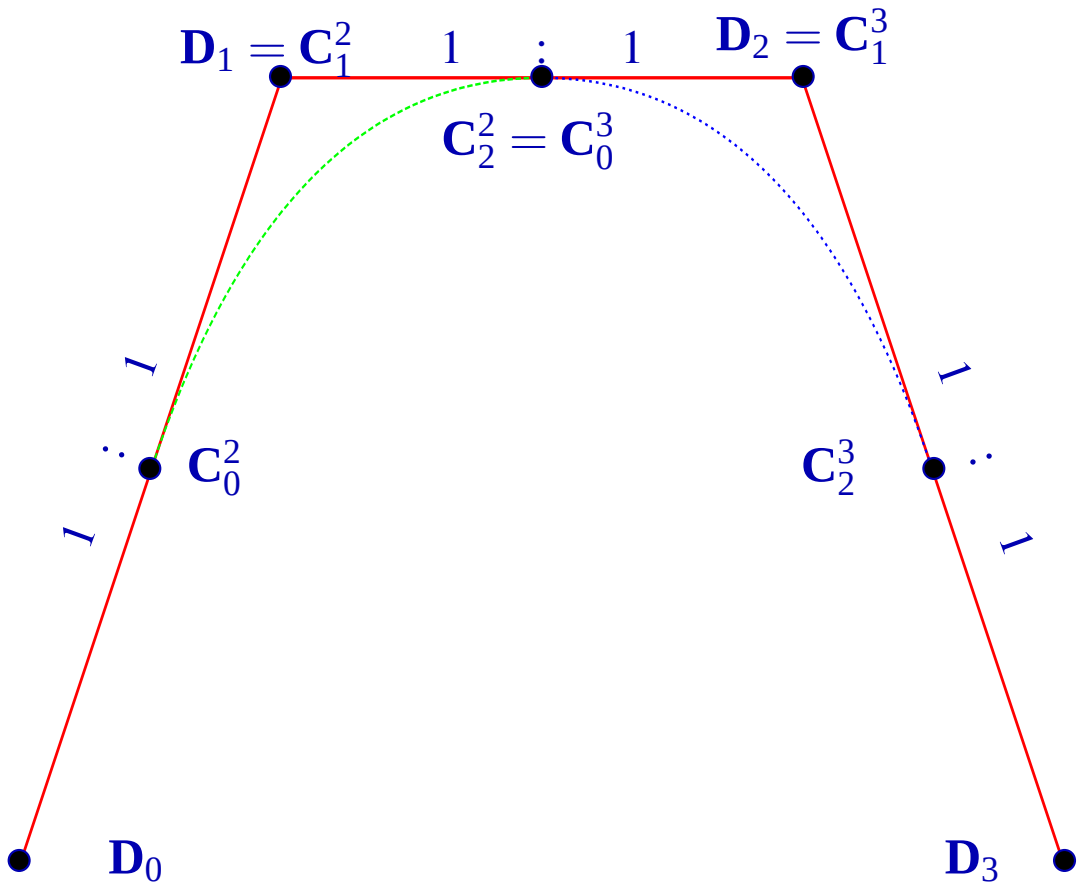
Idee: Nutze innere Kontrollpunkte C^1-stetiger quadratischer Bézier-Splines als Kontrollpunkte

Gegeben: „Innere Kontrollpunkte“ $\mathbf{D}_1, \ldots, \mathbf{D}_k$

Konstruktion: Sequenz

$\mathbf{D}_{j-2}, \ldots, \mathbf{D}_j$ definiert die quadratische Bézier-Kurve $\mathbf{C}^j$:

$$\begin{aligned} \mathbf{C}_0^j &= \frac{1}{2}\mathbf{D}_{j-2} + \frac{1}{2}\mathbf{D}_{j-1}, \\ \mathbf{C}_1^j &= \mathbf{D}_{j-1} \\ \mathbf{C}_2^j &= \frac{1}{2}\mathbf{D}_{j-1} + \frac{1}{2}\mathbf{D}_j \end{aligned}$$



Beachte: Automatisch C^1-stetig

Ansatz: Kubische B-Spline Kurven

Idee: Nutze A-Frame-Punkte C^2-stetiger kubischer Bézier-Splines als Kontrollpunkte

Gegeben: A-Frame Punkte $\mathbf{D}_1, \ldots, \mathbf{D}_k$

Konstruktion: Sequenz

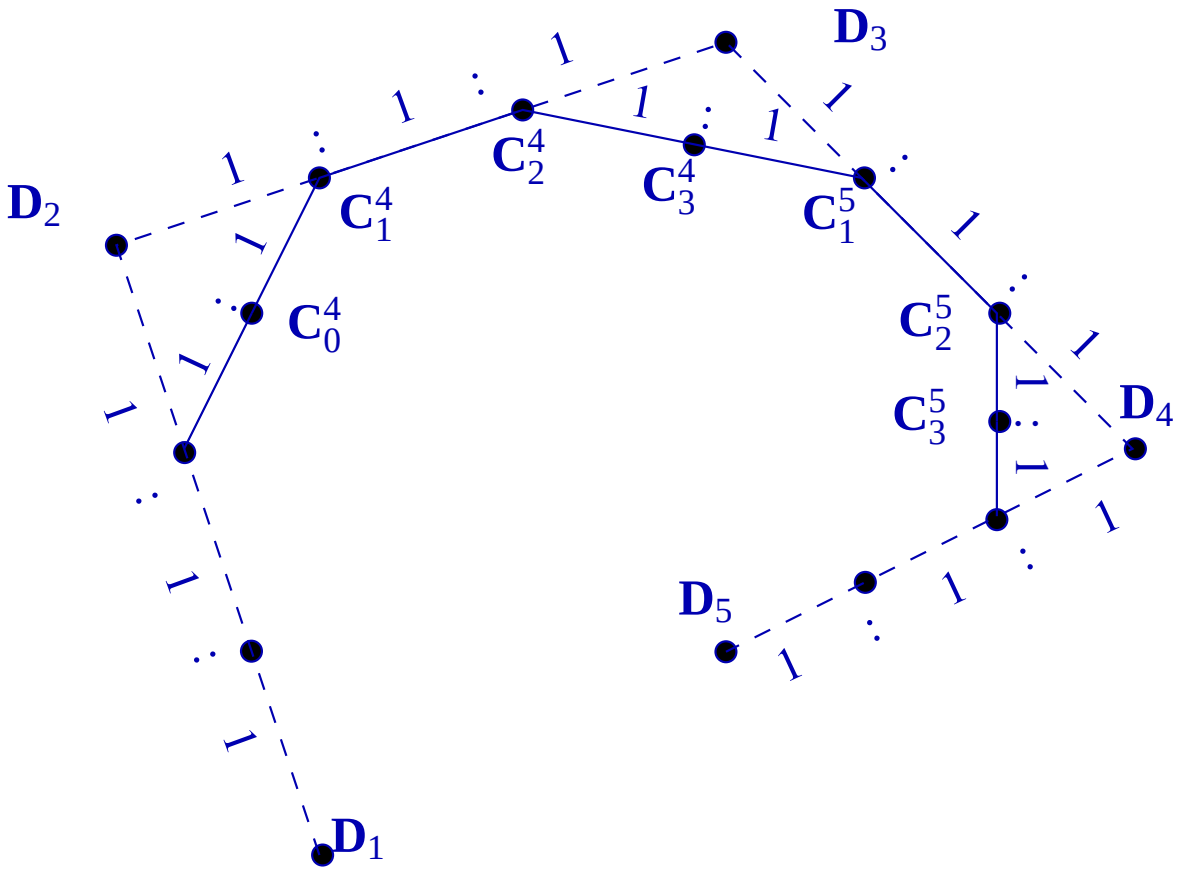
$\mathbf{D}_{j-3}, \ldots, \mathbf{D}_j$ definiert kubische Bézier-Kurve $\mathbf{C}^j$:

$$\mathbf{C}_1^j = \frac{2}{3}\mathbf{D}_{j-2} + \frac{1}{3}\mathbf{D}_{j-1}$$

$$\mathbf{C}_2^j = \frac{1}{3}\mathbf{D}_{j-2} + \frac{2}{3}\mathbf{D}_{j-1}$$

$$\mathbf{C}_0^j = \frac{1}{2}\mathbf{C}_2^{j-1} + \frac{1}{2}\mathbf{C}_1^j$$

$$\mathbf{C}_3^j = \frac{1}{2}\mathbf{C}_2^j + \frac{1}{2}\mathbf{C}_1^{j+1}$$



Bemerkung: Quadratische und kubische B-Splines

Obige Einführung ist sehr speziell, denn

- wie sehen B-Splines für beliebigen Grad n und beliebige Parameterintervalle aus?
- Auswertung ohne Umweg über die Bézier-Darstellung?

Basisfunktionen: Angenommen, obige B-Spline-Kurven haben die Darstellung

$$\mathbf{D}(u) = \sum_i \mathbf{D}_i N_i^n(u) \text{ für unbekannte Basisfunktionen } N_i^n, \quad n = 2,3$$

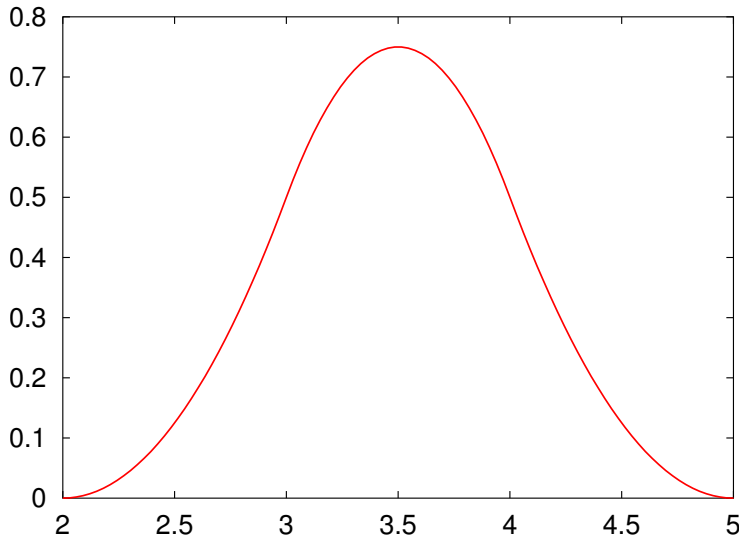
Für ein i_0 gewinnt man $N_{i_0}^n$ durch: $\mathbf{D}_i = 0,\ i \neq i_0$ und $\mathbf{D}_{i_0} = 1$.

Beispiel: Bestimmung von $N_2^2(u)$ als stückw. quadratisches Polynom (lok. Param. für $\mathbf{C}^j : u = u^* - j$):

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

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

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



3.4 B-Spline-Kurven

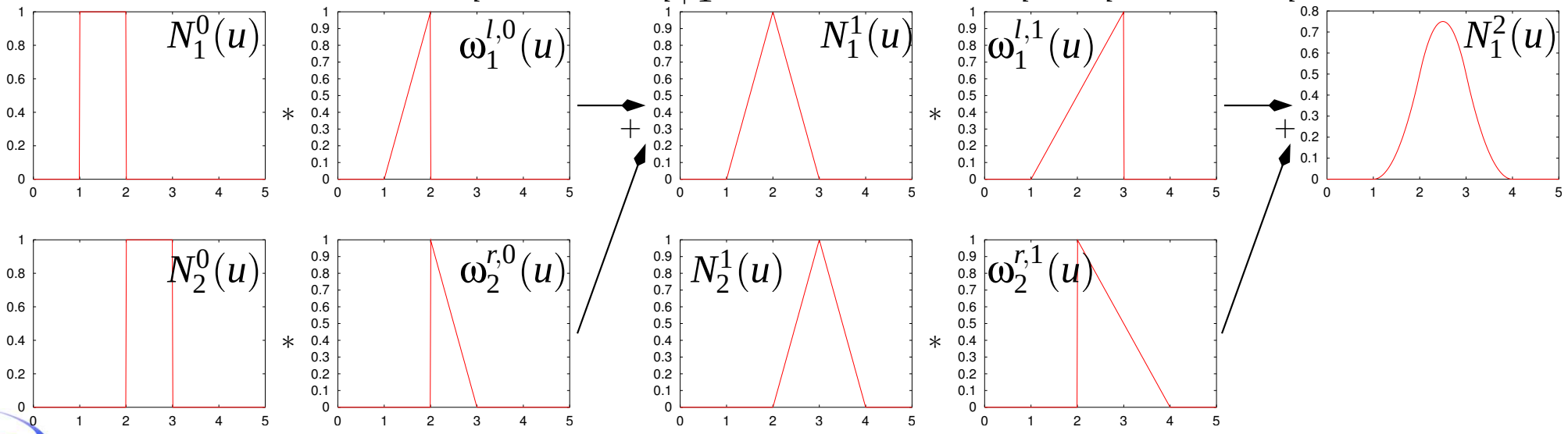
Ziel: Basisfunktion als stückweises Polynom $N_i^n(u)$ mit gegebenem Grad $n \in \mathbb{N}$, so dass

1. max. Stetigkeit zwischen den Teilkurven, also $(n-1)$**-stetig**
2. möglichst geringer Einflußbereich der Kontrollpunkte (**Lokalität**)
3. Erlegung der Eins: $\sum N_i^n(u) \equiv 1,\ N_i^n(u) \geq 0$ ($\Rightarrow$ affine Invarianz, konvexe Hülle)

Knotenvektor: $T = \{\ldots, t_{i-1}, t_i, t_{i+1}, \ldots\},\ t_i \leq t_{i+1},\ t_i \in \mathbb{R}$ mit: $N_i^n \big|_{[t_i, t_{i+1}[} \in \mathcal{P}^n$

Fall n=0: Offensichtlich sinnvolle Festlegung $N_i^0 = \chi([t_i, t_{i+1}[)$

Rekursion: Gewichtung von $N_i^n(u)$ und $N_{i+1}^n(u)$ mit linearen $\omega_i^{l,n}, \omega_i^{r,n}$ ergibt $N_i^{n+1}(u)$



Definition: Normalisierte B-Spline-Basisfunktionen

Gegeben: Knotenvektor $T = \{\ldots, t_{i-1}, t_i, t_{i+1}, \ldots\}, t_i \leq t_{i+1}, t_i \in \mathbb{R}$

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

Rekursion $n \rightarrow n+1$**:**

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

Bemerkung:

- $\omega_i^{l,n}, \omega_i^{r,n}$ steigt/fällt auf $\text{supp}(N_i^n)$ von 0 nach 1 an bzw. von 1 nach 0 ab
- N_i^n sind Zerlegung der Eins, da $N_i^n(u) \geq 0$ und per Induktion: $\sum N_i^0 \equiv 1$ und

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

- Knotenvektor T heißt *uniform*, falls $t_i = i$

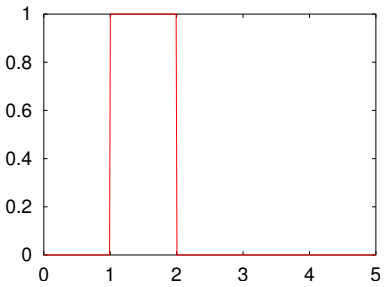
Beispiel: Normalisierte B-Spline-Basisfunktionen

Gegeben: Uniformer Knotenvektor

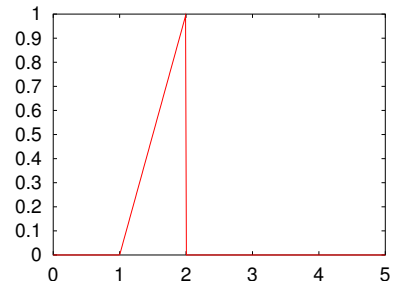
$$n=0\text{:}\ N_i^0 = \chi[i,i+1[= \begin{cases} 1 & \text{falls } u \in [i,i+1[\\ 0 & \text{falls } u \notin [i,i+1[\end{cases}$$

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

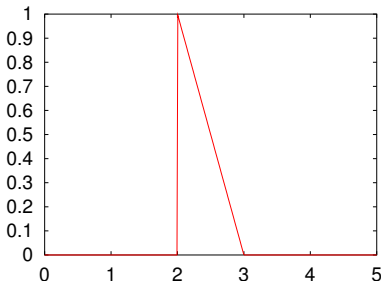
$$N_i^1(u) = (u-i)\cdot\chi[i,i+1[+ (i+2-u)\cdot\chi[i+1,i+2[= \begin{cases} u-i & \text{falls } u \in [i,i+1[\\ i+2-u & \text{falls } u \in [i+1,i+2[\\ 0 & \text{sonst} \end{cases}$$



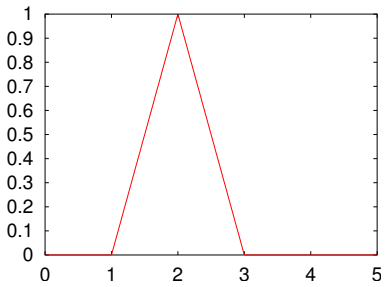
N_1^0



$\omega_1^{l,0}\cdot N_1^0$



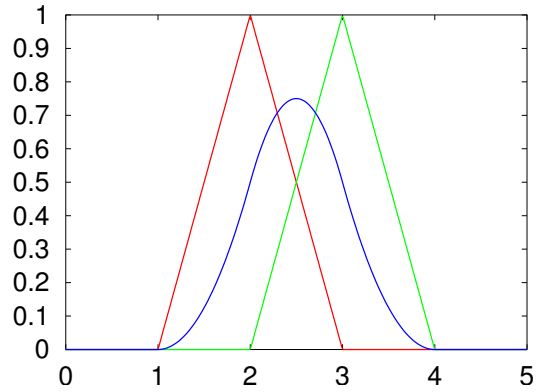
$\omega_2^{r,0}\cdot N_2^0$



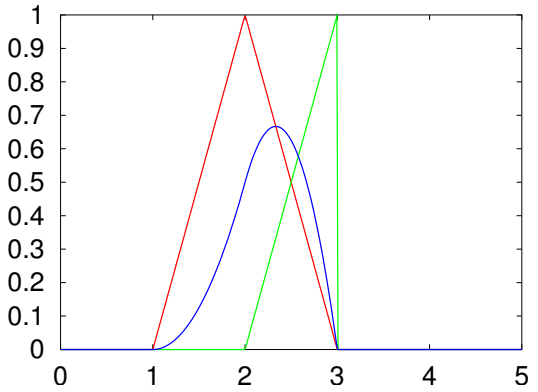
N_1^1

Einfluß der Knoten t_i

1. ein *k-facher Knoten* $t_{i-1} < t_i = t_{i+1} = \ldots = t_{i+k-1} < t_{i+k}$ reduziert die Stetigkeit um k



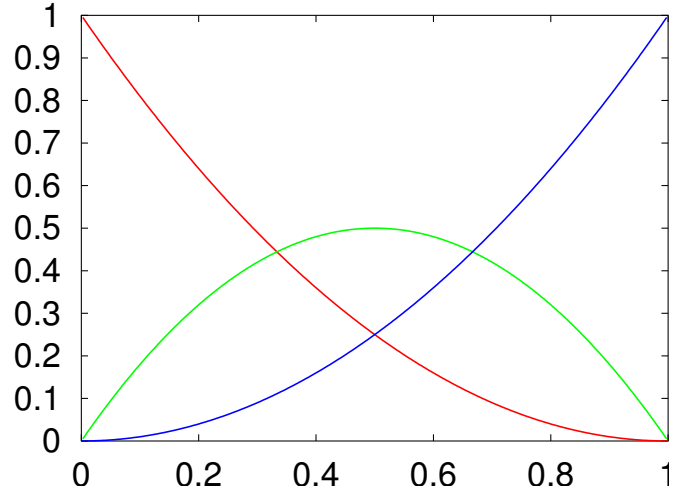
Uniformer Knotenvektor mit N_1^1, N_2^1, N_1^2



Knotenvektor $\{1,2,3,3,4\}$ mit N_1^1, N_2^1, N_1^2

2. $T = \{ \underbrace{0,\ldots,0}_{(n+1)\text{-fach}} , \underbrace{1,\ldots,1}_{(n+1)\text{-fach}} \}$ liefert:

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



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

3. $N_0^n, \ldots, N_m^n$, $m > n$ bilden auf $[t_n, t_{m+1}]$ eine Basis der stückweisen Polynome vom Grad n mit der durch T gegebenen Stetigkeit

B-Spline-Kurven

Eine *B-Spline-Kurve* ist eine Affin-Kombination von *de Boor-Punkten* $\mathbf{D}_i$ mit Gewichten N_i^n zu gegebenem Knotenvektor $T = \{t_0, \ldots, t_{n+m+1}\}$

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

Eigenschaften von B-Spline Kurven

Polynomiell: $\mathbf{D}$ ist eine stückweise Polynomkurve vom Grad n mit C^{n-1}-Übergängen bei disjunkten Knoten bzw. C^{n-k} bei k-fachem Knoten

Lokalität: $\mathbf{D}_i$ beeinflußt Kurve nur lokal, da

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

Affine Invarianz und konvexe Hülle da $\sum_{i=0}^{m} N_i^n = 1$ und $N_i^n \geq 0$ auf $[t_n, t_{m+1}]$

Lokale konvexe Hülle: $\mathbf{D}\big|_{[t_i, t_{i+1}[}$ verläuft in Hülle der $\mathbf{D}_{i-n}, \ldots, \mathbf{D}_i$

Algorithmus: de Boor

Gegeben: de Boor Punkte $\mathbf{D}_0, \ldots, \mathbf{D}_m$ und der Parameter $u \in [t_n, t_{m+1}]$ (beliebig, aber fest)

Gesucht: Kurvenpunkt $\mathbf{D}(u)$ direkt aus den $\mathbf{D}_i$ ohne Berechnung der N_i^n:

Relevante de Boor-Punkte: Für $u \in [t_r, t_{r+1}[$ sind alle $\mathbf{D}_i$ mit $N_i^n(u) \neq 0$ notwendig:

$$\mathbf{D}_i^0 = \mathbf{D}_i,\ i = r-n, \ldots, r$$

Rekursion: Affin-Kombinationen benachbarter Kontrollpunkte:

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

mit Gewichten:

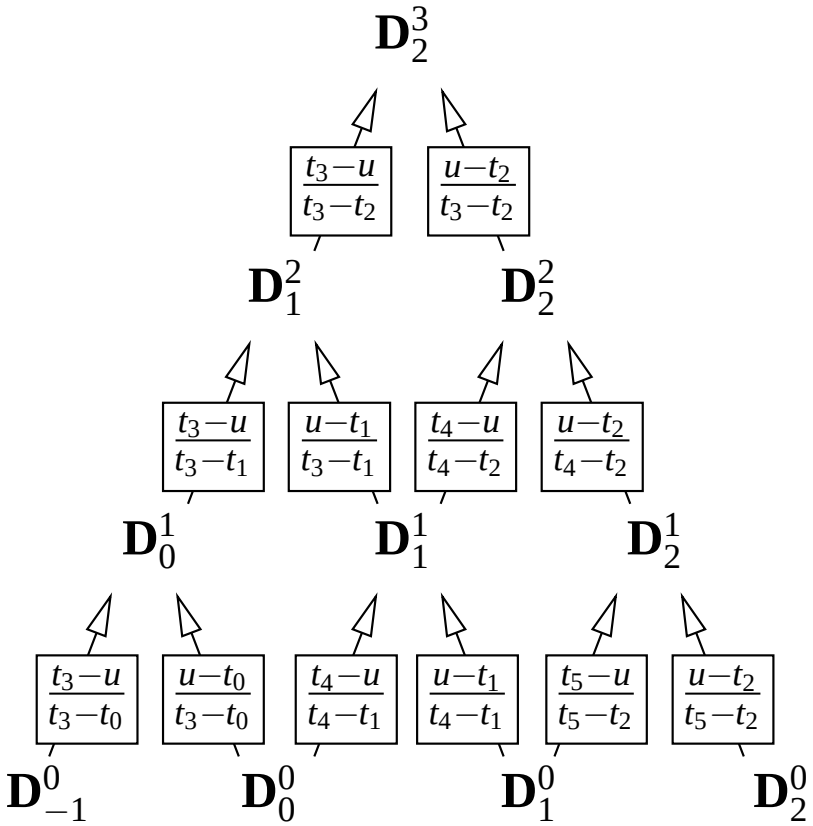
$$\alpha_i^j(u) = \frac{u - t_i}{t_{i+n+1-j} - t_i}, \quad i = r-n+j, \ldots, r,\ j = 1, \ldots, n$$

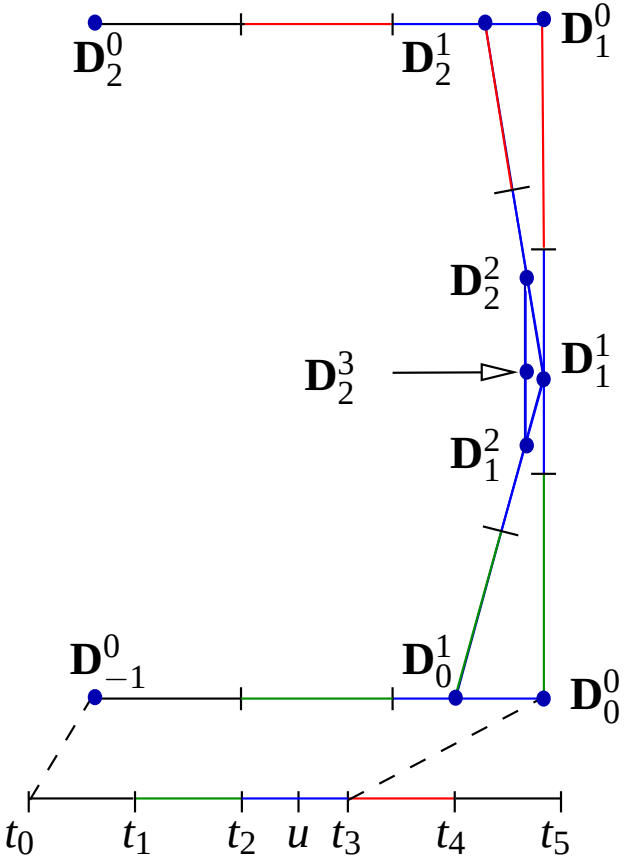
Ergebnis: $\mathbf{D}(u) = \mathbf{D}_r^n$

de Boor: Geometrische Interpretation

Gegeben: de Boor-Punkte $\mathbf{D}_0, \ldots, \mathbf{D}_m$ und $u \in [t_r, t_{r+1}[$

Geometrische Lösung: Hier mit $n = 3$ und $r = 2$



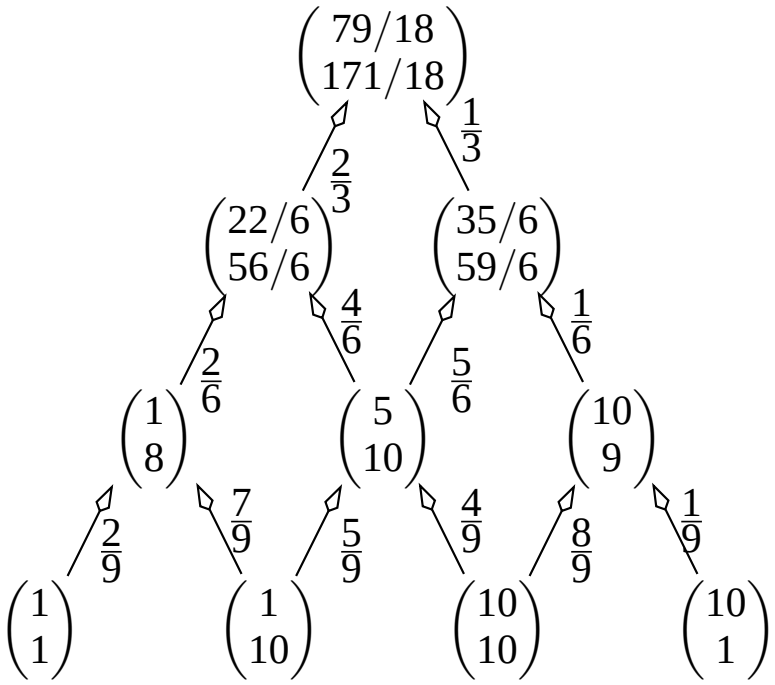


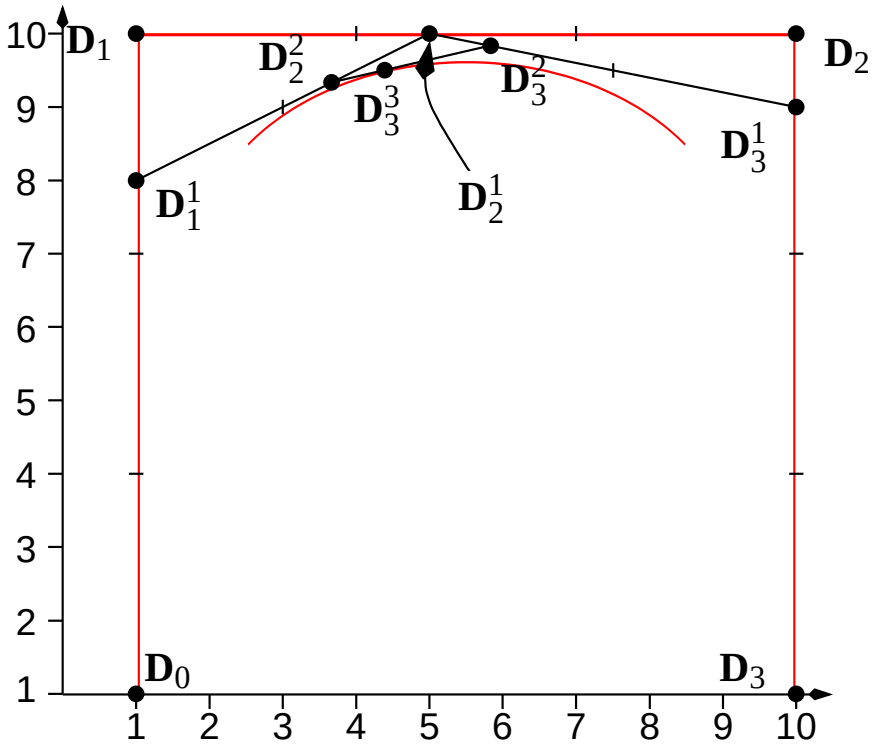
Beispiel: de Boor Algorithmus (uniformer, kubischer Fall)

Gegeben: $\mathbf{D}_0 = (1,1), \quad \mathbf{D}_1 = (1,10), \quad \mathbf{D}_2 = (10,10), \quad \mathbf{D}_3 = (10,1), \quad u = 10/3,\ r = 3$

Gewichte: Ermittlung der Gewichte $\alpha_i^j = \alpha_i^j(10/3) = \frac{u-i}{4-j}$:

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





Ableitung von B-Spline-Kurven

Beachte: Ableitung eines stückw. Polynoms n-ten Grades ist stückw. Polynom vom Grad $n-1$

Darstellung als B-Spline-Kurve: Mit Knotenvektor $T = \{t_0 \leq t_1 \leq \ldots \leq t_{m+n+1}\}$ gilt

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

Knoteneinfügen (nach Böhm '80)

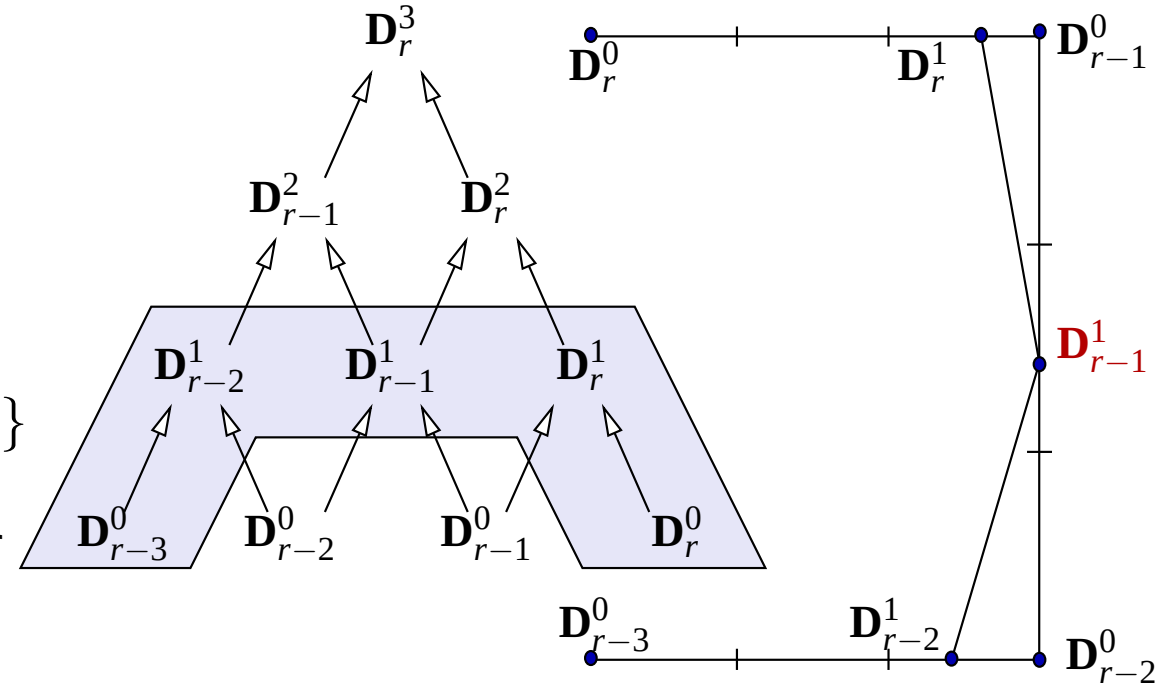
Beachte: Ein stückw. Polynom über

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

ist auch stückw. Polynom über

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

Kontrollpunkte berechnen sich über de Boor-Algorithmus für $u = t^*$



Idee: Tensor-Produkt Flächen

Ausgangslage: Kurvenschema $\mathbf{K}(u) = \sum_{i=0}^{n} \mathbf{P}_i F_i(u)$

Variation der Kontrollpunkte $\mathbf{P}_i$ entlang anderer Kurve

$$\mathbf{P}_i = \mathbf{P}_i(v) = \sum_{j=0}^{m} \mathbf{P}_{ij} F_j(v)$$

TP-Bézier-Flächen: I.a. werden Polynomkurven desselben Grades verwendet:

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

TP-B-Spline-Flächen haben i.a. unterschiedlich viele Segmente in u- und v-Richtung:

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

Eigenschaften von TP-Bézier-Flächen

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

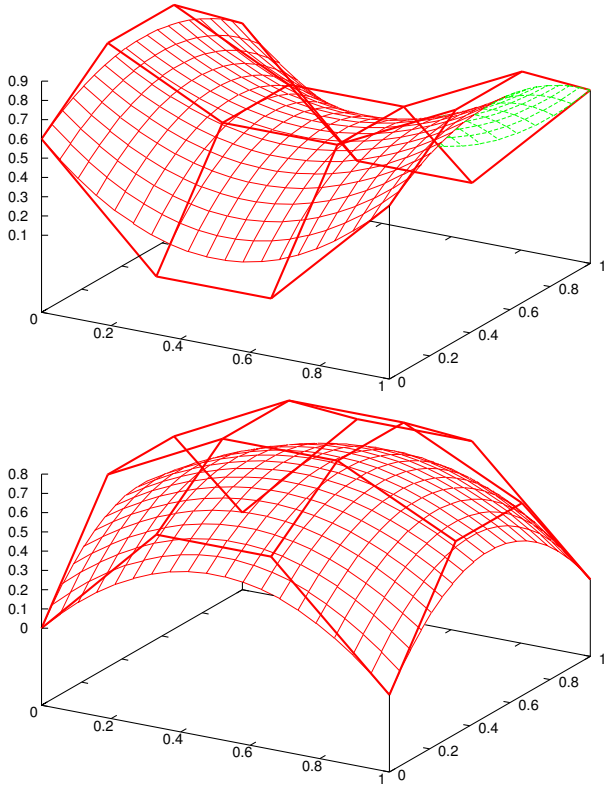
Ableitung:

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

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

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

$$\frac{\partial \mathbf{C}}{\partial u}(0,0) = n\left(\mathbf{C}_{1,0}-\mathbf{C}_{0,0}\right)$$



Globaler Einfluß der $\mathbf{C}_{ij}$: Wie im Kurvenfall, da $B_i^n(u)\cdot B_j^n(v) \neq 0$ auf $(u,v) \in]0,1[^2$

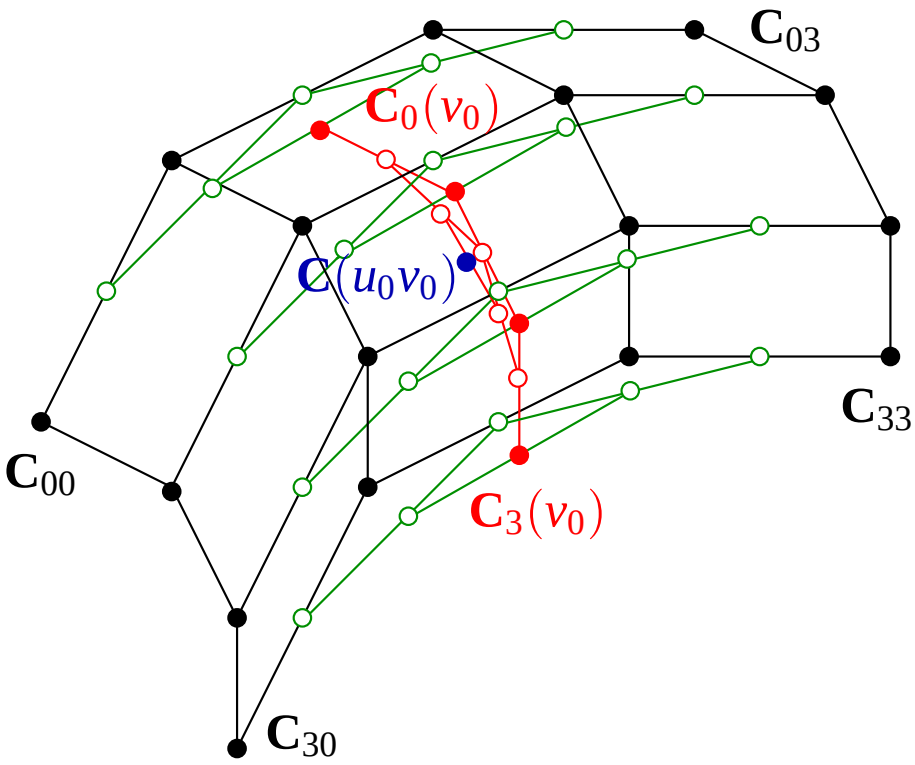
Affine Invarianz und konvexe Hülle gelten, da $\sum_{(i,j)} B_i^n(u)B_j^n(v) = (\sum_i B_i^n(u))(\sum_j B_j^n(v)) = 1$

Auswertung von Bézier-Flächen

Zweistufiger de Casteljau: Separate Auswertung für den 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. $\mathbf{C}(u_0, v_0) = \sum_{i=0}^{n} \mathbf{C}_i(v_0) B_i^n(u_0)$

Beachte: Die Rolle von u und v kann vertauscht werden

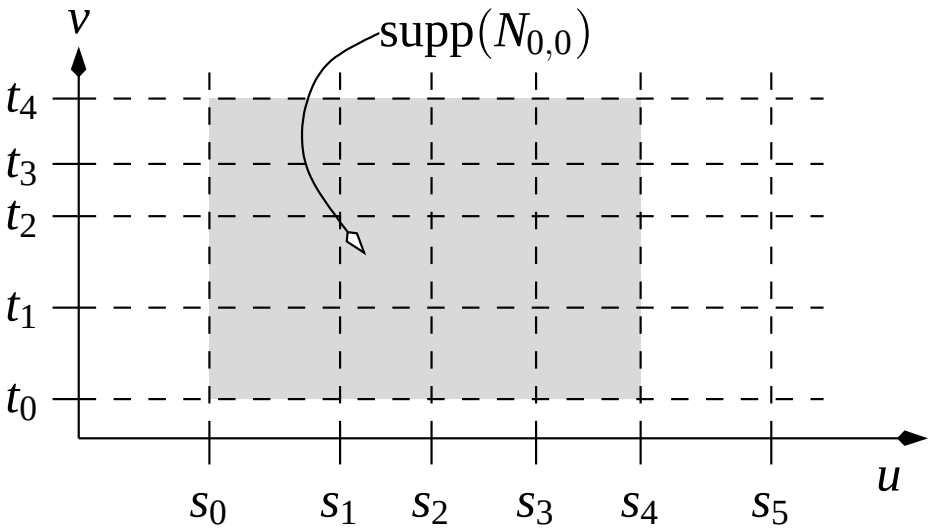


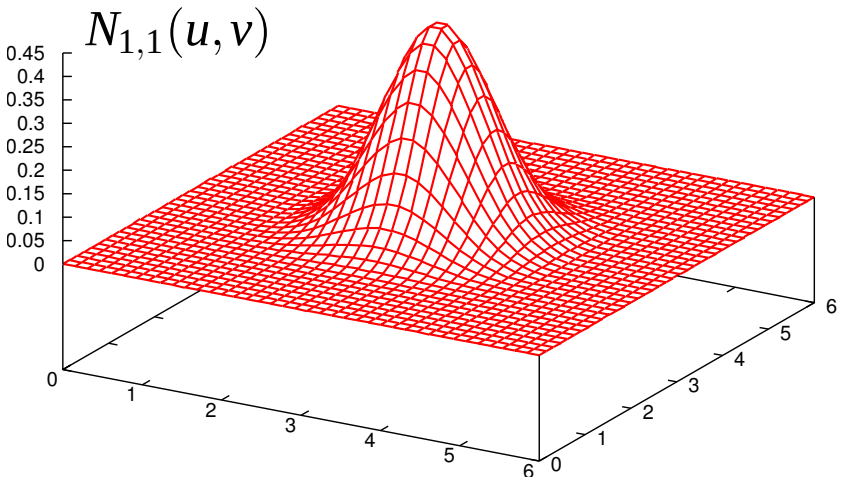
Eigenschaften von TP-B-Spline-Flächen

Knotenvektoren $S = \{s_0, s_1, \ldots, s_{m_u+n+1}\}$, $T = \{t_0, t_1, \ldots, t_{m_v+n+1}\}$ in u- bzw. v-Richtung

Lokaler Einfluß der $\mathbf{D}_{ij}$: $\mathbf{D}_{ij}$ beeinflußt $\mathbf{D}(]s_i, s_{i+n+1}[\times]t_j, t_{j+n+1}[)$, da

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



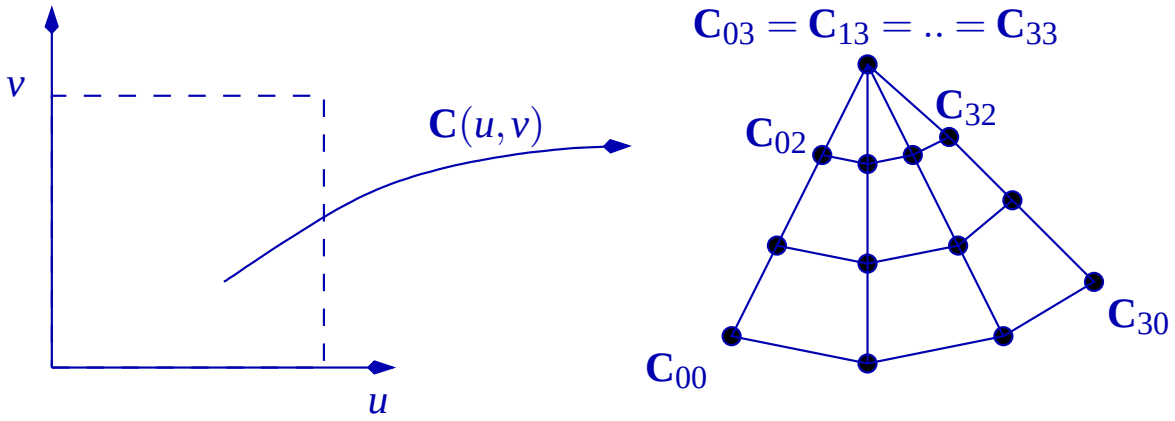


Affine Invarianz und konvexe Hülle: gelten genauso, da $\sum_{(i,j)} N_i^n(u) N_j^n(v) = 1$ auf $[s_n, s_{n+m_u+1}] \times [t_n, t_{n+m_v+1}]$

TP-Bézier-Splineflächen

Probleme: 1. Objekte i.a nicht mit einem Flächenstück (Bézier- oder B-Spline) modellierbar.

2. dreiecksförmige Flächen nur mit *Singularitäten* (zusammenfallen Kontrollpunkte)
   Verschwindende Ableitung $\Rightarrow$ numerisch sehr problematisch!



Ansatz: Verwende mehrere Flächenstücke

Teilaufgaben beim Zusammenfügen von Splineflächen:

1. Kante-Kante Anschluß
2. Konfiguration um eine Ecke

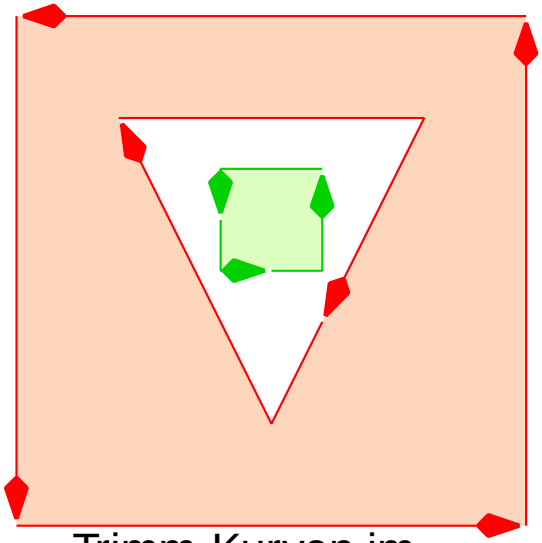
Getrimmte Splineflächen

Aufgabe: Beschreibung speziell geformte Flächen, z.B. mit Löchern

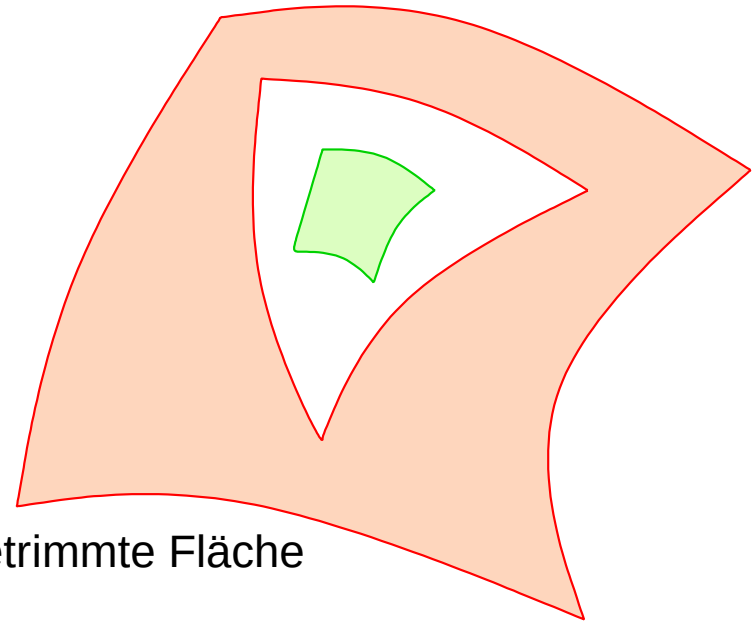
Ansatz: Verwende nicht das gesamte Parametergebiet, z.B. $[0,1]^2$ sondern nur Teilbereiche.

Festlegung der Teilbereiche: Durch geschlossene Kurven im Parametergebiet, wobei

1. pos. orientierte Kurven definieren Innenbereich
2. neg. orientierte Kurven definieren Aussenbereich



Trimm-Kurven im Parametergebiet



getrimmte Fläche

Bislang: Zusammengesetzte Kurven über C^k-Stetigkeit erklärt

Mehrdeutigkeit: Geometrische Form (Kurve) durch verschiedene Funktionen darstellbar:

Für $\mathbf{C}(u), u \in [a,b]$ und bijektives $g : [c,d] \rightarrow [a,b]$ erzeugt

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

dieselbe Geometrie, aber Ableitungen stimmen nicht überein:

Kettenregel: $\mathbf{D}'(u) = \mathbf{C}'(g(u)) \cdot g'(u) \neq \mathbf{C}'(g(u))$ falls $g'(u) \neq 1$

Geometrische Stetigkeit: Kurven $\mathbf{C}$ und $\mathbf{D}$ sind in $\mathbf{D}(u_0)$ G^k-stetig, falls:

$$\exists g \in C^k \text{ bijektiv, } : \mathbf{D}^{(i)}(u_0) = \frac{d^i}{du^i}\mathbf{C}(g(u_0)),\ i = 0,\ldots,k$$

Konkret:

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

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

Definition: Geometrische Größen einer Kurve

Aufgabe: Definition eines lokales Koordinatensystems zu einer Raumkurve (Kamerafahrt).

Gegeben: *Raumkurve* $\mathbf{C} : u \mapsto \mathbb{R}^3$, C^2-stetig mit $\mathbf{C}'(u), \mathbf{C}''(u) \neq \vec{\mathbf{0}}$ und $\mathbf{C}'(u) \not\parallel \mathbf{C}''(u),\ \forall u$

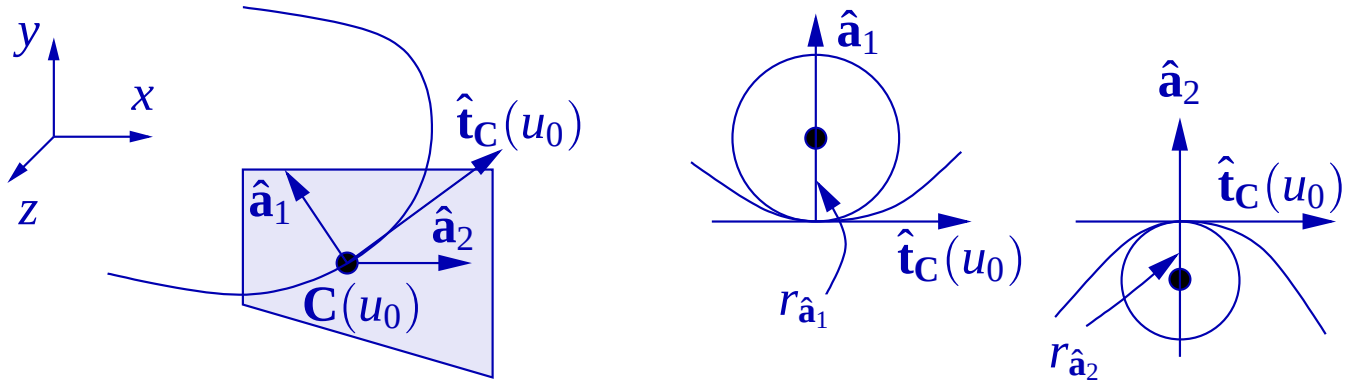
Geometrische Größen von $\mathbf{C}$ in u_0:

Tangente $\hat{\mathbf{t}}_{\mathbf{C}}(u_0)$: Diese weist in Bewegungsrichtung: $\hat{\mathbf{t}}_{\mathbf{C}}(u_0) = \mathbf{C}'(u_0) / \left\|\mathbf{C}'(u_0)\right\|$

Krümmung $\kappa_{\mathbf{C}}(u_0)$: Krümmung wird zunächst in einer Ebene

$$\text{span}\{\hat{\mathbf{t}}_{\mathbf{C}}(u_0), \hat{\mathbf{a}}\} \text{ mit beliebigem } \hat{\mathbf{a}} \perp \hat{\mathbf{t}}_{\mathbf{C}}(u_0)$$

definiert als: $\kappa_{\mathbf{C},\hat{\mathbf{a}}}(u_0) = \dfrac{1}{r_{\hat{\mathbf{a}}}}$, $r_{\hat{\mathbf{a}}}$ Radius des bestangepaßten Kreises



Geometrische Größen einer Kurve (Fortsetzung)

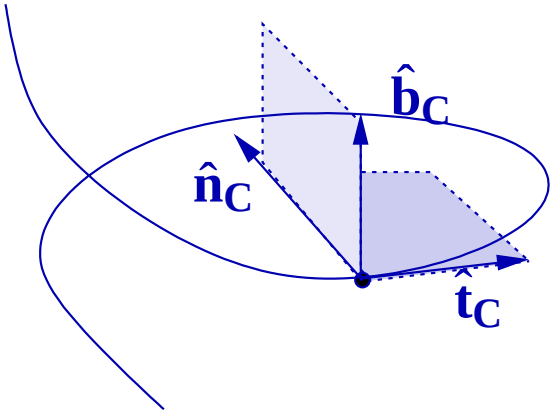
Geometrische Größen von $\mathbf{C}$ in u_0:

Normale $\hat{\mathbf{n}}_{\mathbf{C}}(u_0)$: Entspricht $\hat{\mathbf{a}}$ mit max. Krümmung $\kappa_{\mathbf{C},\hat{\mathbf{a}}}(u_0)$

Es zeigt sich, dass $\hat{\mathbf{n}}_{\mathbf{C}}(u_0) \in \text{span}\{\mathbf{C}'(u_0), \mathbf{C}''(u_0)\}$.

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

Binormale $\hat{\mathbf{b}}_{\mathbf{C}}(u_0)$: Dritter Vektor senkrecht zur Tangente und Normale:



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

Frenet'sche Dreibein: Das Koordinatensystem $\{\hat{\mathbf{t}}_{\mathbf{C}}(u_0), \hat{\mathbf{n}}_{\mathbf{C}}(u_0), \hat{\mathbf{b}}_{\mathbf{C}}(u_0)\}$ im Punkt $\mathbf{C}(u_0)$.

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

Schmiegeebene: $\text{span}\{\hat{\mathbf{t}}_{\mathbf{C}}(u_0), \hat{\mathbf{n}}_{\mathbf{C}}(u_0)\}$, lokal bestandgepaßte Ebene

Normalenebene: Ebene senkrecht zum Kurvenverlauf, also $\text{span}\{\hat{\mathbf{n}}_{\mathbf{C}}(u_0), \hat{\mathbf{b}}_{\mathbf{C}}(u_0)\}$

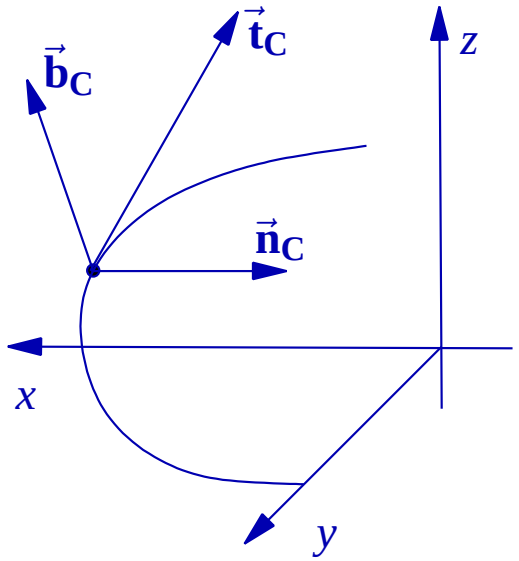
Spirale: $\mathbf{C}(u) = \begin{pmatrix} \sin u \\ \cos u \\ u \end{pmatrix}$; Ableitungen: $\mathbf{C}'(u) = \begin{pmatrix} \cos u \\ -\sin u \\ 1 \end{pmatrix}$; $\mathbf{C}''(u) = \begin{pmatrix} -\sin u \\ -\cos u \\ 0 \end{pmatrix}$

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

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

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

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



3.7 Interpolation mit B-Splines

Gegeben: Sequenz von Parametern $\{u_1, \ldots, u_k\}$ und zugehörige Punkte $\mathbf{P}_i \in \mathbb{R}^3$, $i = 1, \ldots, k$.

Gesucht: *Interpolationskurve*: $\mathbf{C}(u)$ mit $\mathbf{C}(u_i) = \mathbf{P}_i$, $i = 1, \ldots, k$.

Lösungsansätze: Folgende Kurven-Schemata sind denkbar

1. Catmull-Rom Splines (s.o.)
2. B-Splines

<u>Ansatz:</u> Interpolation an den Knoten mit kubischen uniformen B-Splines

Kurve: Kubische uniforme B-Spline-Kurve $\mathbf{D}$ mit (also $t_i = i$):

$$\mathbf{D}(u) = \sum_{i=0}^{m} \mathbf{D}_i N_i^3(u), \quad u \in [3, m+1]$$

Anzahl Segmente: In $[3, m+1]$ gibt es $m-1$ Knoten i, d.h.

$$u_1 = 3, u_2 = 4, \ldots, u_k = k+2 \stackrel{!}{=} m+1 \text{ also } u_i = i+2$$

$\Longrightarrow m = k+1$, aber $m+1 = k+2$ Kontrollpunkte (unterbestimmtes System)

Frage: Wie müssen die $\mathbf{D}_i$ gewählt werden, so dass $\mathbf{D}(u_i) = \mathbf{D}(i+2) = \mathbf{P}_i$, $i = 1, \ldots, k$?

Ansatz: Interpolation mit uniformen B-Splines (Forts.)

Erinnerung: Uniforme, kubische B-Splines $\mathbf{D}(u)$ sind stückweise C^2-stetige Bézier-Kurven

Umrechnung: $\mathbf{D}(u) \rightarrow \mathbf{C}^j(u)$ für $u \in [j, j+1]$:

$$
\begin{aligned}
\mathbf{C}_1^j &= \tfrac{2}{3}\mathbf{D}_{j-2} + \tfrac{1}{3}\mathbf{D}_{j-1} \\
\mathbf{C}_2^j &= \tfrac{1}{3}\mathbf{D}_{j-2} + \tfrac{2}{3}\mathbf{D}_{j-1} \\
\mathbf{C}_0^j &= \tfrac{1}{2}\mathbf{C}_2^{j-1} + \tfrac{1}{2}\mathbf{C}_1^j \\
&= \tfrac{1}{6}\mathbf{D}_{j-3} + \tfrac{2}{3}\mathbf{D}_{j-2} + \tfrac{1}{6}\mathbf{D}_{j-1} \\
\mathbf{C}_3^j &= \tfrac{1}{2}\mathbf{C}_2^j + \tfrac{1}{2}\mathbf{C}_1^{j+1} \\
&= \tfrac{1}{6}\mathbf{D}_{j-2} + \tfrac{2}{3}\mathbf{D}_{j-1} + \tfrac{1}{6}\mathbf{D}_j
\end{aligned}
$$

$$
\underbrace{\begin{pmatrix}
\frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 \\
0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdots & 0 \\
0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 \\
\vdots & \vdots & & \ddots & \ddots & \ddots & & \vdots \\
0 & 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{pmatrix}}_{\in \mathbb{R}^{k,k+2}}
\begin{pmatrix} \mathbf{D}_0 \\ \vdots \\ \mathbf{D}_{k+1} \end{pmatrix}
=
\begin{pmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_k \end{pmatrix}
$$

Bedingung: $\boxed{\mathbf{P}_i = \mathbf{D}(u_i) = \mathbf{D}(i+2) = \mathbf{C}_0^{i+2} = \tfrac{1}{6}\mathbf{D}_{i-1} + \tfrac{2}{3}\mathbf{D}_i + \tfrac{1}{6}\mathbf{D}_{i+1},\ i = 1, \ldots, k}$

Gesucht: Kubische, uniforme B-Spline-Kurve $\mathbf{D}$ mit:

$$\mathbf{D}(3) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \mathbf{D}(4) = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \mathbf{D}(5) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{D}(6) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Lösung: Gleichungssystem mit natürlichen Randbedingungen:

$$\begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 \\ 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\ 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 0 & 0 & 0 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{D}_0 \\ \mathbf{D}_1 \\ \mathbf{D}_2 \\ \mathbf{D}_3 \\ \mathbf{D}_4 \\ \mathbf{D}_5 \end{pmatrix} = \begin{pmatrix} (0,0) \\ (-1,0) \\ (-1,1) \\ (1,1) \\ (1,0) \\ (0,0) \end{pmatrix}$$

$$\Rightarrow \mathbf{D}_0 = \begin{pmatrix} -1/3 \\ -6/5 \end{pmatrix}, \mathbf{D}_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \mathbf{D}_2 = \begin{pmatrix} -5/3 \\ 6/5 \end{pmatrix},$$

$$\mathbf{D}_3 = \begin{pmatrix} 5/3 \\ 6/5 \end{pmatrix}, \mathbf{D}_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{D}_5 = \begin{pmatrix} 1/3 \\ -6/5 \end{pmatrix}$$