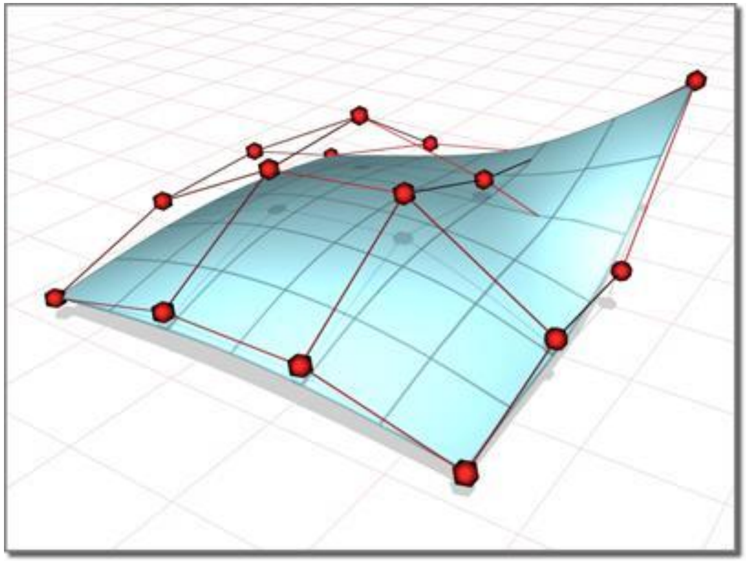
Modellierung mit Kurven



Inhalt

Computergraphik und Visualisierung

- Polynombasen
- Parametrische Kurven
- Beziér Kurven
- Interpolierende Kurven
- Hermite Splines
- Stetiges Aneinanderfügen
- Basis-Splines
- Übersicht

Polynombasen
Monombasis

Computergraphik und Visualisierung

- Wir betrachten hier nur Polynome in einer Variablen t. Für Flächen werden Polynome in zwei Variablen benötigt.
- Die einfachste Darstellung von Polynomen ist in der Monombasis $\{ 1, t, t^2, \ldots, t^g \}$
- Das Monom mit der höchsten Potenz dessen Koeffizient ungleich null ist definiert den Grad g des Polynoms
- Die Koeffizienten an die Basisfunktionen können als Vektor interpretiert werden und bilden den $(g+1)$-dimensionalen Vektorraum der Polynome

$$f(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_g t^g$$

$$= \sum_{i=0}^{g} a_i t^i = \langle \vec{\boldsymbol{a}}, \vec{\boldsymbol{M}}^g(t) \rangle$$

$$= \vec{\boldsymbol{a}}^T \vec{\boldsymbol{M}}^g(t) = \vec{\boldsymbol{M}}^g(t)^T \vec{\boldsymbol{a}}$$

Monombasis:

$$M_i^g(t) = t^i \quad \text{oder} \quad \vec{\boldsymbol{M}}^g(t) = \begin{pmatrix} 1 \\ t \\ \vdots \\ t^g \end{pmatrix}$$

Koeffizientenvektor $\vec{\boldsymbol{a}} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_g \end{pmatrix} \in \boldsymbol{R}^{g+1}$

Polynombasen
Bernsteinbasis

Computergraphik und Visualisierung

- Die Fakultät einer positiven ganzen Zahl n ergibt sich aus dem Produkt aller ganzen Zahlen kleiner gleich n
- Binomialkoeffizienten sind mit Hilfe der Fakultät definiert
- Optional können die Binomialkoeffizienten auch über das Pascal´sche Dreieck berechnet werden
- Bernsteinbasis:

$$B_i^g(t) = \binom{g}{i}(1-t)^{g-i}t^i$$

$$\vec{\mathbf{B}}^2(t) = \left((1-t)^2 \quad 2(1-t)t \quad t^2\right)^T$$

$$\vec{\mathbf{B}}^3(t) = \left((1-t)^3 \quad 3(1-t)^2 t \quad 3(1-t)t^2 \quad t^3\right)^T$$

Fakultät: $n! = n\cdot(n-1)\cdot\ldots\cdot 2\cdot 1$

Binomialkoeffizient: $\binom{n}{i} = \dfrac{n!}{i!(n-i)!}$

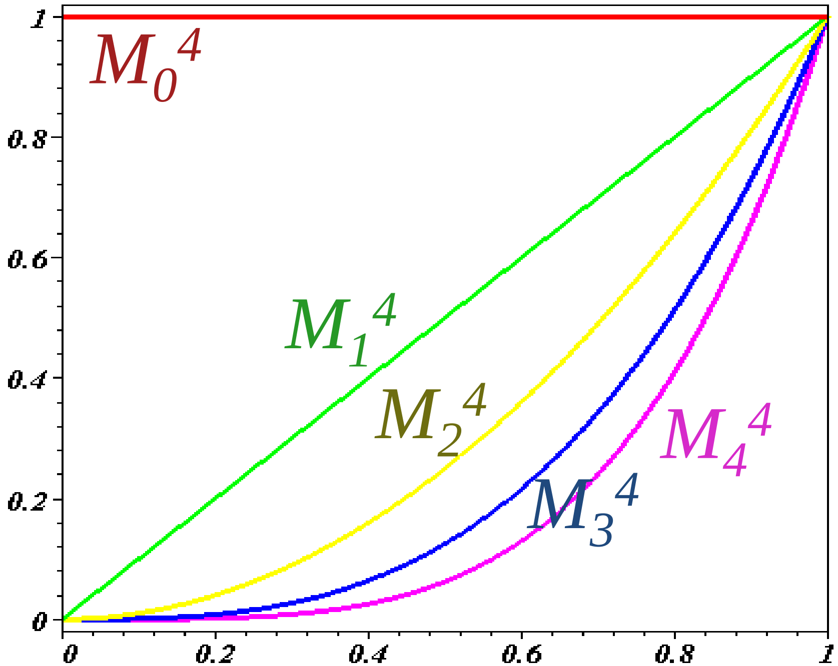
Pascal'sches Dreieck:

1	$n=0$
1 1	$n=1$
1 2 1	$n=2$
1 3 3 1	$n=3$
1 4 6 4 1	$n=4$

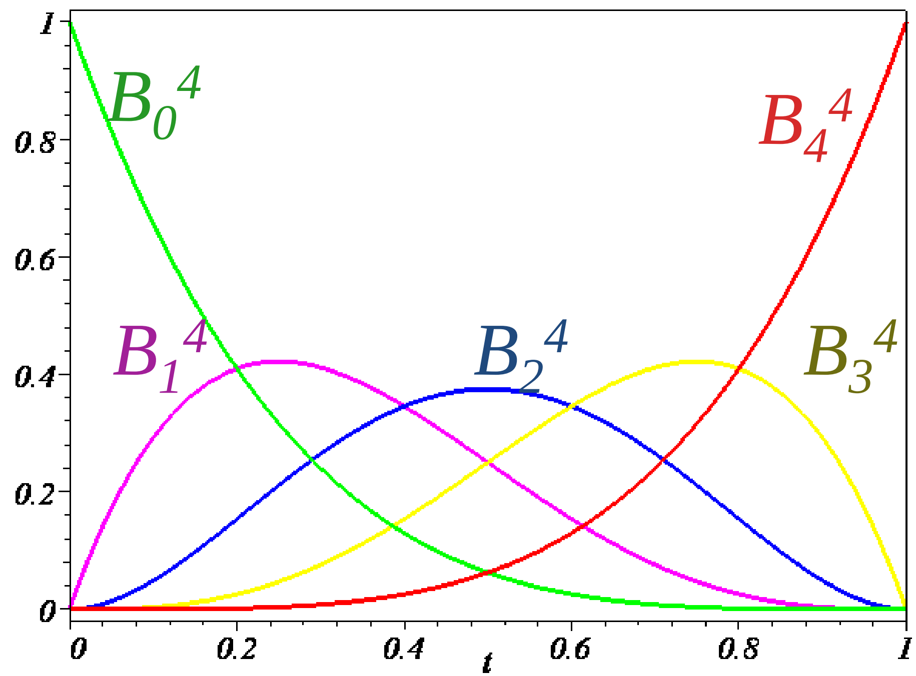
$i=0$ $i=1$ $i=2$ $i=3$

Polynombasen Vergleich

Monombasis



Bernsteinbasis



Wichtig ist, dass nur der Definitionsbereich $t \in [0,1]$ betrachtet wird

Polynombasen
Basistransformation

Computergraphik und Visualisierung

- Dasselbe Polynome kann in verschiedenen Basen dargestellt werden. Dadurch ändert sich der Koeffizientenvektor
- Die Umrechnung des Koeffizientenvektors in eine andere Basis heißt Basistransformation, diese kann mit einer $(g+1)\times(g+1)$-dimensionalen Matrix dargestellt werden
- Die Basistransformation erfolgt durch Matrix-Vektor-Multiplikation
- Die inverse Matrix transformiert den Koeffizientenvektor in umgekehrter Richtung

$$f(t) = \vec{a}^T \vec{M}^g(t) \overset{1}{=} \vec{b}^T \vec{B}^g(t)$$

Basistransformation:

$$\vec{b} = T_{B\leftarrow M}\vec{a}, \quad T_{B\leftarrow M} \in R^{(g+1)\times(g+1)}$$

Inverse Basistransformation:

$$\vec{a} = T_{M\leftarrow B}\vec{b} = \left(T_{B\leftarrow M}\right)^{-1}\vec{b} \qquad 2$$

Zusammenhang zw. Basistransformation und Transformation der Basen:

$$f(t) = \vec{a}^T \vec{M}^g(t) \overset{2}{=} \vec{b}^T \left(\left(T_{B\leftarrow M}\right)^{-1}\right)^T \vec{M}^g(t)$$

$$\overset{1}{\Rightarrow} \vec{B}^g(t) = A_{B\leftarrow M}\vec{M}^g(t)$$

$$A_{B\leftarrow M} = \left(\left(T_{B\leftarrow M}\right)^{-1}\right)^T$$

Polynombasen Beispieltransformation

- Auch die Basen stehen in einem linearen Zusammenhang zueinander, der durch eine Matrix $\boldsymbol{A}_{B\leftarrow M}$ darstellbar ist
- Diese Matrix kann bei Transformation von der Monombasis durch ausmultiplizieren der anderen Basis abgelesen werden
- Rechts ist dies für die Bernsteinbasis durchgeführt
- Die Transformationsmatrix für den Koeffizientenvektor ergibt sich durch Transponieren und Invertieren.

$$\vec{\boldsymbol{B}}^3(t) = \begin{pmatrix} (1-t)^3 \\ 3(1-t)^2 t \\ 3(1-t)t^2 \\ t^3 \end{pmatrix} = \begin{pmatrix} 1-3t+3t^2-t^3 \\ 3t-6t^2+3t^3 \\ 3t^2-3t^3 \\ t^3 \end{pmatrix}$$

$$= \underbrace{\begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{A}_{B\leftarrow M}} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

$$\Rightarrow \boldsymbol{T}_{B\leftarrow M} = \left(\left(\boldsymbol{A}_{B\leftarrow M} \right)^T \right)^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{3} & 0 & 0 \\ 1 & \frac{2}{3} & \frac{1}{3} & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Polynombasen
Effiziente Auswertung

- Für die Auswertung von Polynomen gibt es je nach Basis mehrere Möglichkeiten deren Laufzeit sich in Bezug auf den Grad unterscheidet.
- Für die Monombasis benötigt eine direkte Umsetzung der Formel im Grad quadratisch viele Rechenoperationen
- Mit dem Hornerschema kann, die Anzahl der Rechenoperationen reduziert werden, so dass nur noch proportional zum Grad viele Operationen notwendig sind ist.

$$f(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_g t^g$$

Pseudo-Code Standardauswertung:

```
f=a0
for i=1..g do
    Mi = t
    for j=2..i do
        Mi *= t
    f += ai*Mi
```

Hornerschema

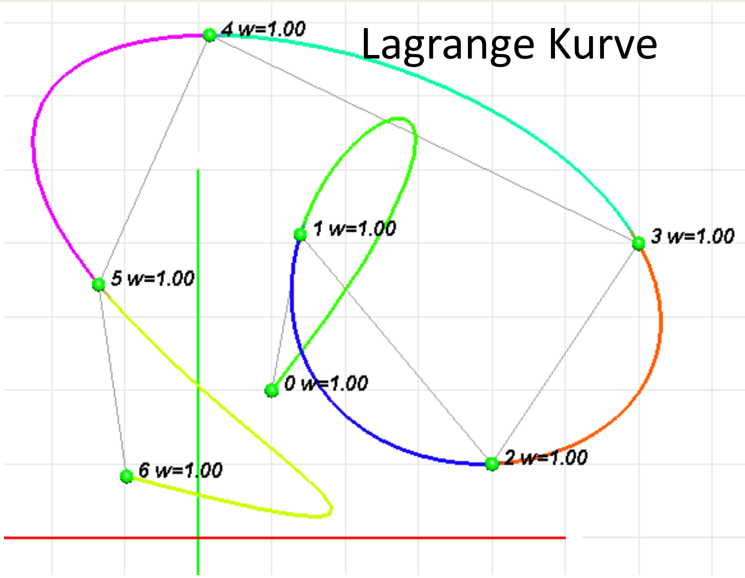
$$f(t) = a_0 + t\left(a_1 + t\left(\ldots + t\left(a_{g-1} + t a_g\right)\ldots\right)\right)$$

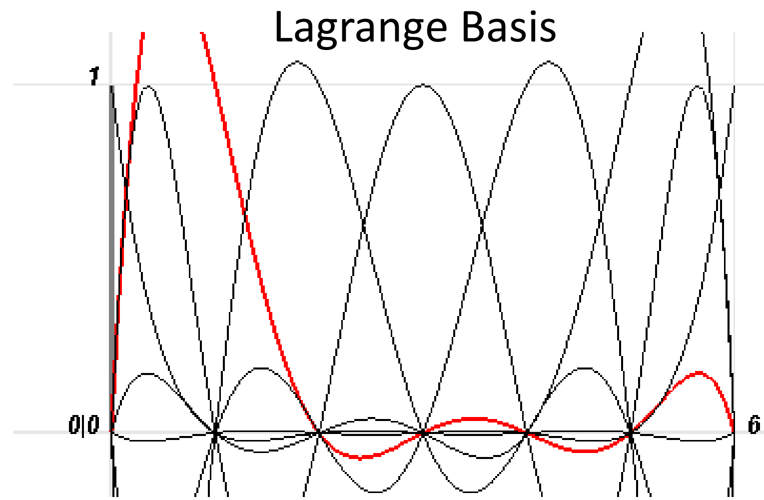
Pseudo-Code Hornerschema

```
f=ag
for i=g-1..0 do
    f = ai+t*f
```

Parametrische Kurven
Modellierung von Kurven

Computergraphik und Visualisierung





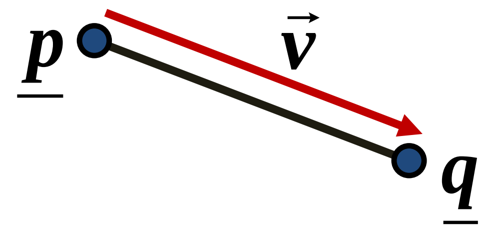
Kontrollpunktparadigma

- Kontrollpolygon aus Kontroll-punkten definiert den Kurvenverlauf
- Gewichte geben bei rationalen Basen zusätzliche Freiheits-grade zum modellieren

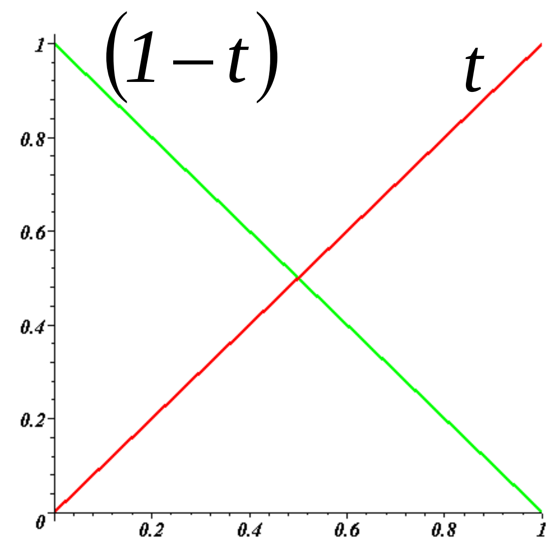
Basisfunktionen

- Der Einfluss der verschiedenen Kontrollpunkte wird durch polynomiale Basisfunktionen gesteuert
- Je nach Wahl der Basis hat die resultierende Kurve andere Eigenschaften

- Die einfachste parametrische Kurve ist durch ein geradliniges Segment zwischen zwei Punkten definiert
- Dazu werden nur lineare Basisfunktionen benötigt
  - Monombasis: $\{\, 1, t \,\}$ wird an Anfangspunkt und Differenzvektor multipliziert
  - Bernsteinbasis: $\{\, 1-t, t \,\}$ wird direkt an die Kontroll-punkte multipliziert
    - Bei t=0 und t=1 ist eine Basis 1 und die andere 0 ➔ End-punkte werden interpoliert
    - Die Basis summiert sich überall zu $1-t+t=1$

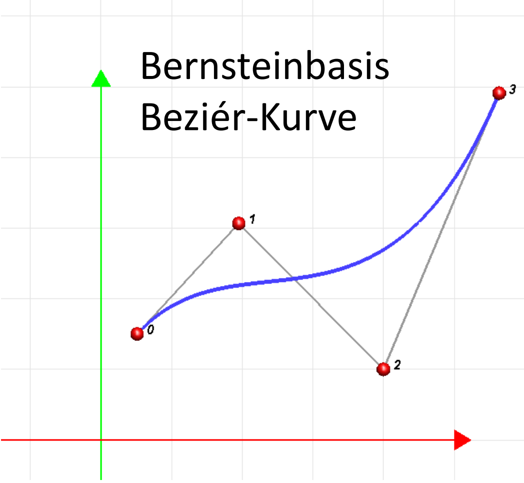


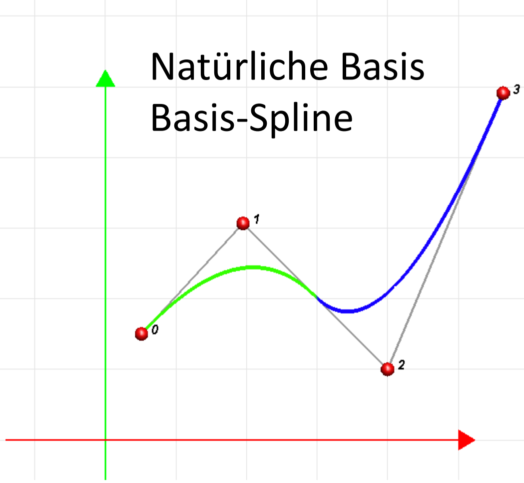
$$\underline{c}(t) = \underline{p} + \vec{v}t$$
$$= (1-t)\underline{p} + t\underline{q}$$

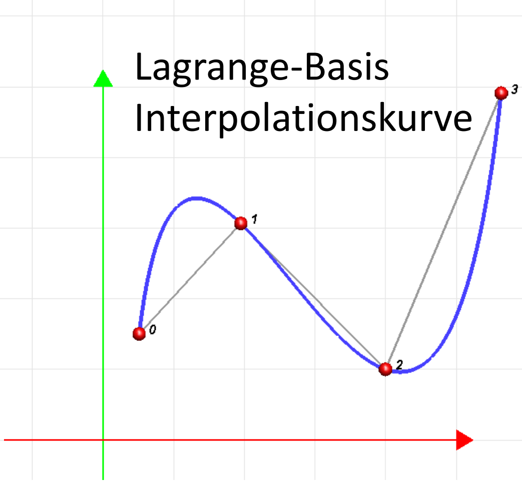


Parametrische Kurven
Eigenschaften die aus Basis folgen

Computergraphik und Visualisierung







- Je nach den gewählten Basisfunktionen ergibt sich mit denselben Kontrollpunkten unterschiedliche Kurven mit anderen Eigenschaften
- Splines sind aus mehreren Kurvensegmenten so zusammengesetzt, dass ein glatter Übergang entsteht
- gewünschte Eigenschaften
  - Glattheit
  - Kontrollpunktinterpolation
  - Endtangenteninterpolation … Tangenten der Kurvenendpunkte entsprichen Endsegmenten des Kontrollpolygons
  - Konvexe Hüllen Eigenschaft: Kurvenverlauf nur innerhalb der konvexen Hülle der Kontrollpunkte

Parametrische Kurven Matrixdarstellung

Computergraphik und Visualisierung

- Im Vergleich zu Polynomen werden die skalaren Koeffizienten zu Kontrollpunkten und die Koeffizientenvektoren zu Kontrollpunktmatrizen
- Bei der Basistransformation bleibt alles gleich, außer dass die Kontrollpunktmatrizen durch Matrix-Matrix-Multiplikation transformiert werden
- Im 3D hat die Kontrollpunktmatrix einfach eine weitere Zeile

Beispiel Beziér-Kurve, die mit Bernsteinbasis definiert wird und deren Kontrollpunkte auch Beziér-Punkte genannt werden

$$\underline{\boldsymbol{c}}(t) = \sum_{i=0}^{g} \underline{\boldsymbol{b}}_i B_i^g(t)$$

$$= \boldsymbol{K}_{\underline{b}} \cdot \boldsymbol{B}^g(t)$$

Kontrollpunktmatrix

Bsp.: $g=3$

$$\underline{\boldsymbol{c}}(t) = \begin{pmatrix} b_{0,x} & b_{1,x} & b_{2,x} & b_{3,x} \\ b_{0,y} & b_{1,y} & b_{2,y} & b_{3,y} \end{pmatrix} \begin{pmatrix} (1-t)^3 \\ 3(1-t)^2 t \\ 3(1-t)t^2 \\ t^3 \end{pmatrix}$$

Beziér Kurven
einleitendes Beispiel

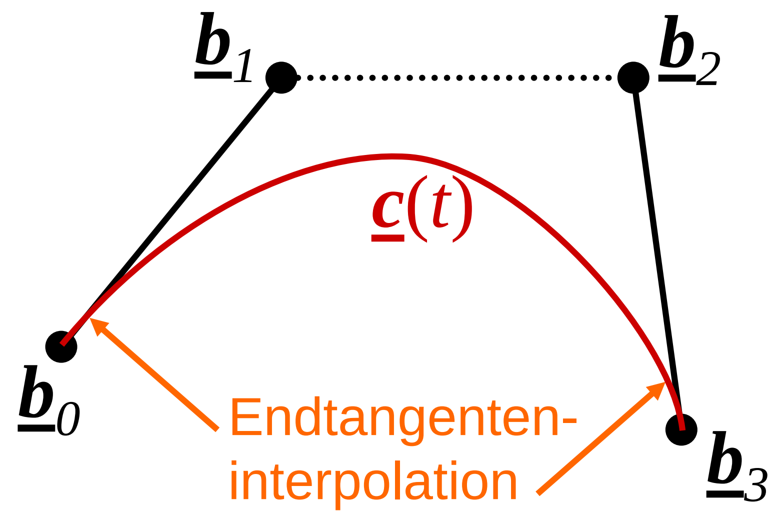
Computergraphik und Visualisierung

- In den meisten Tools werden Bezier-Kurven dritten Grades für die Definition von Kurven verwendet
- Die Kurve ist durch vier Bezier-Punkte $\underline{\boldsymbol{b}}_0, \ldots, \underline{\boldsymbol{b}}_3$ definiert
- $\underline{\boldsymbol{b}}_0, \underline{\boldsymbol{b}}_3$ definieren Start- und Endpunkt
- $\underline{\boldsymbol{b}}_1, \underline{\boldsymbol{b}}_2$ definieren die Tangenten in Start- und Endpunkt

Bezier-Kurve 3.Grades:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = (1-t)^3 \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + 3t(1-t)^2 \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + 3t^2(1-t) \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + t^3 \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}$$

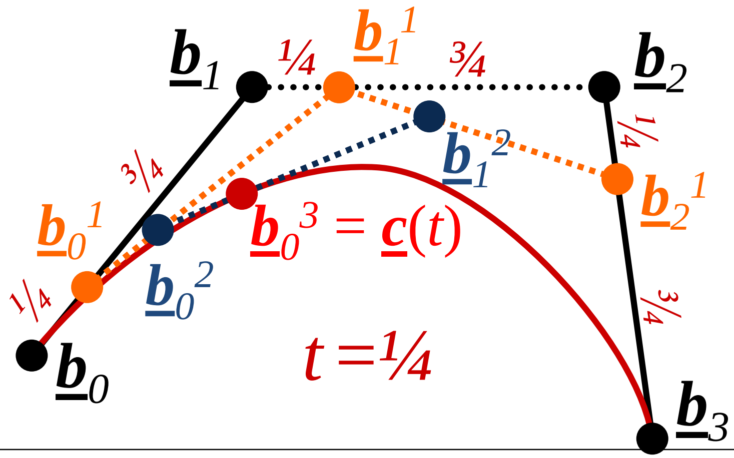
mit $t \in [0,1]$



Kurvenauswertung nach De Casteljau

- Die Produkte von 4-5 Zahlen beim Auswerten von Bezier-Kurven 3. Grades sind numerisch nicht stabil.
- Deshalb verwendet man das Schema nach de Casteljau:

$$\underline{\mathbf{c}}(t) = \underline{\mathbf{b}}_0^g \qquad \underline{\mathbf{b}}_i^0 = \underline{\mathbf{b}}_i$$
$$\underline{\mathbf{b}}_i^r = (1-t)\underline{\mathbf{b}}_i^{r-1} + t\underline{\mathbf{b}}_{i+1}^{r-1}$$



Beweis für de Casteljau:

$$\mathbf{c}(t) = (1-t)^3\underline{\mathbf{b}}_0 + 3t(1-t)^2\underline{\mathbf{b}}_1 +$$
$$3t^2(1-t)\underline{\mathbf{b}}_2 + t^3\underline{\mathbf{b}}_3$$

$$= (1-t)^2\left((1-t)\underline{\mathbf{b}}_0 + t\underline{\mathbf{b}}_1\right) +$$
$$2t(1-t)\left((1-t)\underline{\mathbf{b}}_1 + t\underline{\mathbf{b}}_2\right) +$$
$$t^2\left((1-t)\underline{\mathbf{b}}_2 + t\underline{\mathbf{b}}_3\right)$$

$$= (1-t)^2\underline{\mathbf{b}}_0^1 + 2t(1-t)\underline{\mathbf{b}}_1^1 + t^2\underline{\mathbf{b}}_2^1$$

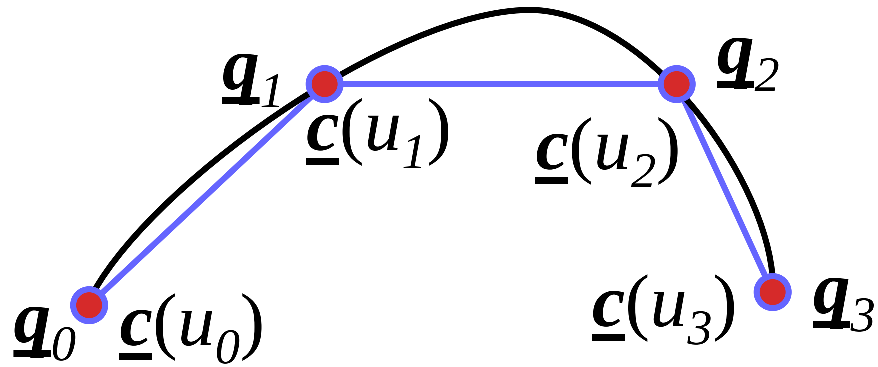
$$= (1-t)\left((1-t)\underline{\mathbf{b}}_0^1 + t\underline{\mathbf{b}}_1^1\right) +$$
$$t\left((1-t)\underline{\mathbf{b}}_1^1 + t\underline{\mathbf{b}}_2^1\right)$$

$$= (1-t)\underline{\mathbf{b}}_0^2 + t\underline{\mathbf{b}}_1^2$$
$$= \underline{\mathbf{b}}_0^3 = \underline{\mathbf{c}}(t)$$

Interpolierende Kurven
Lagrange Interpolation

Computergraphik und Visualisierung

- Wenn die Kurve durch alle Kontrollpunkte gehen soll, (Interpolation) so muss man angeben für welche Parameterwerte $t=u_i$ dies geschehen soll.
- Die u_i nennt man Stützstellen und den Vektor U der aus einer Stützstelle pro Kontrollpunkt definiert ist Stützstellenvektor
- Aus dem Stützstellenvektor definiert man die Lagrange Basis $L_i^g(t)$ so, dass $L_i^g(u_i)=1$ ist und $L_i^g(u_{j\neq i})=0$. Daraus folgt die Interpolationseigenschaft.



Interpolationsproblem:

- geg: Kontrollpunkt $\underline{q}_{i=0..g}$ und Stützstellen $u_{i=0..g}$
- ges.: Kurve $\underline{c}(t)$ mit $\underline{c}(u_i)=\underline{q}_i$
- Lösung: Kurve in Lagrange-Basis: $$\underline{c}(t) = \sum_{i=0}^{g} \underline{q}_i L_i^g(t)$$

$$L_i^g(t) = \prod_{0=k\neq i}^{g} \frac{t-u_k}{u_i-u_k} = \frac{(t-u_0)\cdots\cancel{(t-u_i)}\cdots(t-u_g)}{(u_i-u_0)\cdots\cancel{(u_i-u_i)}\cdots(u_i-u_g)}$$

Aufstellen der Lagrange Basis

Computergraphik und Visualisierung

$$L_i^g(t) = \prod_{0=k\neq i}^{g} \frac{t-u_k}{u_i-u_k}$$

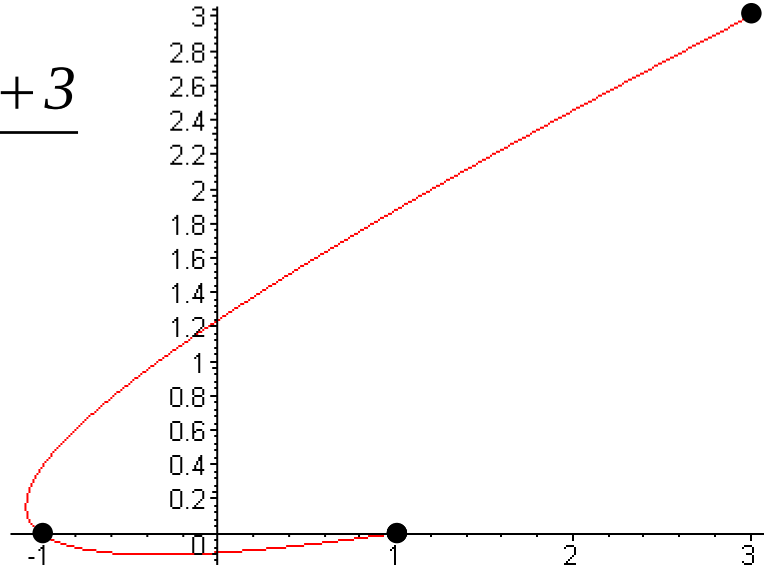
Beispiel:

- Gegeben: $\{u_i\}_i = \{0,1,3\}$
  $\{\underline{\mathbf{q}}_i\}_i = \{(1,0),(-1,0),(3,3)\}$

$$L_0^2(t) = \frac{(t-1)(t-3)}{(0-1)(0-3)} = \frac{t^2-4t+3}{3}$$

$$L_1^2(t) = \frac{t(t-3)}{-2}$$

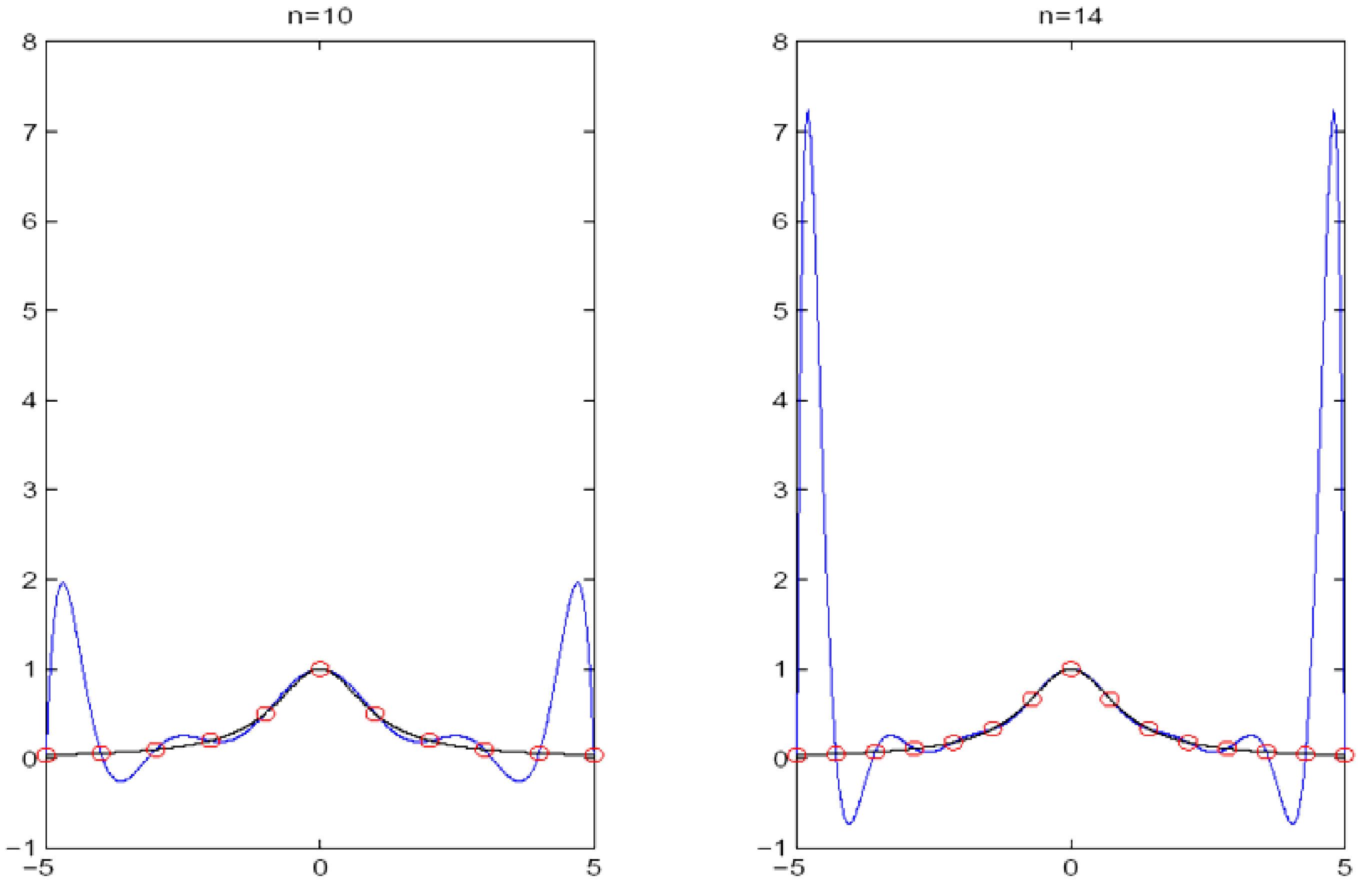
$$L_2^2(t) = \frac{t(t-1)}{6}$$



$$\mathbf{c}(t) = \begin{pmatrix}1\\0\end{pmatrix}\frac{t^2-4t+3}{3} + \begin{pmatrix}-1\\0\end{pmatrix}\frac{t(t-3)}{-2} + \begin{pmatrix}3\\3\end{pmatrix}\frac{t(t-1)}{6}$$

Interpolierende Kurven
Überschwingungsproblematik

- **Vorsicht**: bei vielen Knoten kommt es bei der Lagrange-Interpolation zu Überschwingen



Interpolierende Kurven Weitere Eigenschaften

Computergraphik und Visualisierung

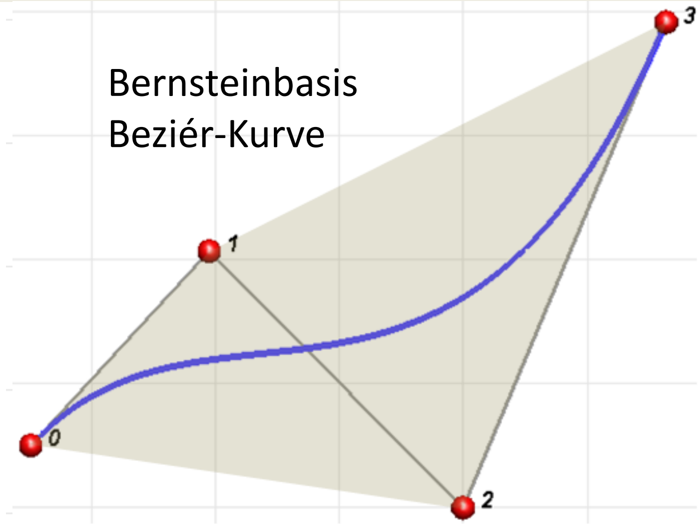
- Affine Invarianz … die Basisfunktionen summieren sich für alle t zu eins:

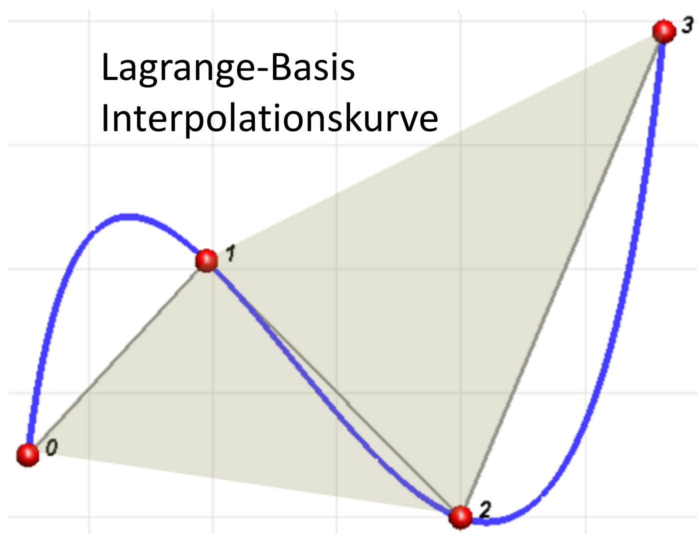
$$\forall t, g, U : 1 = \sum_{i=0}^{g} B_i^g(t) = \sum_{i=0}^{g} L_i^g(t)$$

- Konvexe Hülleneigenschaft … die Kurve liegt in der konv. Hülle der Kontrollpunkte. Immer bei affiner Invarianz zusammen mit positiven Basisfunktionen:

$$\forall t, i, g : B_i^g(t) \geq 0$$

für Lagrange Basis nicht erfüllt.

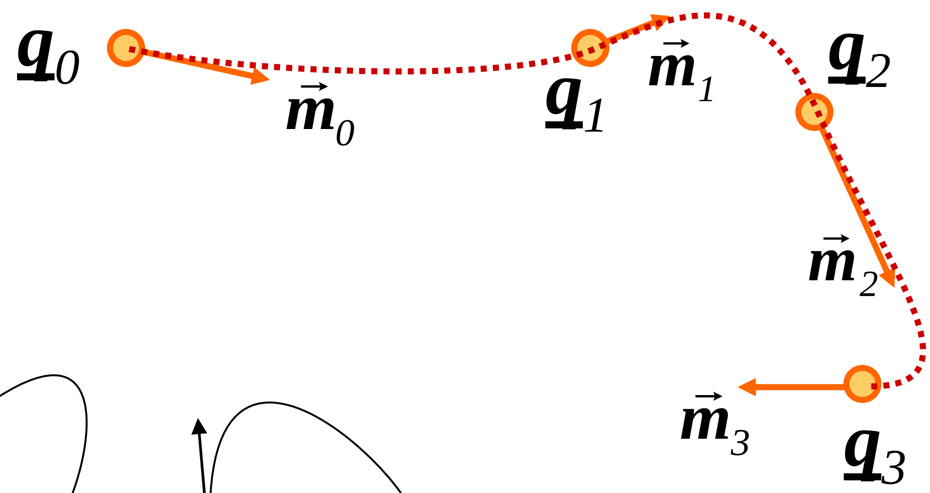




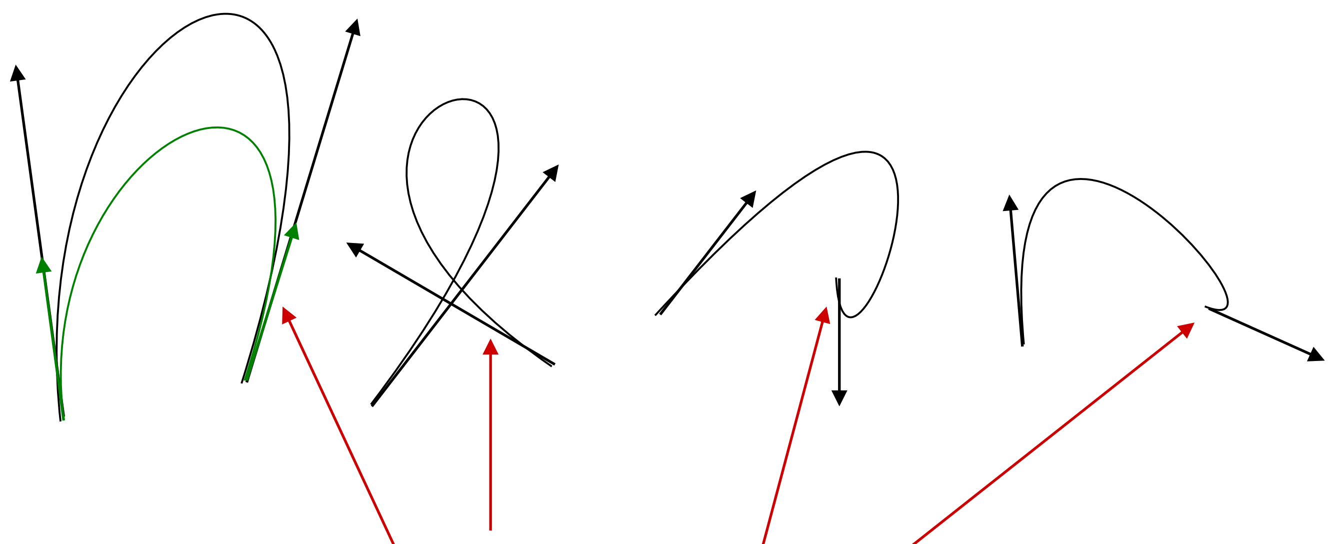
Hermite Interpolation
Motivation

Idee:

- Interpolation von Positions- und Tangenteninformation



Beispiele:



Vorsicht: In Darstellung zeigen die Pfeile am Ende entgegen der Vektoren ***m***, die für die Definition der Hermite-Kurve verwendet wird!!!

Gegeben:

- Parameter $u_0, u_1, \ldots, u_n \in \mathbf{R}$,
- Punkte $\underline{\boldsymbol{q}}_0, \underline{\boldsymbol{q}}_1, \ldots, \underline{\boldsymbol{q}}_n \in \mathbf{R}^d$
- Tangenten $\vec{\boldsymbol{m}}_0, \vec{\boldsymbol{m}}_1, \ldots, \vec{\boldsymbol{m}}_n \in \mathbf{R}^d$

Gesucht:

- n Kurvensegmente $\underline{\boldsymbol{c}}_i(t) : [u_i, u_{i+1}] \rightarrow \mathbf{R}^d$
- mit $\underline{\boldsymbol{c}}_i(u_i) = \underline{\boldsymbol{q}}_i, \quad \dot{\underline{\boldsymbol{c}}}_i(u_i) = \vec{\boldsymbol{m}}_i$

  $\underline{\boldsymbol{c}}_i(u_{i+1}) = \underline{\boldsymbol{q}}_{i+1}, \quad \dot{\underline{\boldsymbol{c}}}_i(u_{i+1}) = \vec{\boldsymbol{m}}_{i+1}$

Lösung der Interpolationsaufgabe

(Kontrolltangenten-interpolation)

$$\underline{\boldsymbol{c}}_i(t) = H_0^3(s_i)\underline{\boldsymbol{q}}_i + \Delta_i \cdot H_1^3(s_i)\vec{\boldsymbol{m}}_i + \\ + \Delta_i \cdot H_2^3(s_i)\vec{\boldsymbol{m}}_{i+1} + H_3^3(s_i)\underline{\boldsymbol{q}}_{i+1} \;, t \in [u_i, u_{i+1}]$$

wobei $s_i = (t - u_i)/\Delta_i$ und $\Delta_i = u_{i+1} - u_i$

in Klausur gegeben

Computergraphik und Visualisierung

Beispiel

- Hermite-Polynome vom Grad 3:

in Klausur gegeben

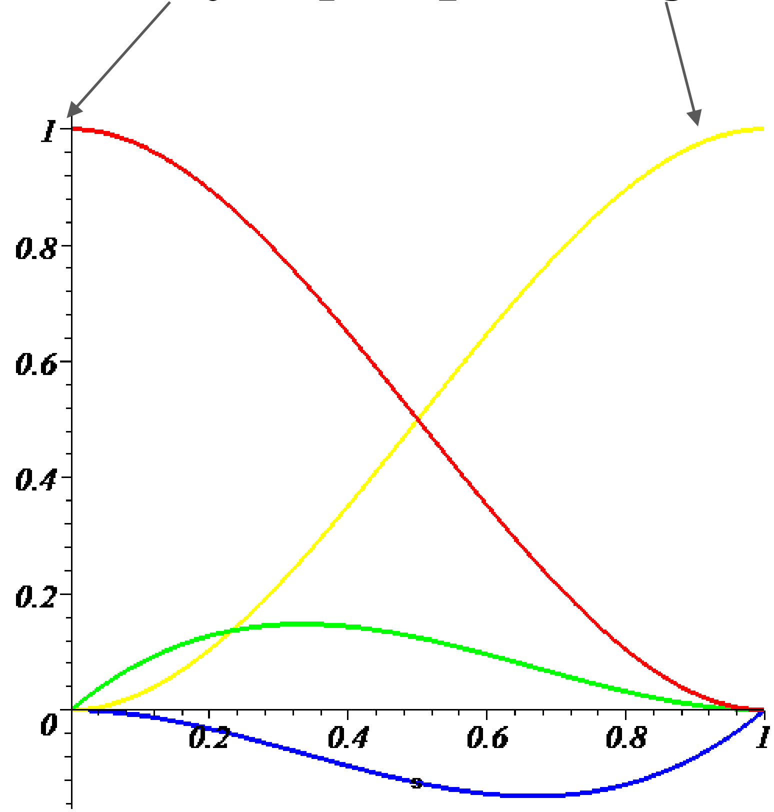
$$H_0^3(t) = (1-t)^2(1+2t)$$

$$H_1^3(t) = t(1-t)^2$$

$$H_2^3(t) = -t^2(1-t)$$

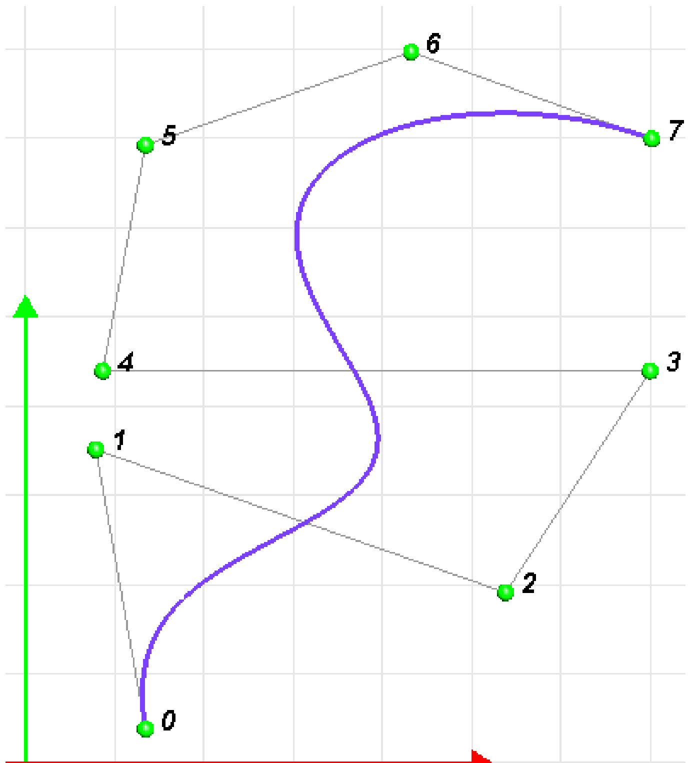
$$H_3^3(t) = (3-2t)t^2$$

H_0^3, H_1^3, H_2^3 und H_3^3

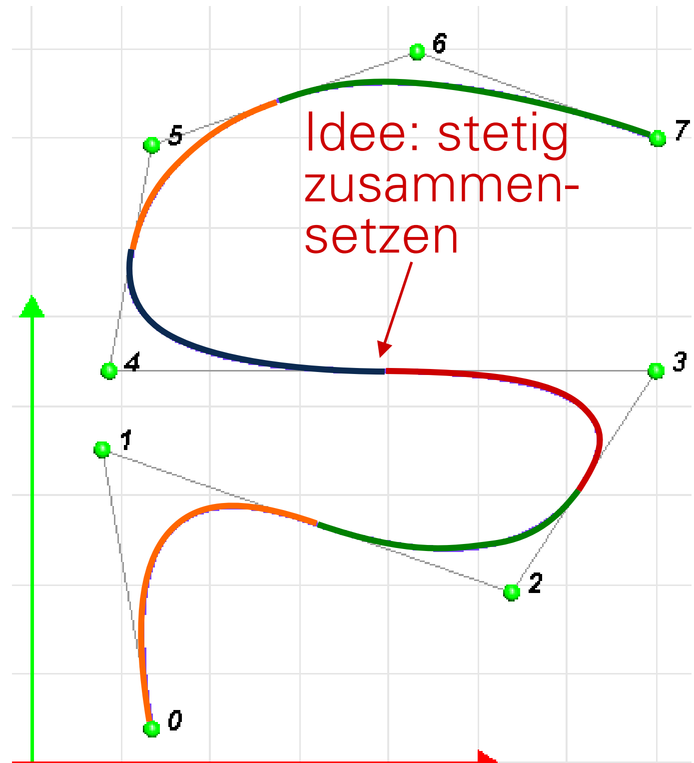


Stetiges Aneinanderfügen
Motivation

- Mit wachsendem Grad folgen Bézier-Kurven immer weniger dem Kontrollpolygon:



Bézier – Grad 7

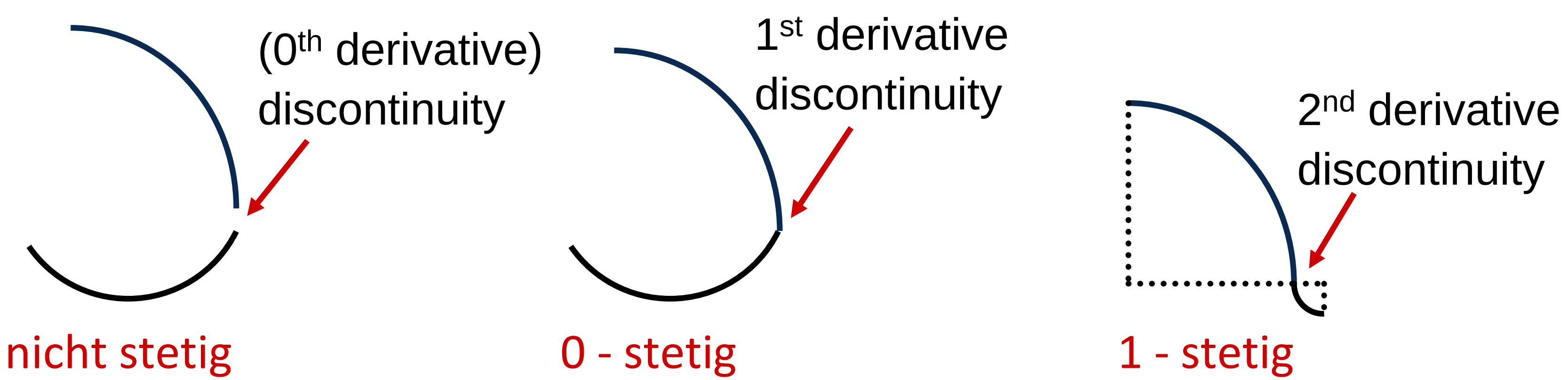


Spline – Grad 2

Stetiges Aneinanderfügen
Verschiedene Stetigkeiten

Computergraphik und Visualisierung

- Der Anschluss zwischen zwei Kurvensegmenten wird nach der Anzahl der übereinstimmenden Ableitungen klassifiziert:



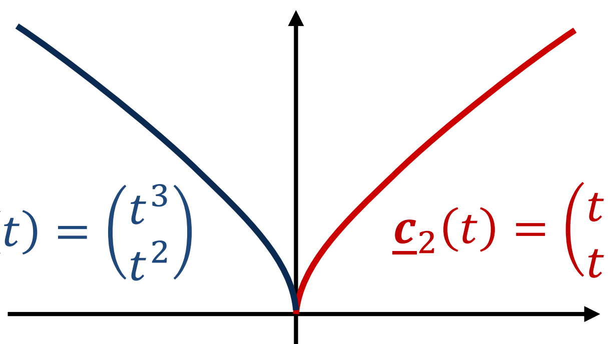
- Man unterscheidet zwischen parametrischer (C) und geometrischer (G) Stetigkeit

Stetiges Aneinanderfügen Parametrisch versus Geometrisch

Computergraphik und Visualisierung

C^k-stetig

- Die ersten k Ableitungen nach dem Parameter stimmen überein



$$\underline{c}_1(t) = \begin{pmatrix} t^3 \\ t^2 \end{pmatrix} \qquad \underline{c}_2(t) = \begin{pmatrix} t^3 \\ t^2 \end{pmatrix}$$

$$\dot{\vec{c}}_1(0) = \dot{\vec{c}}_2(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- Kurve ist C^∞ – aber nicht einmal G^1 – stetig

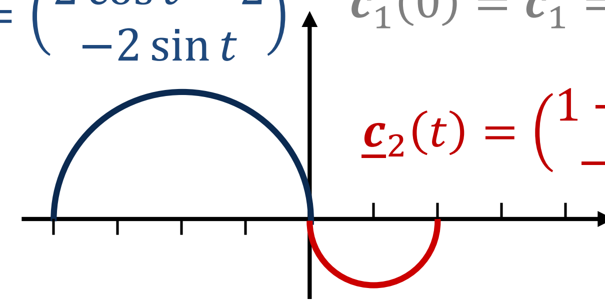
G^k-stetig

- Die ersten k Ableitungen nach der Kurvenlänge stimmen überein

Ableitung nach Bogenlänge

$$\vec{c}_1'(0) = \vec{c}_1' = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

$$\underline{c}_1(t) = \begin{pmatrix} 2\cos t - 2 \\ -2\sin t \end{pmatrix}$$



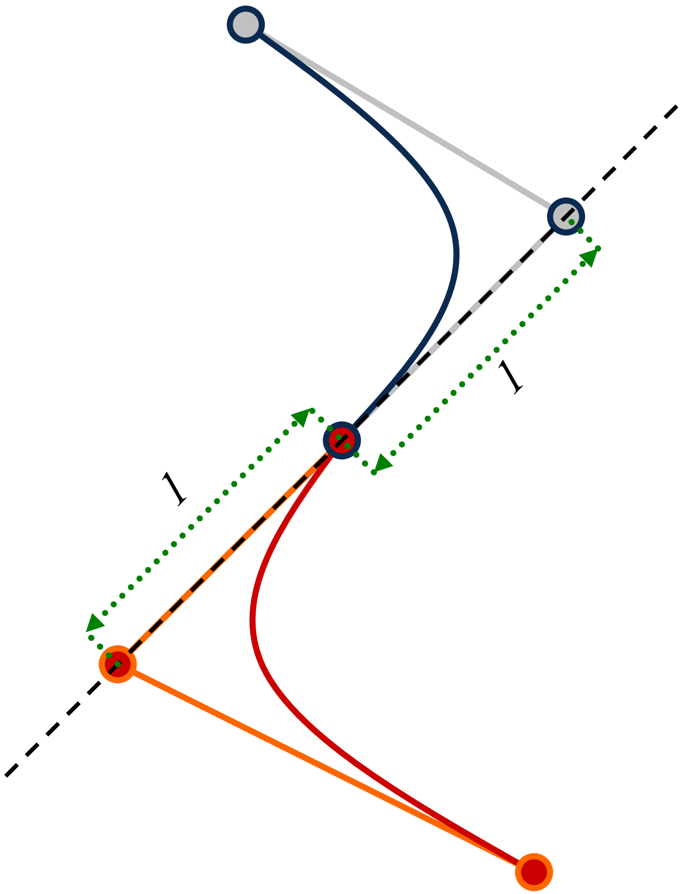
$$\underline{c}_2(t) = \begin{pmatrix} 1 - \cos t \\ -\sin t \end{pmatrix}$$

$$\dot{\vec{c}}_1(0) = \begin{pmatrix} 0 \\ -2 \end{pmatrix} \neq \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \dot{\vec{c}}_2(0)$$

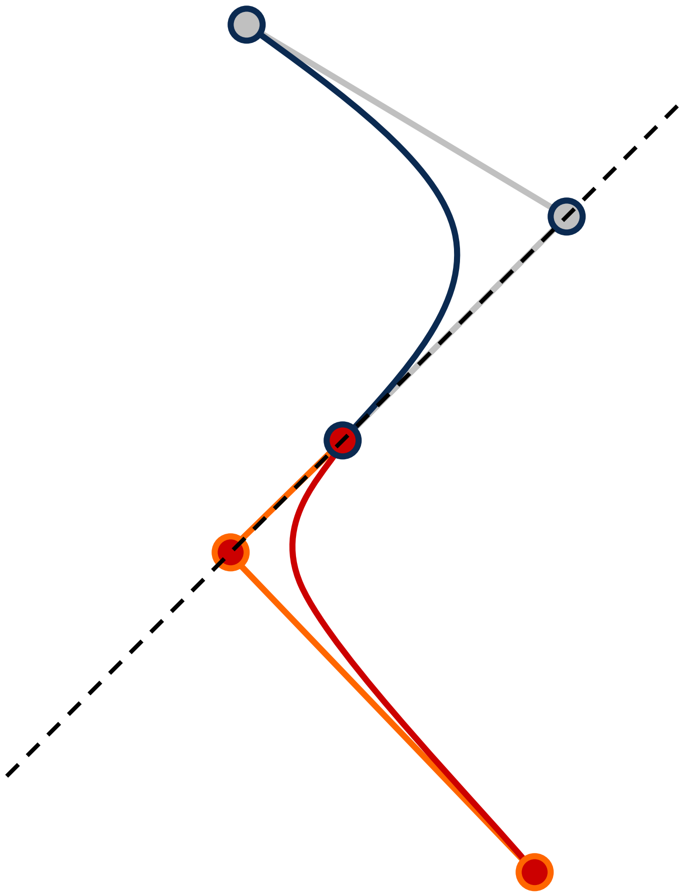
- Kurve ist G^1 – aber nur C^0 – stetig

Stetiges Aneinanderfügen
Tangentenstetigkeit

C^1-stetiger Bézier Spline



G^1-stetiger Bézier Spline

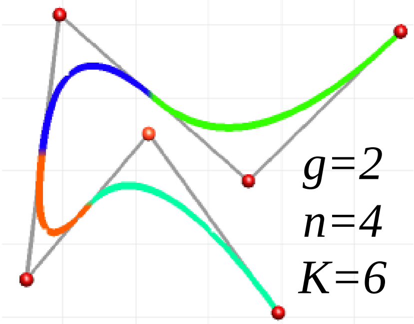


Basis-Splines Grundbegriffe

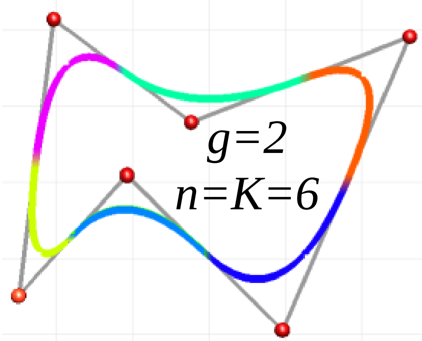
Computergraphik und Visualisierung

- Basis-Splines vom Grad g sind aus n Kurvensegmenten zusammengesetzt, die $k=g-1$ stetig (C^k) aneinanderstoßen
- Um B-Splines wie andere polynomiale Kurven verwenden zu können, werden die natürlichen Basisfunktionen $N_i^g(t)$ mit der rekursiven Konstruktionsformel nach Cox und De Boor definiert
- Die Kontrollpunkte $\underline{\mathbf{d}}_i$ heißen De Boor Punkte
- Man unterscheidet offene und geschlossene Splines

$$\underline{\mathbf{c}}(t) = \sum_{i=0}^{K-1} \underline{\mathbf{d}}_i N_i^g(t)$$



offener Spline



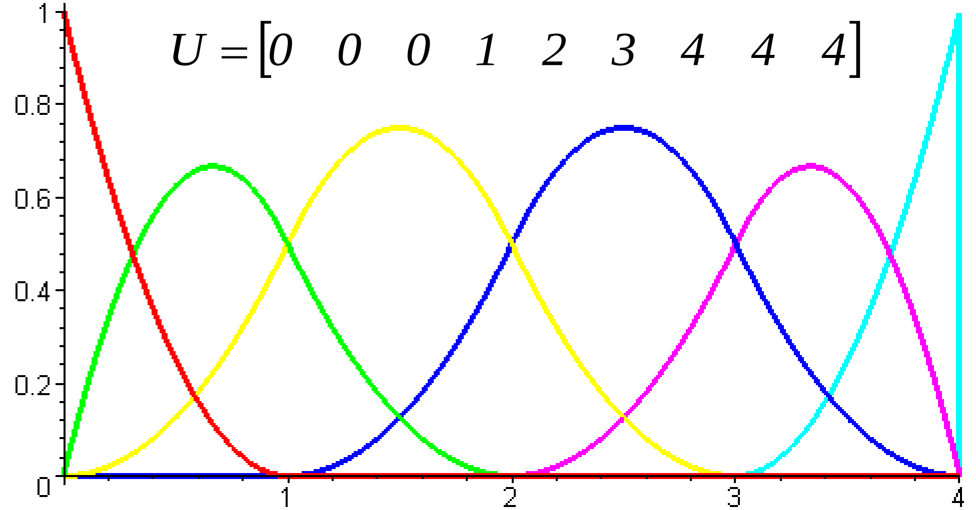
geschlossener Spline

- Bei konstantem Grad g kann beliebige Zahl $K>g$ von Kontrollpunkten genutzt werden
- es gilt
  - $K=n+g$… offene Splines
  - $K=n$… geschlossene Splines

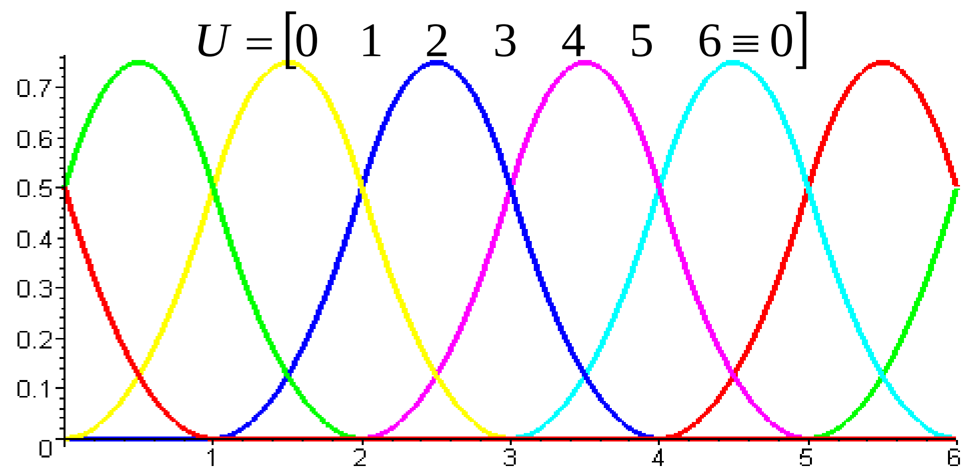
Basis-Splines
Natürliche Basis

Computergraphik und Visualisierung

- Ähnlich zur Lagrange-Interpolation wird ein Stützstellenvektor U mit m Einträgen u_i verwendet, der den Parameterbereich in Intervalle einteilt, es gilt
  - $m=K+g+1$… offene Splines
  - $m=K$… geschlossene Splines
- anstelle von Stützstelle bzw. Stützstellenvektor werden oft die Begriffe Knoten und Knotenvektor verwendet
- Sind die Knoten $u_i=a\cdot i$ äquidistant, so spricht man von einem uniformen ansonsten von einem nicht uniformen Spline



nicht uniforme natürliche Basis (offen)



uniforme natürliche Basis (geschlossen)

Basis-Splines Symbolübersicht

- i … variabel eingesetzter Laufindex
- n … Anzahl der Segmente
- g … Grad der Kurvensegmente ($g+1$ Freiheitsgrade pro Kurvensegment)
- $k=g-1$ … Stetigkeit zwischen Kurvensegmenten ($k+1=g$ Nebenbedingungen zwischen Kurvensegmenten)
- $K=n+g$ oder n … Anzahl Kontrollpunkte
- $m=K+g+1$ oder n … Anzahl Einträge im Knotenvektor

Basis-Splines
Cox De Boor-Rekursion

Computergraphik und Visualisierung

- Rekursion nach Cox De Boor am Bsp. offener Splinebasen

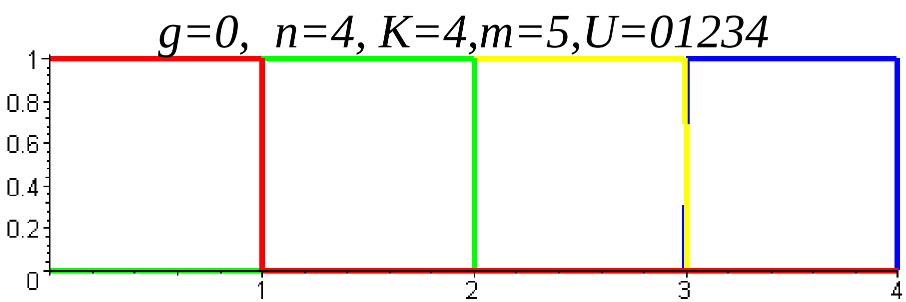
$$N_i^0(t) = \begin{cases} 1 & \text{falls} \quad u_i \le t < u_{i+1} \ \text{und} \ u_i < u_{i+1} \\ 0 & \text{sonst} \end{cases}$$

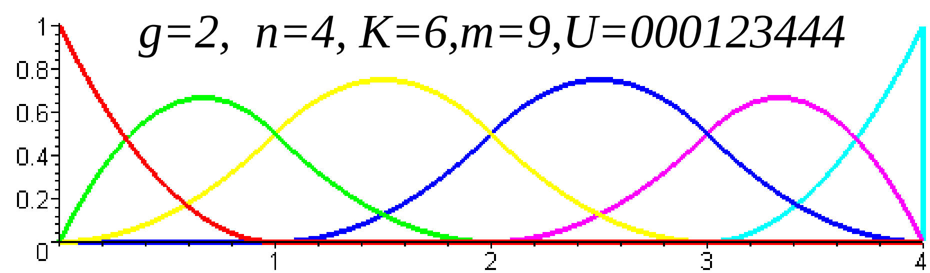
pro t nur einmal $\neq 0$

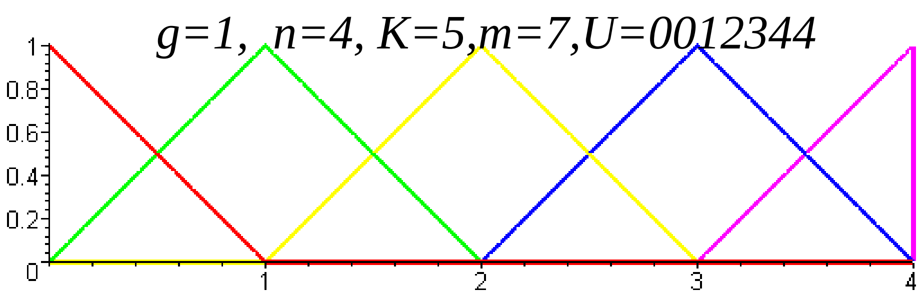
Einfluss nur von nicht degenerierten Intervallen

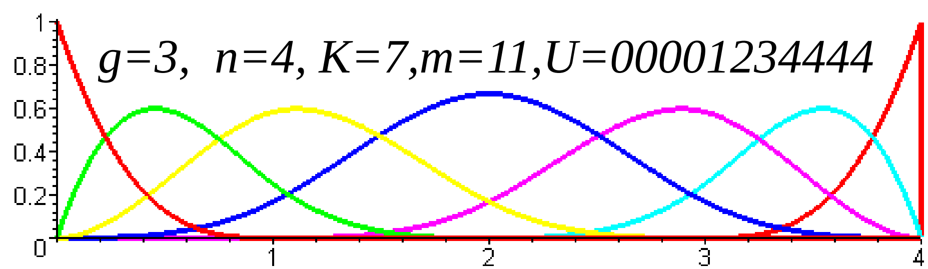
$$N_i^g(t) = \frac{t - u_i}{u_{i+g} - u_i} N_i^{g-1}(t) + \frac{u_{i+1+g} - t}{u_{i+1+g} - u_{i+1}} N_{i+1}^{g-1}(t) \qquad u_i \le t < u_{i+g+1}$$

in Klausur gegeben





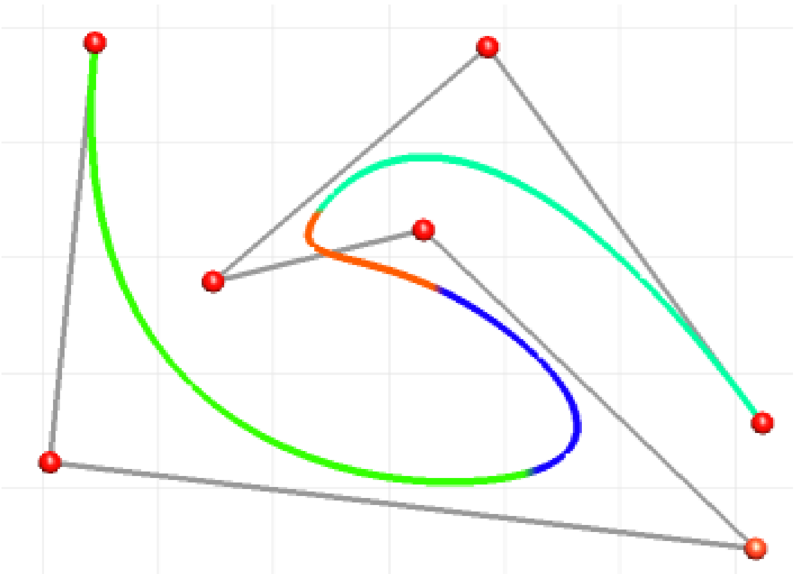




Basis-Splines
Multiplizität

Computergraphik und Visualisierung

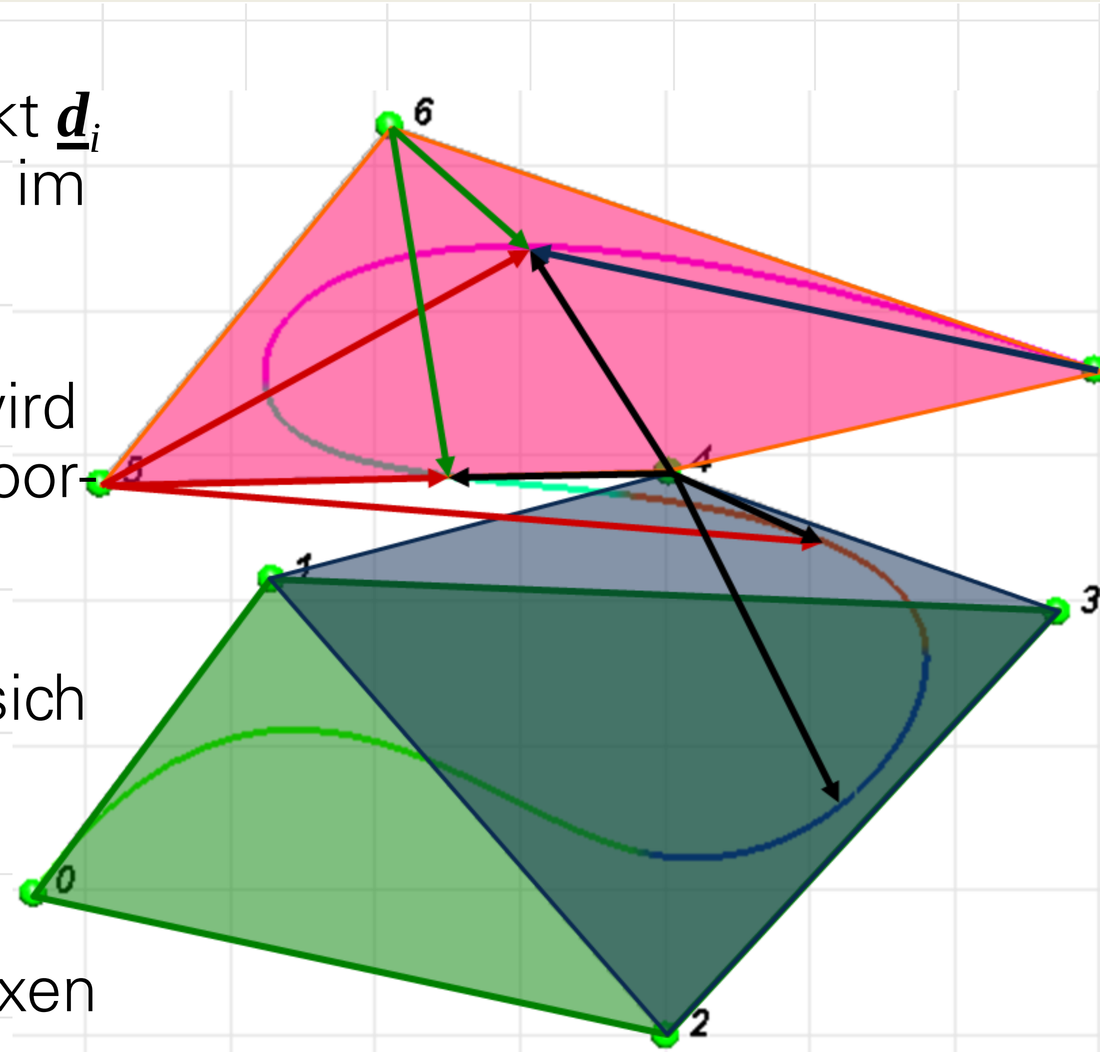
- Wählt man μ aufeinander folgende $u_{i\ldots i+\mu-1}$ gleich, so nennt man u_i einen Knoten der Multiplizität μ.
- in u_i vermindert sich die Stetigkeit auf $C^{k-(\mu-1)=g-\mu}$
- bei einer $\mu=g$ Multiplizität von u_i wird $\underline{\mathbf{d}}_i$ interpoliert.
- Bei $\mu=g+1$ bekommt der B-Spline einen Sprung
- Dies wird vor allem an den Endpunkten von offenen B-Splines genutzt



Bsp.: $n=4$, $g=3$, $K=7$, $m=11$, $U=[0,0,0,0,1,2,3,4,4,4,4]$

Basis-Splines Eigenschaften

- Die Basisfunktion N_i^g bzw. der De Boor Punkt $\underline{\mathbf{d}}_i$ beeinflusst den Spline im Parameterbereich $[u_i,u_{i+g+1})$
- Das Intervall $[u_i,u_{i+1})$ wird nur von den $g+1$ De Boor-Punkten $\underline{\mathbf{d}}_{i-g},\ldots,\underline{\mathbf{d}}_i$ beeinflusst
- Die $N_i^g(t)$ summieren sich zu 1 und sind größer gleich null
- Der Spline liegt in der Vereinigung der konvexen Hüllen von $\underline{\mathbf{d}}_i$, $\underline{\mathbf{d}}_{i+1}$, …, $\underline{\mathbf{d}}_{i+g}$



$g=3$, $n=5$, $m=12$, $K=8$, $U=000012345555$

Übersicht über Basen und Kurven

Computergraphik und Visualisierung

- Bernstein-Basis
  - affin invariant
  - positiv

$$\sum_{i=0}^{g} \underline{\boldsymbol{b}}_i B_i^g(t)$$

- Beziér-Kurve (Beziér-Punkte)
  - approximierend
  - Konvexehülleneigenschaft
  - Endtangenteninterpolation

- Lagrange-Basis
  - affin invariant
  - unabhängig

$$\sum_{i=0}^{g} \underline{\boldsymbol{q}}_i L_i^g(t)$$

- Lagrange-Kurve
  - Kontrollpunktinterpolation

- Hermite-Basis
  - lokale Definition

- Hermite-Spline
  - Kontrolltangenteninterpolation
  - lokaler Kontrolltangenteneinfluss
  - C^1-stetig

- natürliche Basis
  - affin invariant
  - positiv
  - lokale Definition

$$\sum_{i=0}^{K-1} \underline{\boldsymbol{d}}_i N_i^g(t)$$

- Basis-Splines (De Boor Punkte)
  - Endtangenteninterpolation
  - lokaler Kontrollpunkteinfluss
  - Konvexehülleneigenschaft
  - C^{g-1}-stetig