

Kapitel 4

Methoden der Computational Geometry

4.1 Interpolation

An N Stützstellen x_1, \dots, x_N seien Messdaten f_1, \dots, f_N gegeben.

Annahme: Es existiert ein funktionaler Zusammenhang zwischen den f_i .

Man unterscheidet:

- Interpolation: Gesucht \tilde{f} mit $\tilde{f}(x_j) = f_j$ für $1 \leq j \leq N$.
- Approximation: Gesucht \tilde{f} mit $\tilde{f}(x_j) \approx f_j$ für $1 \leq j \leq N$.

Interpolationsproblem:

Gegeben sei ein n -dimensionaler Raum $U \subseteq \mathbb{C}^0(\Omega)$ mit N Wertepaaren.

u_1, \dots, u_N sei eine Basis von U_N .

Dann kann man die Interpolante als $u(x) = \sum_{j=1}^N \alpha_j u_j$ schreiben.

Interpolationsmatrix:

$$\begin{pmatrix} u_1(x_1) & \cdots & u_N(x_1) \\ \vdots & & \vdots \\ u_1(x_N) & \cdots & u_N(x_N) \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix}, \quad A \cdot \alpha = f.$$

Die u_j sind linear unabhängig (Basis).

Bemerkung 4.1

$$u_j(x_i) = \delta_{ij} := \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (\text{Lagrange-Basis oder kardinale Basis})$$

$$\Rightarrow u(x) = \sum_{j=1}^N f(x_j) u_j(x)$$

4.2 Interpolation mit Polynomen

Lagrange-Basis für Polynome $\Pi_n(\mathbb{R}), x \in \mathbb{R}$, Polynome n -ten Grades. Es gilt:

$$L_j(x_k) = \prod_{i=0, i \neq j}^n \frac{x_k - x_i}{x_j - x_i} = \delta_{jk} \quad (4.1)$$

Satz 4.1 Gegeben sind $n+1$ verschiedene Punkte x_0, \dots, x_n . Dann lässt sich das Interpolationspolynom mit der Lagrange-Basis (4.1) darstellen:

$$P_f(x) = \sum_{j=0}^N f_j L_j(x), \quad x \in \mathbb{R}.$$

$P_f(x)$ ist eindeutig bestimmt.

Monombasis: $1, x, x^2, x^3, \dots, x^N, x^{N+1}$.

$P(x_i) = \sum_{j=0}^N a_j x_i^j = f_i, \quad 0 \leq i \leq N$ führt zu einem LGS V:

$$V = (x_i^j) = \begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^N \\ \vdots & & & & \\ 1 & x_N & x_N^2 & \dots & x_N^N \end{pmatrix} \text{ Vandermonde-Matrix}$$

Weitere Verfahren:

- Newton:
 - Liefert das gleiche Polynom wie die Konstruktion über Lagrange-Basen.
 - Vorteil bei Hinzunahme zusätzlicher Messpunkte: Es muss nicht jedesmal das gesamte Polynom neu konstruiert werden wie bei Lagrange.
- Hermite:
 - $x_i \rightarrow f_i$
 - zusätzlich: $x_i \rightarrow f'_i, f''_i$, Ableitungen an Stützstellen festgelegt.

Beispiel 4.1 (Runge):

Interpolante zu $\frac{1}{25x^2+1}$ auf $I = [-5, 5]$. Berechne $\Pi_N(I)$. Für $h \rightarrow 0, n \rightarrow \infty$ wird der Interpolationsfehler immer größer! In Abbildung 4.1 sind auf der linken Seite Interpolationspolynome verschiedenen Grades und auf der rechten Seite die zugehörigen Interpolationsfehler zu sehen. Die Polynome oszillieren an den Rändern des Intervalls mit großen Schwankungen.

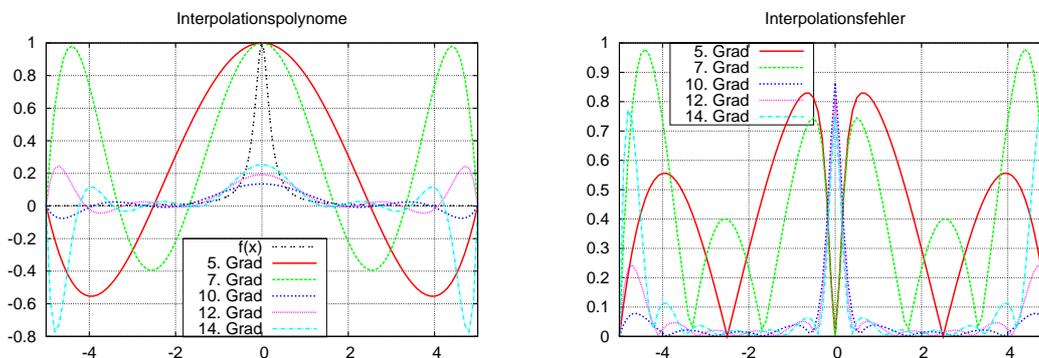


Abbildung 4.1: Die Interpolationspolynome für das Interpolationsproblem von Runge mit zugehörigen Interpolationsfehlern.

4.3 Kurven und Flächen mit Bézier-Polynomen

Definition 4.1 Eine Familie von Funktionen $\{\beta_j\}$, $j \in I$, $\beta_j : T \subseteq \mathbb{R}^k \rightarrow \mathbb{R}$ heißt (positive) Zerlegung oder Teilung der Eins auf T , falls:

1. $\sum_{j \in I} \beta_j(t) = 1, \quad \forall t \in T$
2. $\beta_j(t) \geq 0, \quad \forall t \in T$
3. $\beta_j(t) \neq 0$ für endliche viele $j \in I$ bei festem $t \in T$.

Nach dem binomischen Lehrsatz gilt:

$$1 = ((1-t) + t)^n = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} \cdot t^i$$

Die Bernstein-Polynome n -ten Grades in t auf dem Intervall $[0, 1]$ sind gegeben durch:

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} \cdot t^i.$$

Sie bilden eine Zerlegung der 1. Bézier hat sie in das rechnergestützte Konstruieren eingeführt.

Mit einer Variablentransformation $t = \frac{u-a}{b-a}$ sind die Bernstein-Polynome im Intervall $[a, b]$ gegeben durch:

$$B_{i,[a,b]}^n(u) = \frac{1}{(b-a)^n} \cdot \binom{n}{i} (b-u)^{n-1} (u-a)^i$$

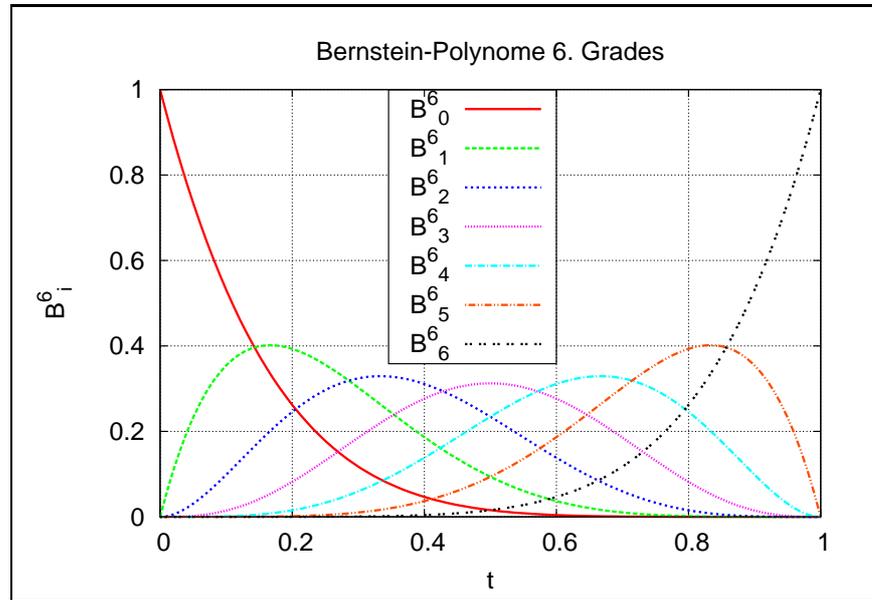


Abbildung 4.2: Bernsteinpolynome vom Grad 6

Satz 4.2 Die Bernstein-Polynome haben für $t \in [0, 1]$ die Eigenschaften:

1. $t = 0$ ist i -fache Nullstelle von B_i^n .
2. $t = 1$ ist $(n - i)$ -fache Nullstelle von B_i^n .
3. Aus der Symmetrie der Binomialkoeffizienten lässt sich folgende Symmetrie der Bernsteinpolynome folgern: $B_i^n(t) = B_{n-i}^n(1 - t)$.
4. $0 \leq B_i^n(t) \leq 1 \forall t \in [0, 1]$, $B_i^n(t) > 0 \forall t \in (0, 1)$.
5. B_i^n hat in $[0, 1]$ genau ein Maximum bei $t_{max} = \frac{i}{n}$.
6. $\{B_i^n(t)\}_{i=0}^n$ sind linear unabhängig und bilden eine Basis für Π_n .
7. Die B_i^n genügen der Rekursionsformel

$$\begin{aligned}
 B_i^n(t) &= t \cdot B_{i-1}^{n-1}(t) + (1-t) \cdot B_i^{n-1}(t), \quad i = 1, \dots, n-1 \\
 B_0^n(t) &= (1-t) B_0^{n-1}(t) \\
 B_n^n(t) &= t \cdot B_{n-1}^{n-1}(t).
 \end{aligned} \tag{4.2}$$

8. Die Ableitungen der Bernstein-Polynome sind gegeben durch

$$\partial_t (B_i^n(t)) = \begin{cases} -n \cdot B_0^{n-1}(t) & i = 0 \\ n \cdot (B_{i-1}^{n-1}(t) - B_i^{n-1}(t)) & i = 1, \dots, n-1 \\ n \cdot B_{i-1}^{n-1}(t) & i = n \end{cases}$$

4.3.1 Bézier-Darstellung eines Polynoms

Da $\{B_i^n\}$ eine Basis des $\Pi_n(\mathbb{R})$ bildet, lässt sich jedes Polynom $p \in \Pi_n$ darstellen als:

$$p(t) = \sum_{i=0}^n \beta_i B_i^n(t) \quad (\text{Bézier-Darstellung von } p)$$

β_i sind Bézier-Koeffizienten und $(\frac{i}{n}, \beta_i) \in \mathbb{R}^2$, $i = 0, \dots, n$ Bézier-Punkte. Die Verbindung der Bézier-Punkte heißt Bézier-Polygon.

Beispiel 4.2 Bézier-Polynom zu Bézier-Punkten $(0, 0)$, $(\frac{1}{3}, -1)$, $(\frac{2}{3}, -1)$, $(1, 1)$:

$$p(t) = 0 \cdot B_0^3(t) - B_1^3(t) - B_2^3(t) + B_3^3(t) = t^3 + 3t^2 - 3t$$

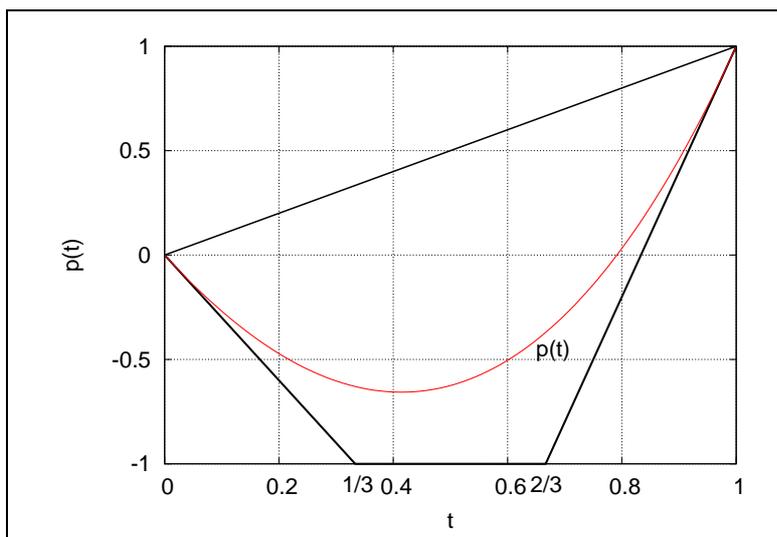


Abbildung 4.3: Bézier-Polynom zu gegebenen Stützpunkten

Satz 4.3

1. Der Graph eines Polynoms liegt in der konvexen Hülle seiner Bézier-Punkte.
2. Der Graph eines Polynoms und sein Bézier-Polygon berühren sich tangential in $t = 0$ und $t = 1$.

3. Die Ableitung eines Polynoms mit Bézier-Koeffizienten β_i lautet:

$$p^{(k)}(t) = n \cdot (n-1) \cdot \dots \cdot (n-k+1) \sum_{i=0}^{n-k} \Delta^k \beta_i B_i^{n-k}(t)$$

$$\Delta^k \beta_i = \Delta^{k-1} \beta_{i+1} - \Delta^{k-1} \beta_i; \quad \Delta^1 \beta_i = \beta_{i+1} - \beta_i$$

4.3.2 Der Algorithmus von de Casteljau

Der Algorithmus von de Casteljau ermöglicht die Berechnung jedes Punktes auf einem Bézier-Polynom über eine Rekursion. Der Algorithmus kann auch für Bézier-Kurven (nächster Abschnitt) angewandt werden.

Die Rekursionsformel (4.2) führt an Bézier-Polynomen in einem ersten Schritt zu (im Folgenden wird die t -Abhängigkeit der Bernstein-Polynome nicht mitgeschrieben):

$$\begin{aligned} p(t) &= \sum_{i=0}^n \beta_i B_i^n = \beta_0 B_0^{n-1} + \beta_1 B_1^n + \beta_2 B_2^n + \dots + \beta_n B_n^n \\ &= \beta_0(1-t)B_0^n + \beta_1[(1-t)B_1^{n-1} + tB_0^n] + \beta_2[(1-t)B_2^{n-1}(t) + tB_1^{n-1}] \\ &\quad + \dots + \beta_n[(1-t)B_n^{n-1} + B_{n-1}^n] \\ &= [\beta_0(1-t) + t\beta_1]B_0^{n-1}(t) + [\beta_1(1-t) + t\beta_2]B_1^{n-1} + \dots \\ &\quad + [\beta_{n-1}(1-t) + \beta_n]B_{n-1}^{n-1} \\ &= \sum_{i=0}^{n-1} \beta_i^{(1)} B_i^{n-1} \end{aligned}$$

mit der Definition

$$\beta_i^{(1)} = \beta_i(1-t) + \beta_{i+1} \cdot t \quad \text{für } i = 0, \dots, n.$$

Führt man dies rekursiv fort, kommt man zu folgender Darstellung:

$$p(t) = \beta_0^{(n)}.$$

Diese rekursive Berechnung eines Polynomwerts $p(t)$ entspricht folgendem Dreiecksschema:

$$\begin{array}{ccccccc} \beta_0 & = & \beta_0^{(0)} & & & & \\ & & & \beta_0^{(1)} & & & \\ & & & & \ddots & & \\ & & & & & \beta_0^{(n)} & = p(t) \\ & & & & & & \ddots \\ & & & & & & & \beta_{n-1}^{(1)} \\ \beta_n & = & \beta_n^{(0)} & & & & \end{array}$$

und charakterisiert den Algorithmus von de Casteljau.

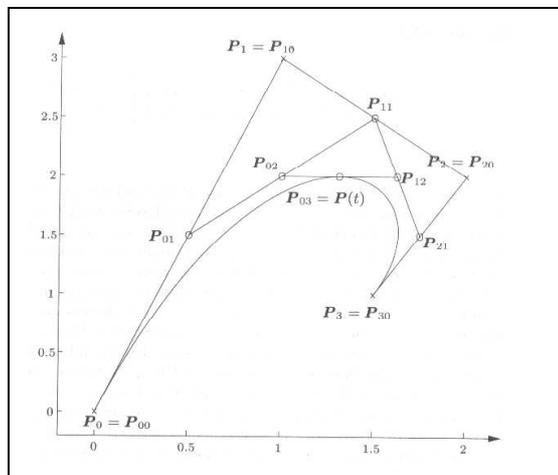


Abbildung 4.4: Algorithmus von deCasteljau

Bemerkung 4.2

- Die Koeffizienten $\beta_i^{(1)}$ besitzen folgende geometrische Bedeutung: Der Punkt $(x_i, \beta_i^{(1)})$ mit

$$x_i = (1-t) \frac{i}{n} + t \cdot \frac{i+1}{n} = \frac{i+t}{n}$$

$$\beta_i^{(1)} = (1-t) \beta_i + t \cdot \beta_{i+1}, \quad i = 0, \dots, n-1$$

liegt auf der Strecke $\left[\left(\frac{i}{n}, \beta_i \right), \left(\frac{i+1}{n}, \beta_{i+1} \right) \right]$, also auf dem Bézier-Polygon.

- Mit dem Algorithmus von de Casteljau entsteht rekursiv ein System $\beta_i^{(k)}$, $k = 1, \dots, n$. Die $\beta_i^{(k)}$ entstehen durch lineare Interpolation aus den $\beta_i^{(k-1)}$, bis $p(t) = \beta_0^{(n)}$ erreicht ist.

4.3.3 Bézier-Kurven

Der Übergang von Bézier-Polynomen zu Bézier-Kurven geschieht durch Parametrisierung.

Definition 4.2 Sei $0 \leq t \leq 1$ und seien die Funktionen $x_k(t)$, $k = 1, \dots, d$ Bézier-Polynome, dann ist $(x_1(t), x_2(t), \dots, x_d(t))^T$ eine Bézier-Kurve im Raum \mathbb{R}^d .

Bemerkung 4.3

- Mit $d = 2$ ergibt sich die Bézier-Kurve $p(t) = (x(t), y(t))^T$.
- Der de Casteljau-Algorithmus zur geometrischen Konstruktion kann über die Bézier-Punkte $P_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \in \mathbb{R}^2$ verwendet werden.

Satz 4.4 Mit den Bézier-Punkten $P_0, \dots, P_n \in \mathbb{R}^2$ in der Ebene ist die Bézier-Kurve durch $\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = P(t) = \sum_{i=0}^n B_i^n(t) \cdot P_i$ gegeben. $x(t)$, $y(t)$ sind Bézier-Polynome.

Der Algorithmus von de Casteljau lässt sich für Bézier-Kurven vollständig übertragen. Seien beispielsweise für $n = 3$ die Bézier-Punkte $P_i^0 = P_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \in \mathbb{R}^2$, $i = 0, 1, 2, 3$, gegeben. Es ergibt sich die Rekursion

$$\begin{aligned} P_i^1 &= (1-t)P_i^0 + t \cdot P_{i+1}^0, & i = 0, 1, 2 \\ P_i^2 &= (1-t)P_i^1 + t \cdot P_{i+1}^1, & i = 0, 1 \\ P_0^3 &= (1-t)P_0^2 + t \cdot P_1^2 \\ &= (1-t)((1-t)P_0^1 + t \cdot P_1^1) + t((1-t)P_1^1 + t \cdot P_2^1) \\ &= (1-t)^3 P_0^0 + 3(1-t)^2 t P_1^0 + 3(1-t)t^2 P_2^0 + t^3 P_3^0. \end{aligned}$$

Im Algorithmus von de Casteljau wird eine Bézier-Kurve durch zwei verbundene Bézier-Kurven dargestellt. Dies kann rekursiv weitergeführt werden, wobei das Kontrollpolygon der zusammengesetzten Bézier-Kurve die Originalkurve immer besser approximiert.

Satz 4.5

1. Die Menge der Punkte einer Bézier-Kurve

$$M := \left\{ P(t) = \sum_{i=0}^n B_i^n(t) P_i, \quad t \in [0, 1] \right\}$$

liegt in der konvexen Hülle der $n + 1$ Bézier-Punkte P_0, P_1, \dots, P_n .

2. Für die Randpunkte einer Bézier-Kurve gelten:

$$\begin{aligned} P(0) &= P_0, & P(1) &= P_n \\ P'(0) &= n(P_1 - P_0), & P'(1) &= n(P_n - P_{n-1}). \end{aligned}$$

3. Berechnung von $P(t)$ mit $0 \leq t \leq 1$:

$$P(t) = (P_0 \ P_1 \ \dots \ P_n) \cdot B_n \cdot \begin{pmatrix} t^n \\ \vdots \\ t \\ 1 \end{pmatrix}$$

Die $(n+1) \times (n+1)$ -Matrix B_n hat dabei die Einträge

$$b_{i+1,k+1} = \begin{cases} (-1)^{i+k+n} \binom{n-i}{k} \binom{n}{i} & i = 0, \dots, n; \ k = 0, \dots, n-i; \\ 0 & \text{sonst.} \end{cases}$$

Beispiel 4.3 Für das eben betrachtete Beispiel mit drei Bézier-Punkten ergibt sich:

$$P_3(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \end{pmatrix} \cdot \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}.$$

4.3.4 Bézier-Kurven mit speziellen Eigenschaften

Reicht zur Modellierung komplexer Körper eine einzige Bézier-Kurve nicht aus, kann der Körper oft aus einzelnen Bézier-Kurvensegmenten zusammengesetzt werden. Gegeben sei eine durch u parametrisierte Kurve $P(t)$ und eine Aufteilung des Parameterintervalls in m Teile, $u_0 < u_1 < \dots < u_m$.

Die Kurve sei aus m Bézier-Kurvensegmenten vom Grad n zusammengesetzt. Das Segment $P_j(u)$ (insgesamt m Segmente, $j = 1, \dots, m$) hat dabei die Darstellung

$$P_j(u) = \sum_{i=0}^n P_{ij} B_i^n \left(u|_{[u_{j-1}, u_j]} \right).$$

Dann gilt:

- \mathcal{C}^0 -Stetigkeit $\forall j, j+1$ -Segmente, falls: $P_{n,j} = P_{0,j+1}$.
- \mathcal{C}^1 -Stetigkeit im Punkt $u = u_j$, falls zusätzlich: $P'_j(u_j) = P'_{j+1}(u_j) \Leftrightarrow$

$$P_{n,j} = \frac{h_{j+1}}{h_j + h_{j+1}} P_{n-1,j} + \frac{h_j}{h_j + h_{j+1}} P_{1,j-1} \quad (\text{konvexe Linearkombination}) \quad (4.3)$$

mit $h_j = u_j - u_{j-1}$. Geometrisch bedeutet (4.3), dass die Strecke $[P_{n-1,j}, P_{1,j+1}]$ im Verhältnis $h_j : h_{j+1}$ unterteilt sein muss.

Zur Darstellung von zusammengesetzten Bézier-Kurven können $m \cdot n + 1$ Punkte $\in \mathbb{R}^d$ in einer d -zeiligen Matrix gespeichert werden:

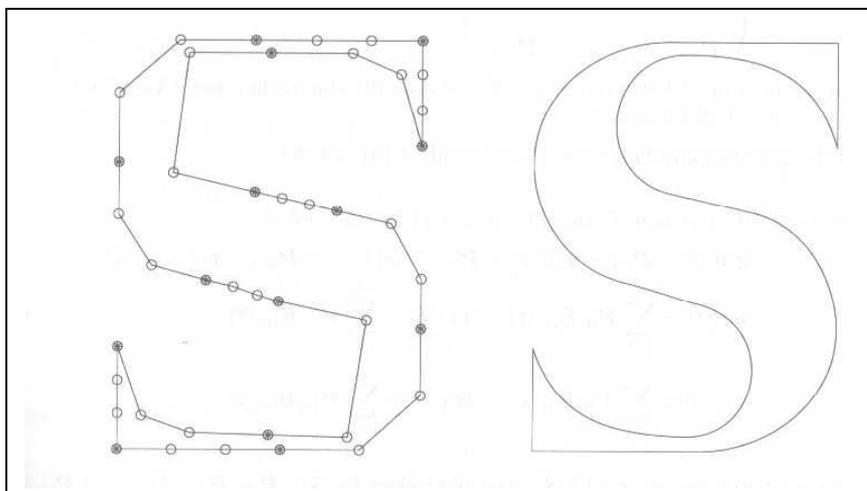


Abbildung 4.5: Darstellung eines Schriftzeichens mittels Bézier-Kurven

$$B := (P_0, P_1, \dots, P_{m \cdot n - 1}, P_{m \cdot n}).$$

Mit vorgegebenen Parameterwerten u_j besitzt das j -te Kurvensegment nun die Darstellung:

$$P_j(u) = \sum_{i=0}^n B_{(j-1) \cdot n + i} B_i^n \left(u|_{[u_{j-1}, u_j]} \right), \quad j = 1, \dots, m.$$

Bemerkung 4.4 Die Spalten von B bedingen \mathcal{C}^p -Stetigkeit bei Erfüllen gewisser Nebenbedingungen, $p \geq 1$. Die Bézierkurve ist dann p -fach stetig differenzierbar.

4.3.5 Bézier-Flächen

Eine Darstellung von Bézier-Flächen erfolgt mit Parametern s und t , o.B.d.A. auf dem Einheitsintervall.

Es liegt folgende Idee zu Grunde: Definiere 2 Bézierkurven vom Grad m und eine weitere Kurve vom Grad n , die längs zu diesen gleitet ($t(s)$).

Mit diesen Kurven können Bézierflächen als Tensorprodukte von Bézier-Polynomen dargestellt werden:

$$x(s, t) = \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_i^m(s) \cdot B_j^n(t); \quad s, t \in [0, 1].$$

P_{ij} sind die Bézier-Punkte der Fläche $x(s, t)$ und werden in der Bézier-Punktematrix zusammengefaßt. Die Bézier-Punktematrix hat folgende Gestalt:

$$\begin{pmatrix} P_{00} & P_{01} & \dots & P_{0m} \\ P_{10} & P_{11} & \dots & P_{1m} \\ \vdots & & & \vdots \\ P_{n0} & P_{n1} & \dots & P_{nm} \end{pmatrix}.$$

Das stetige Zusammensetzen von Flächensegmenten ist einfach. Es hilft folgender Satz:

Satz 4.6 Die Menge der Punkte der Bézier-Fläche $M = x(s, t)$, $s, t \in [0, 1]$ liegt in der konvexen Hülle ihrer $(n + 1)(m + 1)$ Bézier-Punkte.

Somit müssen die Bézier-Punkte am Rand übereinstimmen. Zum glatten Zusammensetzen von Flächensegmenten hilft:

Satz 4.7 Für die partiellen Ableitungen einer Bézier-Kurve $x(s, t)$ längs der Randkurve gilt:

$$\begin{aligned} \frac{\partial x(0, t)}{\partial s} &= n \cdot \sum_{j=0}^m (P_{1j} - P_{0j}) B_j^m(t); & \frac{\partial x(s, 0)}{\partial t} &= m \cdot \sum_{i=0}^n (P_{i1} - P_{i0}) B_i^n(s) \\ \frac{\partial x(1, t)}{\partial s} &= n \cdot \sum_{j=0}^m (P_{nj} - P_{n-1j}) B_j^m(t); & \frac{\partial x(s, 1)}{\partial t} &= m \cdot \sum_{i=0}^n (P_{im} - P_{i,m-1}) B_i^n(s) \end{aligned}$$

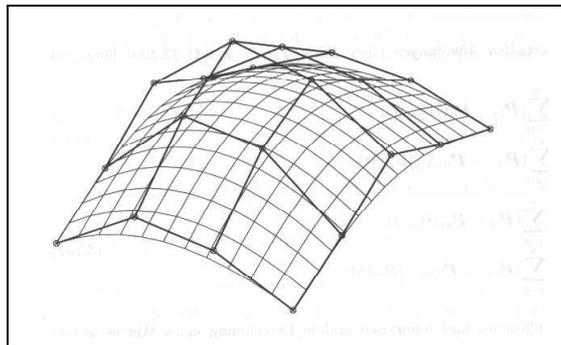


Abbildung 4.6: Bézier-Fläche mit Kontrollnetz

Zur effizienten und numerisch stabilen Berechnung eines Wertes $x(s, t)$ erfolgt ein zweimaliges Anwenden des Algorithmus von de Casteljau:

$$x(s, t) = \sum_{j=0}^m \underbrace{\left(\sum_{i=0}^n P_{ij} B_i^n(s) \right)}_{Q_j(s)} B_j^m(t).$$

Wir erhalten $(m + 1)$ von s abhängige Bézier-Punkte

$$Q_j(s) = \sum_{i=0}^n P_{ij} B_i^n(s), \quad j = 0, 1, \dots, m,$$

durch Anwendung des Algorithmus von de Casteljau für einen festen Wert s . Mit den Hilfspunkten $Q_j(s)$ führt man den Algorithmus von de Casteljau dann für jedes t durch.

Bemerkung 4.5

- *Flächen mit besonderen Eigenschaften werden aus einzelnen Bézier-Flächensegmenten zusammengesetzt.*
- *Insbesondere für einen glatten C^1 -Übergang müssen eine Reihe von Bedingungen erfüllt sein. Siehe Literatur, z.B. Bungartz, Griebel, Zenger: Einführung in die Computergraphik, Vieweg 2002.*

4.4 Kurven und Flächen mit B-Splines

Bézier-Polynome haben einige gravierende Nachteile:

- Der Polynomgrad ist abhängig von der Anzahl der Kontrollpunkte.
- Hohe Anzahl Kontrollpunkte \rightarrow hoher Polynomgrad \rightarrow Stabilitätsprobleme.
- Jeder Kontrollpunkt hat Einfluss auf das Kurvenbild.
- Auswertung eines Punktes ist linear in der Anzahl der Kontrollpunkte $O(n)$.

Bei dem computergestützten Modellieren von Körpern verwendet man deswegen eine andere Klasse von Polynomen, die sogenannten B-Splines, die obige Nachteile vermeiden.

Definition 4.3 (*Spline*):

Seien $X := a = x_0 < \dots < x_n = b$ eine Zerlegung des Intervalls $I := [a, b]$ und $m \in \mathbb{N}$. Dann heisst eine Abbildung $s : I \rightarrow \mathbb{R}$ Spline vom Grad m über der Zerlegung X , falls

1. $s \in C^{m-1}(I)$,
2. $s|_{[x_i, x_{i+1}]} \in \Pi_m(\mathbb{R})$, $0 \leq i \leq n-1$.

Im Schiffsbau zu Beginn des 20. Jahrhunderts und später im Flugzeugbau benutzte man dünne Balsaholzstäbe (Straklatten, engl. Splines) um glatte Flächen zu konstruieren.

Die Straklatte nimmt eine energieminimale Form an, welcher eine Kurve mit kleinster Krümmung entspricht, d.h. das Funktional

$$J(s) := \frac{1}{2} \int_a^b s''(x)^2 dx \quad (4.4)$$

wird bezüglich des Splines s minimiert.

Bedingung 1.) bedeutet, dass der Spline an den Teilintervallgrenzen glatt verklebt ist, d.h. für die i .te Ableitung gilt an den Segmentübergängen:

$$s^{(i)}(x_j^+) = s^{(i)}(x_j^-), \quad 0 \leq i \leq m-1.$$

Seien $f \in \mathcal{C}^2(I)$, s ein linearer Spline ($m = 1$) und $(x_i, f_i)_{i=0}^n$ eine gegebene Interpolationstabelle. Dann gilt:

$$|f(x) - s(x)| \leq \frac{1}{8} h^2 \|f\|_{L^\infty(I)}$$

mit $h = \max_{0 \leq i \leq n} (x_i - x_{i-1})$. Der Fehler bei der Spline-Interpolation reduziert sich also quadratisch (quadratische Konvergenz).

4.4.1 B-Splines

Wir erweitern die Zerlegung X zu einer unendlichen Knotenfolge

$$T : \dots \leq t_{-2} \leq t_{-1} \leq t_0 \leq t_1 \leq t_2 \leq \dots,$$

bei der Mehrfachknoten mit $t_j = t_{j+1} = \dots = t_{j+k}$ erlaubt sind.

T ist eine Zerlegung des $\mathbb{R} : \lim_{j \rightarrow \pm\infty} t_j = \pm\infty$.

Definition 4.4 (B-Splines):

Zu der bi-infiniten Knotenfolge $T := \{t_j\}_{j \in \mathbb{Z}}$ und $m \in \mathbb{N}_0$ sei

$$\omega_j^m(t) = \begin{cases} \frac{t-t_j}{t_{j+m}-t_j} = \frac{1}{t_{j+m}-t_j} \cdot t - \frac{t_j}{t_{j+m}-t_j} & t_j < t_{j+m}, \\ 0 & \text{sonst.} \end{cases}$$

Die rekursiv definierten Funktionen

$$B_j^{(0)} = \chi_{[t_j, t_{j+1}]}(t) = \begin{cases} 1 & t \in [t_j, t_{j+1}], \\ 0 & \text{sonst,} \end{cases}$$

$$B_j^{(m)}(t) = \omega_j^m(t) B_j^{(m-1)}(t) + (1 - \omega_{j+1}^m(t)) B_{j+1}^{(m-1)}(t)$$

heißen B-Splines vom Grad m zur Knotenfolge T .

Bemerkung 4.6

- m ist ein oberer Index und kein Grad eines Polynoms.
- B-Splines bilden eine Zerlegung der Eins:

$$\sum_{j \in \mathbb{Z}} B_j^{(m)} = 1, \quad t \in \mathbb{R}.$$

- B-Splines $\{B_j^{(m)}, -m \leq j \leq n-1\}$ bilden eine Basis für den B-Spline-Raum $S_m(x)$.
- Lineare Splines werden per Linearkombination aus den charakteristischen Funktionen der beteiligten Intervalle gebildet. Gewichte sind die Geraden ω_j^1 , die dadurch zu Hutfunktionen zusammengesetzt werden. Diese sind in Abbildung 4.8 zu sehen.

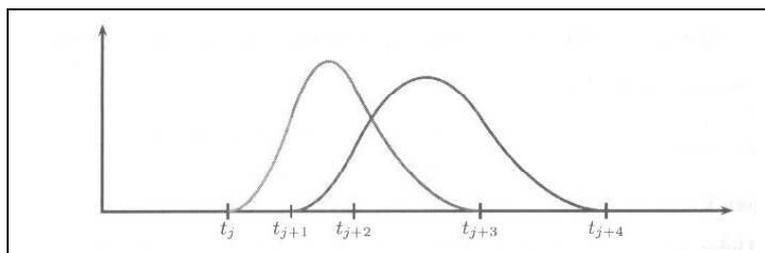


Abbildung 4.7: Quadratischer B-Spline

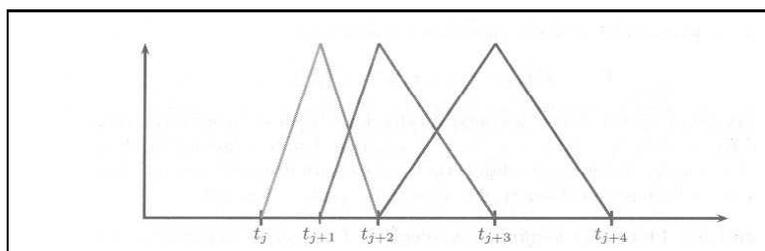


Abbildung 4.8: Lineare B-Splines

4.4.2 B-Spline-Kurven

Definition 4.5 Eine Kurve $s(t) = \sum_{j \in \mathbb{Z}} d_j B_j^{(r)}(t)$, $t \in \mathbb{R}$ heißt B-Spline-Kurve vom Grad r , die Kontrollpunkte heißen de-Boor-Punkte.

Satz 4.8 Sei $r \geq 1$. Verändert man in $s(t) = \sum_j d_j B_j^{(r)}$ den de-Boor-Punkt d_i , so ändert sich die Kurve höchstens im Bereich (x_i, x_{i+r+1}) . Die Kontrollpunkte $d_{i-r}, d_{i-r+1}, \dots, d_i$ haben nur Einfluss auf das Bild der Kurve über $[x_i, x_{i+1}]$, d.h. $s(t) = \sum_{j=i-r}^i d_j B_j^{(r)}$, $t \in [x_i, x_{i+1}]$.

Bemerkung 4.7

- Die Punktauswertung ist vom Aufwand linear im Grad r und nicht mehr in der Anzahl der Kontrollpunkte n wie bei der Interpolation mit Bézier-Polynomen.
- Praktische Anwendung haben kubische B-Splines ($r = 3$) \rightarrow Aufwand konstant, falls Auswertungsintervall $[x_i, x_{i+1}]$ in konstanter Zeit bestimmt werden kann.

Die Auswertung von B-Splines erfolgt ähnlich wie bei Bézier-Kurven mit dem De-Boor-Verfahren.

Satz 4.9 Gegeben seien eine bi-infinite, monoton wachsende Folge T und die B-Spline-Kurve $s = \sum_{j \in \mathbb{Z}} d_j B_j^{(r)}$. Für $t \in [x_k, x_{k+1}]$ definieren wir

$$d_j^{(0)}(t) = d_j, \quad k - r \leq j \leq k$$

und

$$d_j^{(l+1)}(t) = \frac{x_{j+r-l} - t}{x_{j+r-l} - x_j} d_{j-1}^{(l)}(t) + \frac{t - x_j}{x_{j+r-l} - x_j} d_j^{(l)}(t)$$

für $0 \leq l \leq r - 1$ und $k - r + l + 1 \leq j \leq k$, dann gilt:

$$s(t) = \sum_{j=k-r+l}^k d_j^{(l)}(t) B_j^{(r-l)}(t)$$

für $0 \leq l \leq r$, insbesondere $s(t) = d_k^{(r)}(t)$.

Beweis erfolgt per Induktion nach l :

$l = 0$ ist trivial.

Induktionsschritt:

$$\begin{aligned}
 s(t) &= \sum_j d_j^{(l)}(t) B_j^{(r-l)}(t) \\
 &= \sum_j d_j^{(l)}(t) \left(\frac{x_{j+r-l+1} - t}{x_{j+r+l+1} - x_{j+1}} \right) B_{j+1}^{(r-l-1)} + \left(\frac{t - x_j}{x_{j+r-l} - x_j} \right) B_j^{(r-l-1)}(t) \\
 &= \sum_j \left(\frac{x_{j+r-l} - t}{x_{j+r} - x_j} d_j^{(l)}(t) + \frac{t - x_j}{x_{j+r-l} - x_j} d_j^{(l)}(t) \right) B_j^{(r-l-1)}(t) \\
 &= \sum_j d_j^{(l+1)} B_j^{(r-l-1)}(t)
 \end{aligned}$$

Außerdem gilt $B_j^{(0)}(t) = 1$. q.e.d.

4.4.3 Kubische B-Splines

B-Splines höherer Ordnung lassen sich von Hand kaum noch berechnen, für den Computer sind sie jedoch gut geeignet, da sie aus B-Splines niedrigerer Ordnung rekursiv berechnet werden.

Aus der Knotenfolge $T : t_0 \leq t_1 \leq t_2 \leq \dots \leq t \leq t_{n+1} \leq t_{n+2} \leq t_{n+3}$ und den B-Splines vom Grad 0 kann man die B-Splines höherer Ordnung berechnen:

$$\begin{array}{cccccccc}
 B_0^0 & B_1^0 & B_2^0 & \dots & B_n^0 & B_{n+1}^0 & B_{n+2}^0 & B_{n+3}^0 \\
 \downarrow & \swarrow & \swarrow & & \downarrow & \swarrow \downarrow & \swarrow & \\
 B_0^1 & B_1^1 & \dots & \dots & B_n^1 & B_{n+1}^1 & & \\
 \downarrow & \swarrow & \swarrow & & & & & \\
 B_0^2 & B_1^2 & & & & & & \\
 \downarrow & \swarrow & & & & & & \\
 B_0^3 & & & & & & &
 \end{array}$$

Zur Berechnung eines Punktes $x \in [t_i, t_{i+1}]$ mit $t_i \leq t_{i+1}$ für B^3 -Splines werden alle B-Spline-Werte $B_i^k \neq 0$ benötigt, $k = 0, 1, 2, 3$.

Diese kann man in einem Dreieckschema anordnen:

$$\begin{array}{cccc}
 0 & 0 & 0 & 0 \\
 B_i^0 & B_{i-1}^1 & B_{i-2}^2 & B_{i-3}^3 \\
 0 & B_i^1 & B_{i-1}^2 & B_{i-2}^3 \\
 & 0 & B_i^2 & B_{i-1}^3 \\
 & & 0 & B_i^3 \\
 & & & 0
 \end{array}$$

$B_i^3(x) > 0$, falls $x \in (x_{i-2}, x_{i+2})$, $i = 4, \dots, n - 4$.

Bemerkung 4.8

- Die Werte der B-Splines wie die ihrer Ableitungen und ihrer Integrale lassen sich numerisch stabil berechnen.
- B-Splines eignen sich somit auch für die Lösung von Differential- und Integralgleichungen (FE).

Für äquidistante Stützstellen haben die inneren kubischen B-Splines die Gestalt:

$$x_i = x + ih, \quad i = 0, 1, \dots, n.$$

Dann ist für $i = 4, 5, \dots, n - 4$:

$$B_i^3(x) = \frac{1}{6h^3} \begin{cases} (x - x_{i-2})^3 & x \in [x_{i-2}, x_{i-1}] \\ h^3 + 3h^2(x - x_{i-1}) + 3h(x - x_{i-1})^2 - 3(x - x_{i-1})^3 & x \in [x_{i-1}, x_i] \\ h^3 + 3h^2(x_{i+1} - x) + 3h(x_{i+1} - x)^2 - 3(x_{i+1} - x)^3 & x \in [x_i, x_{i+1}] \\ (x_{i+2} - x)^3 & x \in [x_{i+1}, x_{i+2}] \\ 0 & \text{sonst} \end{cases}$$

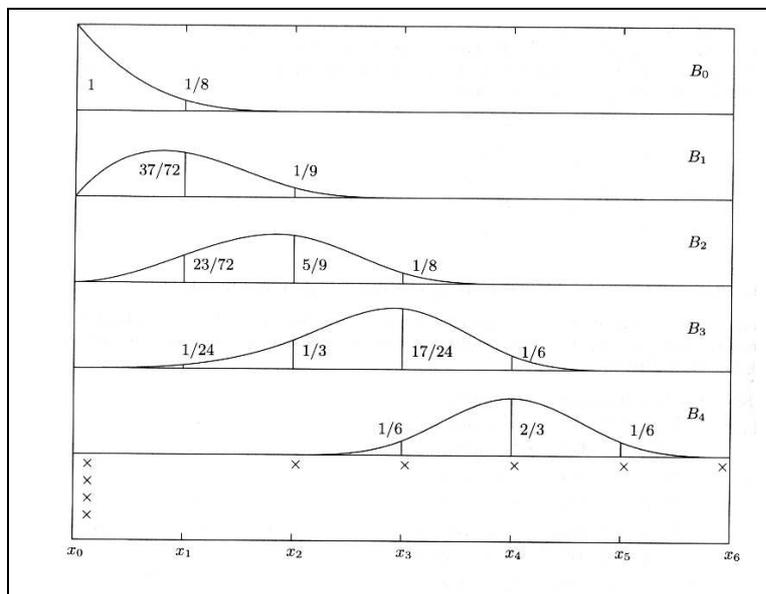


Abbildung 4.9: kubische Spline-Basis mit äquidistanten Stützstellen

Lemma 4.1 *Ist t_j ein m -facher innerer Knoten, dann ist ein kubischer B-Spline an der Stelle t_j mindestens $(3 - m)$ -mal stetig differenzierbar.*

Bemerkung 4.9 *Dadurch können Splines mit gewünschten Ecken und Sprüngen konstruiert werden. Siehe Abbildung 4.10. An der Stelle 3 ist ein dreifacher innerer Knoten. Damit hat der kubische B-Spline an dieser Stelle nur C^0 -Differenzierbarkeit, er ist an dieser Stelle somit nur noch stetig.*

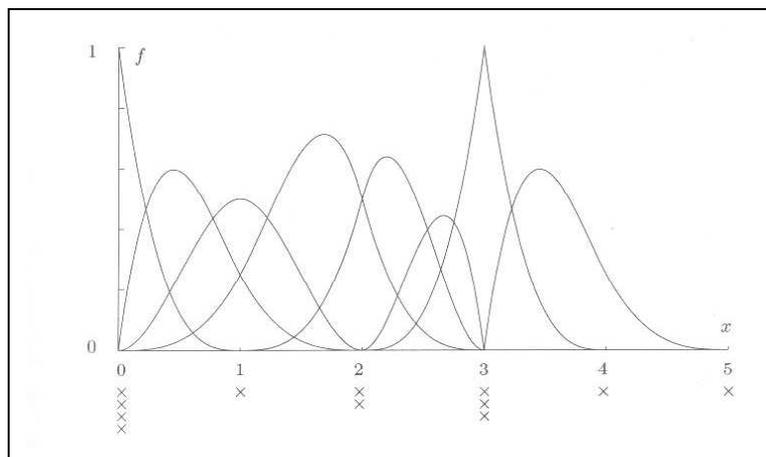


Abbildung 4.10: Kubische B-Splines mit mehrfachen inneren Knoten.

4.4.4 B-Spline-Flächen

Wir beschränken uns auf bi-kubische Tensorsplines. Seien $B_\nu(x)$, $\nu = 1, \dots, n$ und $B_\mu(y)$, $\mu = 1, \dots, l$ kubische B-Splines mit den Stützstellenmengen $\{x_i\}$ bzw. $\{y_i\}$.

Es ergeben sich zweidimensionale Ansatzfunktionen:

$$B_{\nu,\mu} = B_\nu \cdot B_\mu$$

In jedem Rechteck R_n des Gitters sind dies Polynome vom Höchstgrad 6:

$$\hat{B}_{\nu,\mu}(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j, \quad (x, y) \in R_n$$

Bemerkung 4.10 *Die $\hat{B}_{\nu,\mu}$ gehen an den Rechteckseiten zweimal stetig differenzierbar ineinander über, falls alle inneren Knoten verschieden sind.*

Das Interpolationsproblem liefert das LGS

$$\sum_{\nu=1}^n \sum_{\mu=1}^l c_{\nu\mu} B_{\nu}(x_i) B_{\mu}(y_j) = f_{ij}, \quad i = 1, \dots, n, \quad j = 1, \dots, l$$

Dieses LGS hat $n \cdot l$ Gleichungen und ebensoviele Unbekannte. Es hat Block-Band-Struktur und ist damit effizient lösbar (QR-Algorithmus).

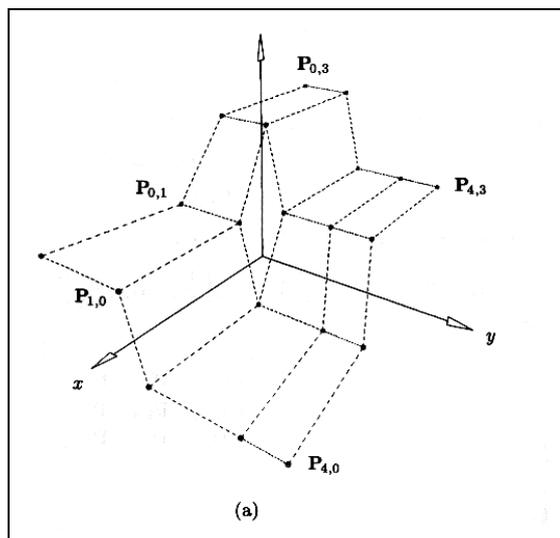


Abbildung 4.11: B-Spline-Oberflächennetz

4.5 Kurven, Flächen, Transformationen

Definition 4.6 Eine abgeschlossene Kurve im \mathbb{R}^d ist eine stetige Abbildung $f : T \rightarrow \mathbb{R}^d$ eines abgeschlossenen Intervalls $T \subseteq \mathbb{R}$. Ist $T^* \subseteq \mathbb{R}^2$ eine abgeschlossene Punktmenge mit mindestens einem inneren Punkt, so ist eine Fläche in \mathbb{R}^d über T^* als Abbildung $f^* : T^* \rightarrow \mathbb{R}^d$ definiert.

Bildmengen $f(T)$ einer Kurve oder Fläche heißen Bilder oder Spuren der Kurve oder Fläche. Leider ist die Bezeichnung in der Mathematik nicht eindeutig. Verschiedene Parametrisierung können auf das gleiche Bild führen, weshalb zwischen der Parametrisierung f und dem Bild $f(T)$ unterschieden wird. Manchmal wird die Kurve (als Abbildung) dann *Weg* genannt und das Bild des Weges als Kurve bezeichnet. Umgekehrt ist jedoch auch die Bezeichnung *Weg* für das Bild der Kurve (als Abbildung f aufgefasst) in Gebrauch.

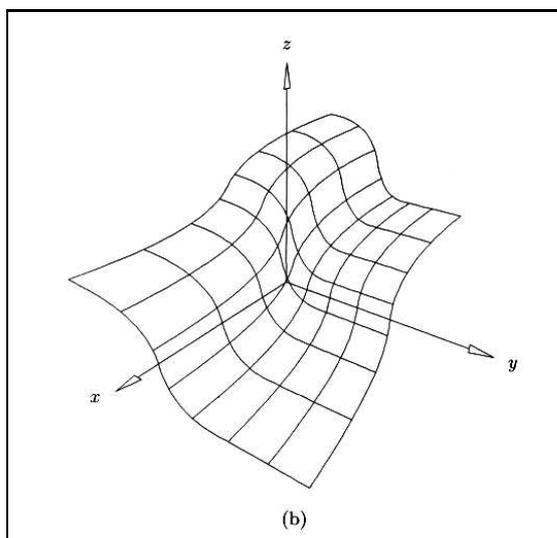


Abbildung 4.12: B-Spline-Oberfläche

Definition 4.7 Ist eine Kurve $f = (f_1, \dots, f_d) : T \subseteq \mathbb{R} \rightarrow \mathbb{R}^d$ in einem Punkt $t \in T$ differenzierbar, so heißt $f'(t) = (f'_1(t), \dots, f'_d(t))^T$ der Tangentialvektor auf $f(t)$ in t . Die Tangente an f in t ist der von $f'(t)$ aufgespannte Teilraum von \mathbb{R}^d . Der Tangentialraum einer differenzierbaren Fläche $f : T \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}^d$ am Punkt $f(t)$ ist der von den Vektoren $\frac{\partial f}{\partial t_1}(t)$ und $\frac{\partial f}{\partial t_2}(t)$ aufgespannte Unterraum des \mathbb{R}^d .

Eine Umparametrisierung einer Kurve/Fläche $f : t \rightarrow \mathbb{R}^d$ ist eine bijektive, stetig differenzierbare Abbildung $\varphi : S \rightarrow T$ mit nirgends verschwindender Ableitung, deren Umkehrabbildung wiederum stetig differenzierbar ist. Die Umparametrisierung führt zu einer zweiten Kurve $f \circ \varphi : S \rightarrow \mathbb{R}^d$ mit gleichem Bild. Die Parametrisierungen unterscheiden sich in der Regel, d. h. $\varphi \neq id$. Die Umparametrisierung muss nicht linear sein, auch kann sich der Durchlaufsinne des Kurvenbildes umdrehen.

Bemerkung 4.11

- Tangentialraum und Tangente sind invariant gegenüber Umparametrisierung.
- Definitionsbereiche T von Kurven/Flächen sind durch Polyeder charakterisiert (Intervalle, Rechtecke, Dreiecke, ...). Somit kann man sie als Konvexkombination ihrer Ecken beschreiben.

Definition 4.8 Die Punkte $x_0, \dots, x_k \in \mathbb{R}^k$ erzeugen das Polyeder

$$T = \left\{ x = \sum_{j=0}^n \lambda_j x_j; \quad \lambda_j \in [0, 1]; \quad \sum_{j=0}^n \lambda_j = 1 \right\}$$

Sind x_0, \dots, x_k in allgemeiner Lage, d.h.

$$\det \begin{pmatrix} x_0 & \cdots & x_k \\ 1 & \cdots & 1 \end{pmatrix} \neq 0,$$

dann sind für $x \in T$ die Koeffizienten

$$\lambda(x) = (\lambda_0(x), \dots, \lambda_k(x))^T \in \mathbb{R}^{k+1}$$

eindeutig definiert. Die Funktionen λ_i sind linear unabhängig auf T und es gilt $\lambda_i(x_j) = \delta_{ij}$. Die λ_i heißen baryzentrische Koordinaten.

Definition 4.9 Gegeben sei eine Kurve oder Fläche $F(t)$ mit der Darstellung $F(t) = \sum_i b_i \beta_i(t)$. Diese Darstellung heißt Kontrollnetzdarstellung der Kurve oder Fläche. $\{\beta_i\}_{i \in I}$ bildet eine Zerlegung der 1 auf T . Die b_i bezeichnen Kontrollpunkte und die Menge $\{b_i\}_{i \in I}$ wird Kontrollnetz genannt.

Definition 4.10 Eine Abbildung $A : \mathbb{R}^d \rightarrow \mathbb{R}^D$ heißt affin-linear, falls $\forall x_i \in \mathbb{R}^d$:

$$A \left(\sum_{i=1}^n \alpha_i x_i \right) = \sum_{i=1}^n (\alpha_i A(x_i)), \quad \alpha_i \in \mathbb{R}, \quad \sum_{i=1}^n \alpha_i = 1.$$

Satz 4.10 Sei $F(t) = \sum_i b_i \beta_i(t)$ eine Kontrollnetzdarstellung im \mathbb{R}^d und $A : \mathbb{R}^d \rightarrow \mathbb{R}^D$ eine affine Transformation. Dann ist auch $A \circ F$ eine Kontrollnetzdarstellung mit derselben Zerlegung der Eins und dem Kontrollnetz $\{A(b_i)\}_{i \in I}$.

Bemerkung 4.12 Dies erleichtert die Realisierung von Transformationen, Projektionen, Rotationen und Translationen bei der Darstellung von Objekten. Äquivalent ist es beispielsweise, für eine gegebene Kontrollpunktmenge zunächst die Bézier-Kurve zu berechnen und diese dann mit einer affin-linearen Abbildung zu transformieren, oder zunächst das Kontrollnetz zu transformieren und dann die Bézier-Kurve zu berechnen. Algorithmisch ist die zweite Variante in der Regel weniger aufwändig.