

9. Parametrische Kurven und Flächen

Polylinien bzw. Polygone sind stückweise *lineare* Approximationen für Kurven bzw. Flächen

Nachteile:

- hohe Zahl von Eckpunkten für genaue Repräsentation erforderlich
- interaktive Manipulation schwierig

Suche nach kompakteren und einfacher manipulierbaren Repräsentationen

⇒ "Freiformflächen", -kurven

insbesondere:

stückweise *polynomiale* Kurven bzw. Flächen

Geschichtlicher Hintergrund:

- ab ca. 1958 Anwendungen von Freiformflächen im CAD-Bereich, insbes. im Automobil-Karosseriebau
- **Pierre Bézier** (Renault): parametrische Repräsentation auf der Basis von Bernstein-Polynomen: System UNISURF
- **P. de Casteljaou** (Citroen)

Welche Repräsentationen von Kurven und Flächen gibt es?

Formen der Darstellung von Kurven oder Flächen:

- **explizit**
- **implizit**
- **Parametrisch**

Kurze Diskussion am Beispiel der Kurven:

Explizite Darstellung

◆ $y = f(x)$

◆ Bsp.: $y = \sqrt{x^3} - \sqrt{x}$

◆ Probleme:

- Für ein x darf es nur einen y Wert geben (Probleme z.B. ist ein Kreis nicht geschlossen darstellbar)
- Beschreibung nicht invariant gegenüber Rotationen
- Keine Kurven mit (echt) vertikalen Tangenten möglich (impliziert unendliche Steigung)

Implizite Darstellung

◆ $f(x,y) = 0$

◆ Bsp.: $x^3 - 2x^2 + x - y^2 = 0$

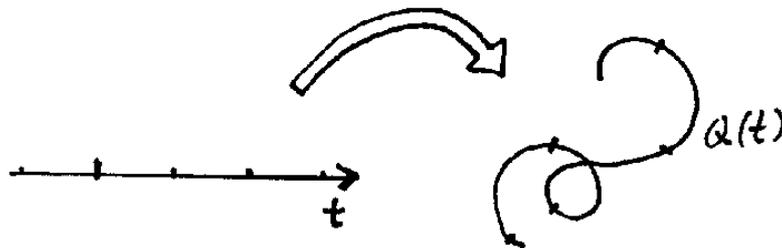
◆ Probleme:

- Gleichung kann mehr Lösungen als gewollt haben, was zur Notwendigkeit von Randbedingungen führt
- Richtung der Tangente ist schwer zu ermitteln

Parametrische Darstellung

- ◆ $Q(t) = (x(t), y(t))$
- ◆ Bsp.: $Q(t) = (t^2, t^3 - t)$

- ◆ Vorteile:
 - Keine Mehrdeutigkeiten
 - Geometrische Steigungen (potentiell unendlich) werden durch Tangentenvektoren (niemals unendlich) ersetzt
 - Invarianz gegenüber Rotationen
- ◆ Wir verwenden daher im folgenden nur die **parametrische Darstellung**



Parametrische Kurven sind i.d.R.

- (ganz rationale) Polynome
- gebrochen rationale Polynome

n-ten Grades (n: höchster auftretender Exponent),
z.B.

$$Q(u) = p_0 + p_1 u + p_2 u^2 + \dots + p_k u^k$$

In der CG werden ganz überwiegend kubische
(also $k=3$) Repräsentationen genutzt:

Definition einer 3D-Kurve durch stückweise Spezifikation dreier kubischer Polynome $x(t)$, $y(t)$, $z(t)$ mit dem Parameter t .

- Definition eines Kurvensegmentes $Q(t) = [x(t) \ y(t) \ z(t)]$ durch ($t \in [0, 1]$):

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$

$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z$$

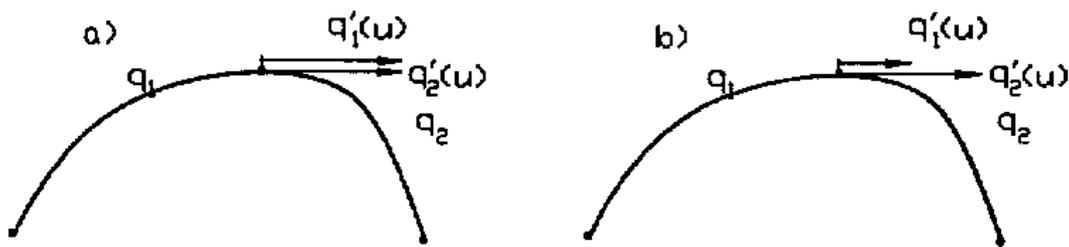
- andere Repräsentation: $Q(t) = [x(t) \ y(t) \ z(t)] = T \cdot C$ mit:

$$C = \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{pmatrix} \quad \text{und} \quad T = [t^3 \ t^2 \ t \ 1]$$

Übergangsstellen

zwei aneinander hängende Kurvensegmente sind:

- G^0 stetig \Leftrightarrow verbunden
- G^1 stetig \Leftrightarrow Steigungen stimmen überein (Tangentenvektoren),
 $TV_1 = k TV_2$
- G^n stetig \Leftrightarrow n -ten Ableitungen stimmen in Richtung überein
- C^n stetig (parametrische Stetigkeit) \Leftrightarrow n -te Ableitungen stimmen in Richtung und Länge überein



links: C^1 -stetiger Übergang, Richtung und Länge der Tangentenvektoren von 2 aufeinanderfolgenden Kurvensegmenten stimmen überein.

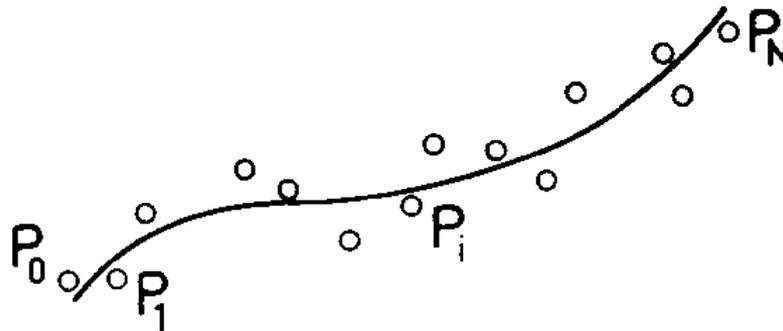
rechts: G^1 -stetiger Übergang, nur die Tangentenrichtungen stimmen überein (in der Praxis meist ausreichend, da kein Unterschied im Aussehen der Kurve).

G^2 -Stetigkeit bedeutet: die Kurve ist stetig, die Tangenten besitzen an den Übergangsstellen dieselbe Richtung und die Krümmungsvektoren stimmen dort überein.

Typen der Darstellung von Kurven (und Flächen)

- ◆ Exakte Darstellung
 - Jeder Punkt ist durch eine Formel definiert
 - Problem: Formel ist meist nicht bekannt oder zu komplex
- ◆ Interpolatorische Darstellung
 - Kurve ist durch Stützstellenbedingungen beschrieben
 - Kurve ist an den Stützstellen determiniert
- ◆ Approximative Darstellung
 - Kurve ist durch Stützstellenbedingungen beschrieben
 - Kurve ist an den Stützstellen nicht determiniert

Prinzip der Approximation:



Interpolation mit Polynomen:

- Gesucht sind die Koeffizienten eines Polynoms $P(t)$ derart, daß $P(t_i) = P_i$ für alle Stützpunkte P_i gilt
- Für $n+1$ paarweise verschiedene Stützpunkte gibt es genau ein Polynom vom Grad n , das die obige Bedingung erfüllt
- Nachteil: Berechnung der Koeffizienten ist aufwendig
- Nachteil: Änderung eines Stützpunktes bedingt Neuberechnung aller Koeffizienten

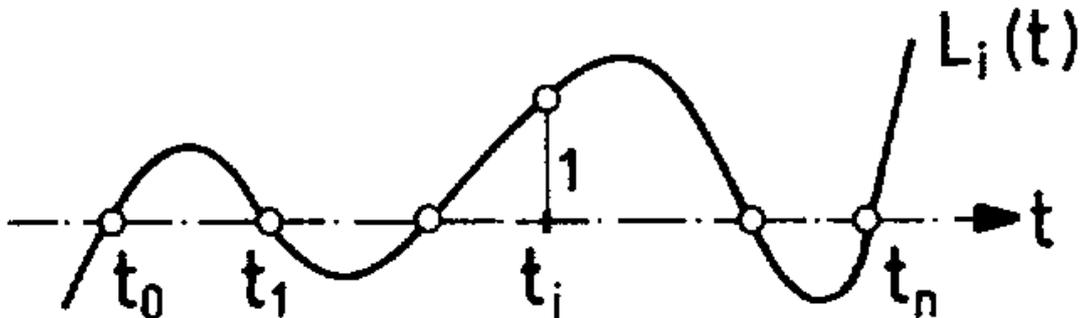
Eine mittels der Stützpunkte gewichtete Summe von sog. *Lagrange-Polynomen* liefert genau das Polynom, das durch alle P_i verläuft (**Lagrange-Interpolation**):

$$Q(t) = \sum_{i=0}^n P_i \cdot L_{n,i}(t)$$

(P_i : gegebene Punkte,
 t_i : entspr. Stützstellen,
 $i = 0, 1, \dots, n$)

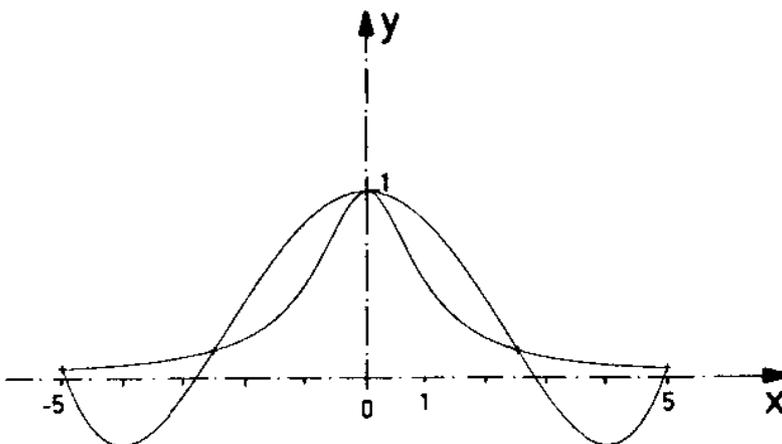
$$\text{mit } L_{n,i}(t) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t-t_j}{t_i-t_j} .$$

Die Lagrange-Polynome $L_{n,i}(t)$ haben die Eigenschaft, an der Stützstelle t_i den Wert 1 zu liefern und an allen anderen Stützstellen 0:

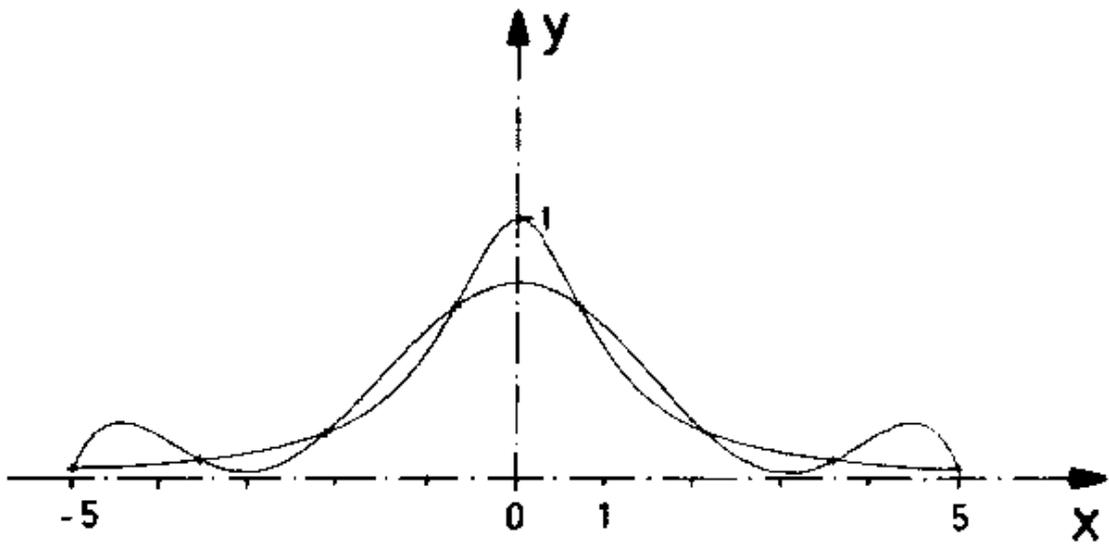


- bei Hinzufügen eines weiteren Stützpunktes müssen alle Koeffizienten neu berechnet werden
- "globales Verfahren": bei einer lokalen Änderung, z.B. eines einzelnen Stützpunktes, kann sich die Gestalt der ganzen Kurve ändern
- genereller Nachteil der Interpolation mit Polynomen höheren Grades: **Oszillationen**

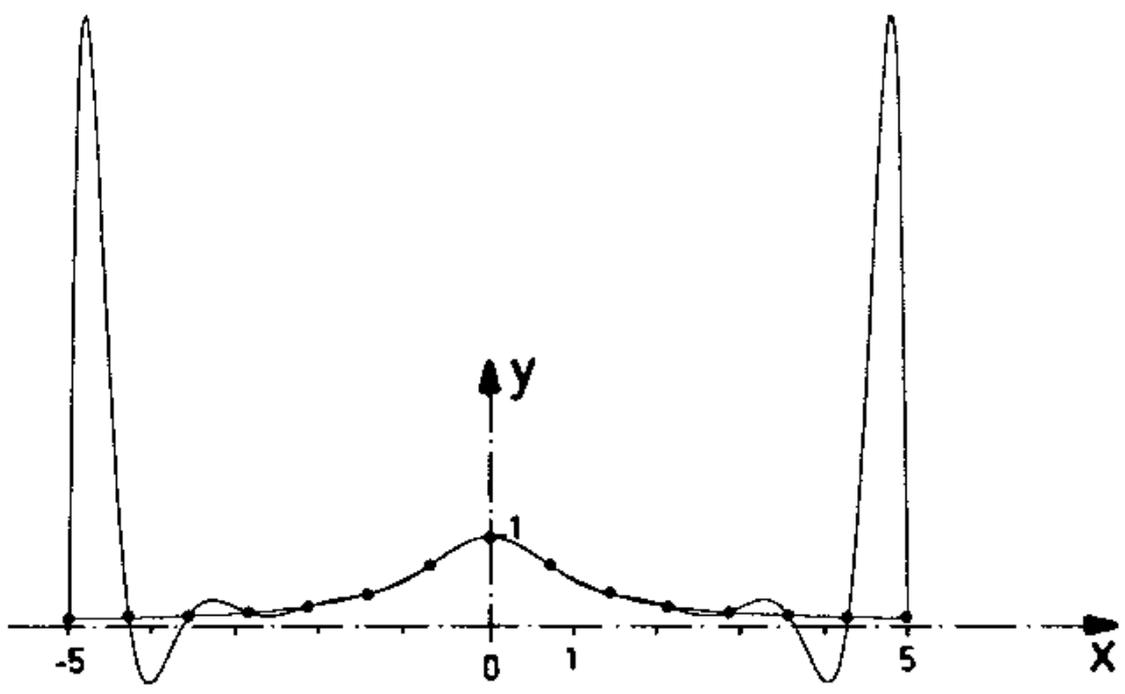
Beispiel: Interpolation des Funktionsgraphen von $y = 1/(1+x^2)$ zwischen -5 und 5



Polynom 4. Grades



Polynom 7. Grades



Polynom 14. Grades

- weiterer Nachteil: bei der Interpolation durch ein Polynom n -ten Grades weicht die Tangente stellenweise stark von der Tangente der darzustellenden Kurve ab

Abhilfe für den letzten Punkt:

man gibt neben den Stützpunkten auch die ersten m Ableitungen an den Stützpunkten vor und wählt das Polynom so, dass auch diese übereinstimmen.

für einzelnes Kurvensegment:

gegeben: $P_0^{(k)}, P_1^{(k)}$ ($k = 0; 1; \dots; m$) (Werte der Ableitungen an den beiden Stützstellen).

Der Parameterbereich sei $[t_0, t_1]$ ($t_0 < t_1$).

Ansatz für dieses Segment:

$$Q(t) = \sum_{i=0}^1 \sum_{k=0}^m P_i^{(k)} \cdot H_{i,k}(t) \quad , \quad t \in [t_0, t_1]$$

darin sind die $H_{i,k}(t)$ die sog. *Hermite-Interpolationspolynome* vom Grad $2m+1$.

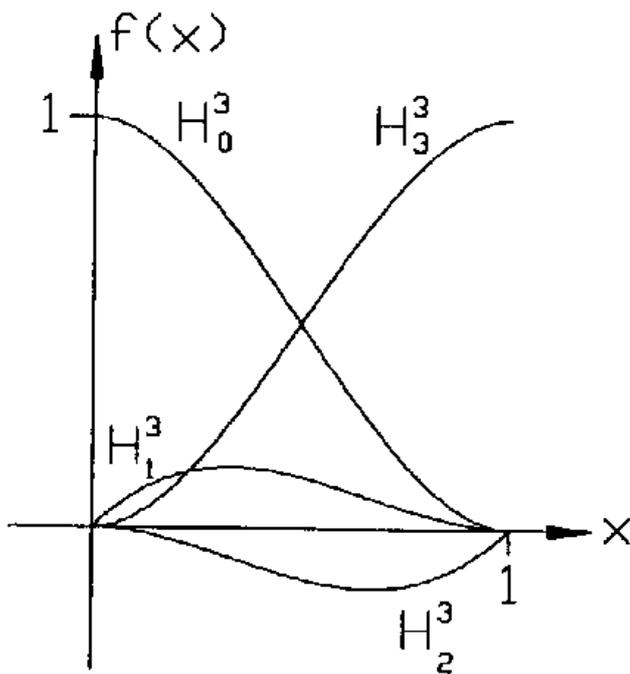
Für diese gilt (in Analogie zu den Lagrange-Polynomen) die Eigenschaft:

$$\forall i, j \in \{0, 1\}, k, r \in \{0, 1, \dots, m\}: \left. \frac{d^r}{dt^r} H_{i,k}(t) \right|_{t=t_j} = \delta_{i,j} \cdot \delta_{k,r}$$

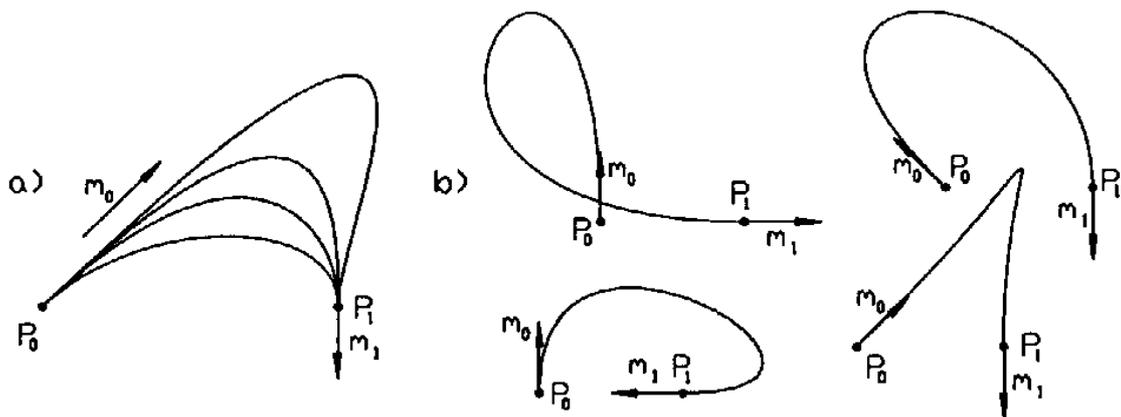
$$\left(\delta_{i,j} = \begin{cases} 0 & \text{falls } i \neq j \\ 1 & \text{falls } i = j \end{cases} \right)$$

Durch diese Gleichungen sind die $H_{i,k}$ als Polynome $2m+1$ -ten Grades eindeutig bestimmt.

Das folgende Diagramm zeigt die Hermite-Polynome vom Grad 3 über dem Intervall $[0; 1]$ (hier mit 0; 1; 2; 3 indiziert, entspr. $H_{0,0}, H_{0,1}, H_{1,1}, H_{1,0}$):



Durch die Vorgabe der Ableitungen, insbes. der ersten Ableitung (Tangentenvektor), kann der Kurvenverlauf besser gesteuert werden als allein durch die Positionen an den Stützstellen:



- a) Variation der Länge des Tangentenvektors im Anfangspunkt des Kurvensegments
- b) die Richtungen der beiden Tangenten in den Endpunkten werden variiert

zur Vermeidung von Oszillationen und globalen Auswirkungen bei lokalen Änderungen:

Zerlegung in Teilintervalle mit *verschiedenen*, nur begrenzt voneinander abhängigen Interpolationskurven

→ Idee der *Spline-Kurven*

Splines

spline (engl.): dünner Stab, von Schiffsbauern benutzt, um die richtige Form der "stringer" zu konstruieren (Planken, die in Schiffs-Längsrichtung quer zu den Spanten angebracht werden und die Außenwand des Rumpfes bilden)

Entspr. deutsches Wort: "Straklatte",

daher in der älteren dt. Lit. für Splinefunktionen z.T. auch "Strakfunktionen".

Seien $t_0, \dots, t_n \in \mathbb{R}$, $t_0 < t_1 < \dots < t_n$, Funktion S def. auf $[t_0, t_n]$.

Def.:

S heißt *Splinefunktion* der Ordnung k (oder vom Grad $k-1$), wenn gilt:

(1) S ist in jedem Teilintervall $[t_r, t_{r+1}]$ ein Polynom vom Grad $k-1$.

(2) S und die Ableitungen $S^{(m)}$, $m = 1; \dots; k-2$, sind stetig auf $[t_0, t_n]$.

S heißt *Sub-Splinefunktion*, wenn (1) gilt, S stetig ist, (2) aber nicht gilt.

Kubische Splines

$k = 4$, also Polynomsegmente vom Grad 3, zweimal stetig differenzierbar.

Eine kubische Splinefunktion beschreibt näherungsweise die Biegelinie eines dünnen Stabes durch eine vorgegebene Reihe von festen Stützhaken.

(Lösung d. Euler-Lagrange-Gleichung, Variationsrechnung, s. Hoschek & Lasser 1992)

Kubische Splines benutzen auf den Teilintervallen Hermite-Interpolationspolynome vom Grad $3 = 2m+1$ ($m = 1$).

Gegeben: $n+1$ Punkte $P_r^{(0)}$, $r = 0; \dots; n$ ($n > 0$);
 Stützstellen $a = t_0 < t_1 < t_2 < \dots < t_n = b$.

Auf dem Teilintervall $[t_r, t_{r+1}]$: Hermite-Interpolation 3. Grades

$$Q_r(t) = \sum_{i=r}^{r+1} \sum_{k=0}^1 p_i^{(k)} \cdot H_{i,k}(t) = (t^3; t^2; t; 1) \cdot M_H$$

$(t \in [t_r, t_{r+1}])$

darin ist M_H die sog. Hermite-Matrix (4,3-Matrix)
 (beachte: die $P_i^{(k)}$ werden hier als Zeilenvektoren interpretiert!)
 M_H ist unabhängig von t .

Die Berechnung von M_H erfolgt durch Einsetzen von t_r und t_{r+1} in die obige Gleichung und in die einmal differenzierte Version der obigen Gleichung. Diese 4 Einsetzungsoperationen zusammengefasst liefern:

$$\begin{pmatrix} p_r^{(0)} \\ p_{r+1}^{(0)} \\ p_r^{(1)} \\ p_{r+1}^{(1)} \end{pmatrix} = \underbrace{\begin{pmatrix} t_r^3 & t_r^2 & t_r & 1 \\ t_{r+1}^3 & t_{r+1}^2 & t_{r+1} & 1 \\ 3t_r^2 & 2t_r & 1 & 0 \\ 3t_{r+1}^2 & 2t_{r+1} & 1 & 0 \end{pmatrix}}_K \cdot M_H \xRightarrow{K \text{ regulär}} M_H = K^{-1} \cdot \begin{pmatrix} p_r^{(0)} \\ p_{r+1}^{(0)} \\ p_r^{(1)} \\ p_{r+1}^{(1)} \end{pmatrix}$$

Zur Vereinfachung nehmen wir an: $t_r = 0$, $t_{r+1} = 1$
 (dies lässt sich ggf. durch lineare Transformation des Parameterbereichs erreichen). Dann vereinfacht sich K :

$$K = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix}, \quad K^{-1} = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Daraus lassen sich die Hermite-Polynome, und damit die Splinefunktion, berechnen:

$$Q_r(t) = [t^3; t^2; t; 1] \cdot K^{-1} \cdot \begin{pmatrix} p_r^{(0)} \\ p_{r+1}^{(0)} \\ p_r^{(1)} \\ p_{r+1}^{(1)} \end{pmatrix}$$

$$H_{r,0}(t) = 2t^3 - 3t^2 + 1$$

$$H_{r+1,0}(t) = -2t^3 + 3t^2$$

$$H_{r,1}(t) = t^3 - 2t^2 + t$$

$$H_{r+1,1}(t) = t^3 - t^2$$

$$Q_r(t) = A_r \cdot t^3 + B_r \cdot t^2 + C_r \cdot t + D_r$$

mit $A_r = 2(p_r^{(0)} - p_{r+1}^{(0)}) + p_r^{(1)} + p_{r+1}^{(1)}$

$$B_r = 3(p_{r+1}^{(0)} - p_r^{(0)}) - 2p_r^{(1)} - p_{r+1}^{(1)}$$

$$C_r = p_r^{(1)}$$

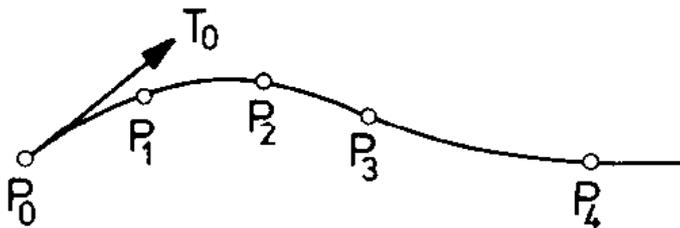
$$D_r = p_r^{(0)}$$

Eine explizite Vorgabe der 1. Ableitungen auch in den inneren Stützstellen ist i. allg. nicht sinnvoll
fordere stattdessen: Stetigkeit der 1. Ableitung (siehe Def. der Spline-Funktionen)

$2n$ Werte bislang unbestimmt (für jedes Segment die 1. Ableitungen in den beiden Endpunkten)
Stetigkeitsforderung liefert $n-1$ weitere Bedingungen
 $\Rightarrow n+1$ Koeffizienten immer noch unbestimmt

fordere zusätzlich die Stetigkeit der 2. Ableitung
liefert $n-1$ weitere Bedingungen
 $\Rightarrow 2$ Koeffizienten immer noch unbestimmt

man gibt die 1. Ableitung in den Endpunkten vor



\Rightarrow liefert Bestimmungsgleichungen für die restlichen 2 Koeffizienten

Berechnung der Koeffizienten für die kubischen Polynome in den Teilintervallen läuft auf Lösen eines lin. Gleichungssystems heraus!

Zusammenfassung zur stückweisen Polynom-Interpolation
(gilt entsprechend auch für Approximationsansätze):

- Grad $n = 1$:
Polygonzug, unstetige Steigungen an den Eckpunkten
- Grad $n = 2$:
In 3D können nur planare Kurven erhalten werden
(d.h. Kurve liegt immer in einer Ebene)
- **Grad $n = 3$ (übliche Wahl)**
Die vier Koeffizienten können z.B. durch Startpunkt, Endpunkt, Tangente am Startpunkt, Tangente am Endpunkt gegeben werden
- Grad $n > 3$:
Rechenaufwendig, nur in speziellen Anwendungen
— benutzt

Allgemeine Darstellung für $k=3$

$$Q(u) = \sum_{i=0}^k \mathbf{p}_i \cdot b_i(u)$$

mit $b_i(u)$: Basisfunktionen

\mathbf{p}_i : Kontrollpunkte

$$Q(u) = [x(u) \ y(u) \ z(u)] = [u^3 \ u^2 \ u \ 1] \cdot M \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

Parametervektor Basismatrix Geometrievektor
(Kontrollpunkte)

Blendingfunktionen

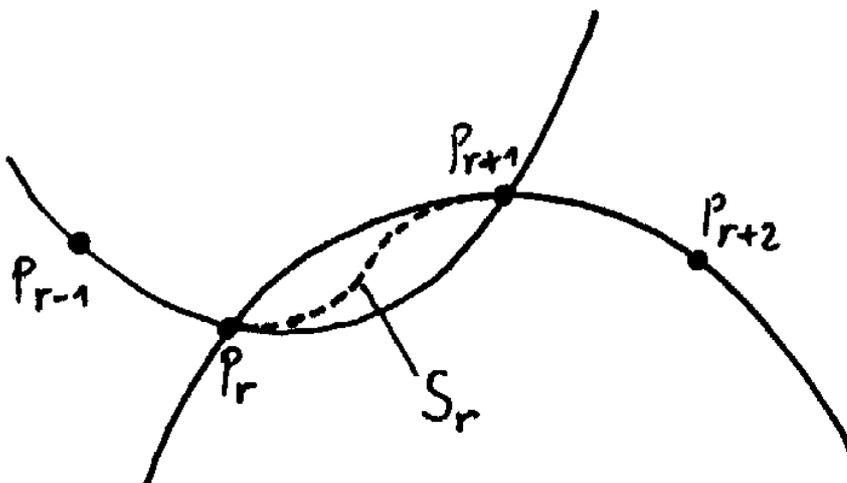
- ◆ Der erste Faktor ist der Parametervektor U
- ◆ Der zweite Faktor, die 4×4 Matrix M heißt **Basismatrix**
- ◆ Der dritte Faktor heißt **Geometrievektor G** , die G_i sind geometrische Nebenbedingungen (z.B. Punkte oder Tangentenvektoren)
- ◆ Das Produkt aus U und M ergibt die *Blendingfunktionen* (diese gewichten die geometrischen Nebenbedingungen G_i)

Ergänzung:

Einfaches Interpolationsverfahren für 2D-Anwendungen:
Parabolic Blending

Es seien $n+1$ Punkte P_0, P_1, \dots, P_n im \mathbb{R}^2 gegeben ($n > 2$).

Idee: Man bestimmt für je 3 aufeinanderfolgende Punkte das Lagrange-Polynom 2. Grades (Parabel) und bildet zwischen je 2 Punkten eine gewichtete "Mittel-Kurve" S_r aus den beiden überlappenden Parabeln:



Parabeln Q_r :

$$Q_r(0) = P_r$$

$$Q_r(1) = P_{r+1}$$

$$Q_{r+1}(0) = P_r$$

$$Q_{r+1}(1) = P_{r+1}$$

die gesuchte Kurve S_r als gewichtetes Mittel der Parabeln:

$$S_r(t) = (1-t) Q_r(t) + t Q_{r+1}(t) \quad (t \in [0; 1].)$$

Damit: Stetigkeit der aus den S_r zusammengesetzten Kurve und der 1. Ableitung.

Randfälle: Kurvenstücke zwischen P_0 und P_1 sowie zwischen P_{n-1} und P_n , dort kein Blending, man behält die Parabelstücke bei.

Man erhält:

$$S_0(t) = P_0 \cdot \frac{(t-1)(t-2)}{2} - P_1 \cdot t(t-2) + P_2 \cdot \frac{t(t-1)}{2}$$

$$S_r(t) = (1-t) \cdot \left(P_{r-1} \cdot \frac{t(t-1)}{2} - P_r \cdot (t+1)(t-1) + P_{r+1} \cdot \frac{t(t+1)}{2} \right) \\ + t \cdot \left(P_r \cdot \frac{(t-1)(t-2)}{2} - P_{r+1} \cdot t(t-2) + P_{r+2} \cdot \frac{t(t-1)}{2} \right)$$

und $S_{n-1}(t)$ analog zu $S_0(t)$.

$$\Rightarrow \forall r \in \{0; 1; \dots; n-1\}: S_r(0) = P_r, S_r(1) = P_{r+1},$$

$$\forall r \in \{0; 1; \dots; n-2\}: \frac{d}{dt} S_r(t) \Big|_{t=1} = \frac{d}{dt} S_{r+1}(t) \Big|_{t=0}$$

somit hat man die stetige (einmalige) Differenzierbarkeit der Gesamtkurve sichergestellt.

Eigenschaften:

- Die parabolischen Blendingfunktionen sind Subsplinesfunktionen dritten Grades
- leicht zu programmieren
- lokale Interpolation, Oszillation gering.