

7. Modelle für Flächen

gekrümmte Flächen / Freiformflächen (analog zur Kurvendarstellung)

man unterscheidet 2 Typen:

- *finite* Interpolationen / Approximationen: endliche Zahl von Stützstellen / Kontrollpunkten vorgegeben
- *transfinite*: unendliche Menge von Punkten ist vorgegeben, z.B. die Randkurven der zu modellierenden Fläche

Rand der Fläche soll exakt eingehalten werden: häufige Anforderung, z.B. in der Konstruktion von Bauteilen

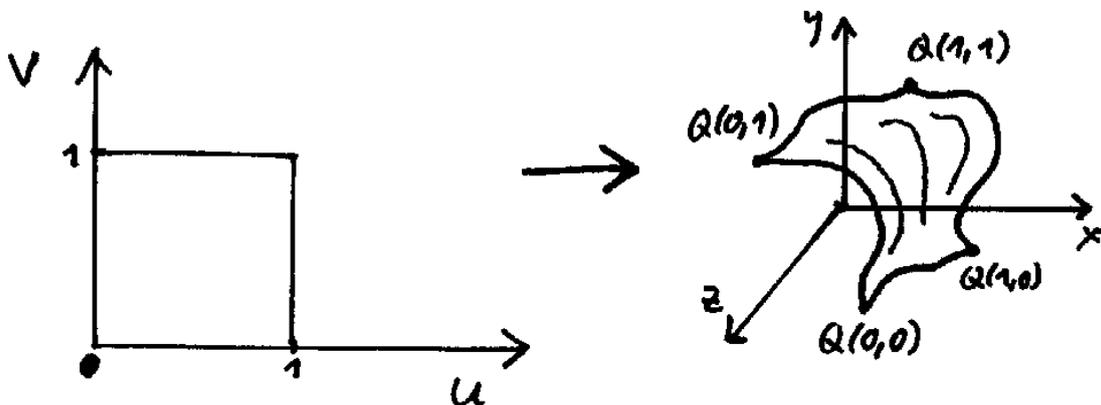
1. Transfinite Interpolationsverfahren

Ausgangspunkt:

Parameterdarstellung eines Flächenstücks mit 2 reellwertigen Parametern u, v

$$u \in [a, b], \quad v \in [c, d]$$

Abbildung des Rechtecks $[a, b] \times [c, d]$ in den dreidim. Raum:



Parameterdarstellung der Fläche:

$$Q(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix}$$

für festes $u_i \in [a, b]$: $Q(u_i, v)$

für festes $v_j \in [c, d]$: $Q(u, v_j)$ stellen Linien auf der Fläche dar

Randkurven: $Q(a, v)$, $Q(b, v)$, $Q(u, c)$, $Q(u, d)$

Vorgegebene Daten:

- Eckpunkte $Q(a, c)$, $Q(a, d)$, $Q(b, c)$, $Q(b, d)$
- Randkurven
- oft zusätzlich: Richtungsvektoren in den Randpunkten oder auf dem Rand

man definiert:

$$q^{(r,s)}(u_i, v_j) = \left. \frac{\partial^{r+s}}{\partial u^r \partial v^s} Q(u, v) \right|_{\substack{u = u_i \\ v = v_j}}$$

Oft normiert man auf $a = c = 0$, $b = d = 1$: also Einheitsquadrat als Parameterbereich.

Wie kann man zwischen den 4 Randkurven sinnvoll interpolieren?

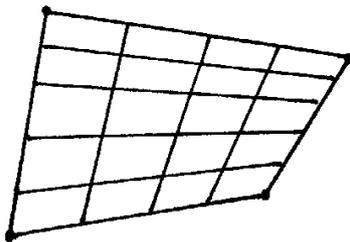
Idee: zweimalig linear, in beiden Parameterrichtungen (u und v)

Ausgangspunkt:

bilineare Interpolation zwischen den 4 *Eckpunkten*

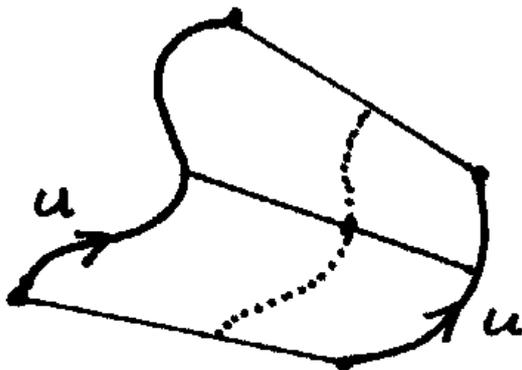
$$Q_b(u, v) = Q(0, 0)(1-u)(1-v) + Q(0, 1)(1-u)v \\ + Q(1, 0)u(1-v) + Q(1, 1)uv$$

liefert *bilineares Flächenstück* (i.allg. keine Ebene, sondern parabolisches Hyperboloid = (Ausschnitt aus) Sattelfläche!)



2. Schritt:

Lineare Interpolation zwischen 2 gegenüberliegenden Randkurven:



$$Q_1(u, v) = Q(u, 0)(1-v) + Q(u, 1)v \quad (u, v \in [0, 1]).$$

analog für das andere Randkurven-Paar:

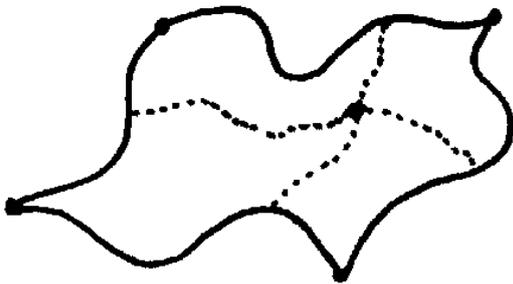
$$Q_2(u, v) = Q(0, v)(1-u) + Q(1, v)u$$

Nachteil: die beiden Interpolations-Flächen enthalten nur je eines der vorgegebenen Randkurven-Paare (das andere Paar ist durch Geradenstücke ersetzt)

3. Schritt: Addition von Q_1 und Q_2 ; zur Korrektur muss das bilineare Flächenstück Q_b wieder abgezogen werden:

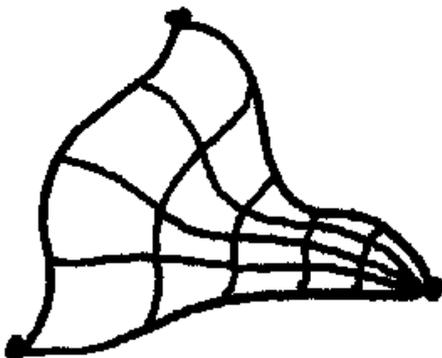
$$Q(u, v) = Q_1(u, v) + Q_2(u, v) - Q_b(u, v)$$

$\Rightarrow Q(u, v)$ hat die 4 vorgegebenen Randkurven



"Coons-Patch", "Coons-Pflaster", Coons-Flächenstück.

Coons-Fläche durch 3 Punkte:
man lasse eine der Randkurven zu einem Punkt schrumpfen



"degeneriertes Flächenstück"

– analog für 2 Punkte (2 gegebene Randkurven) oder sogar für nur 1 Punkt (1 gegebene, geschlossene Randkurve) möglich.

Um mehr Freiheitsgrade zu erhalten (z.B. für differenzierbare Übergänge zwischen aneinanderhängenden Flächenstücken!):

Verallgemeinerung auf nichtlineare Interpolation

die Faktoren $(1-u)$, u , $(1-v)$, v in obiger Formel für $Q(u, v)$ werden ersetzt durch vorgegebene Funktionen: "Überblend-Funktionen", blending functions, Mischfunktionen, Binfunktionen

$f_{0,0}(u)$ ersetzt $(1-u)$

$f_{0,1}(u)$ ersetzt u

und analog für v

(man kann auch verschiedene Funktionspaare für u und v nehmen)

[– der erste Index (0) steht für "nullte Ableitung", vgl. Erweiterung des Ansatzes unten]

Bedingung: $f_{0,0}$ und $f_{0,1}$ sind auf $[0, 1]$ def. und erfüllen

$$\forall i, j \in \{0, 1\} : f_{0,i}(j) = \delta_{i,j} = \begin{cases} 1 & \text{falls } i=j \\ 0 & \text{sonst} \end{cases}$$

Damit ergibt sich die Def. des *verallgemeinerten Coons-Patches*:

$$Q(u, v) = \sum_{i=0}^1 Q(i, v) f_{0,i}(u) + \sum_{j=0}^1 Q(u, j) f_{0,j}(v) - \sum_{i=0}^1 \sum_{j=0}^1 Q(i, j) f_{0,i}(u) f_{0,j}(v) \quad (u, v \in [0, 1])$$

Beim Aneinanderfügen von Coons-Patches fordert man die *Stetigkeit der 1. Ableitung* (und eventuell auch höherer Ableitungen) [Vermeidung von "Knicken"]

– man braucht weitere Binfunktionen: $f_{1,0}$, $f_{1,1}$
 Bedingungen an die neuen Binfunktionen:

$$\forall i, j \in \{0, 1\} : \left. \frac{d}{du} f_{r,i}(u) \right|_{u=j} = \delta_{i,j}$$

Definition der verallgemeinerten Coons-Patches mit C^1 -
 stetigem Übergang am Rand:

$$\begin{aligned} Q(u, v) = & \sum_{i=0}^1 \sum_{r=0}^1 q^{(r,0)}(i, v) f_{r,i}(u) \\ & + \sum_{j=0}^1 \sum_{s=0}^1 q^{(0,s)}(u, j) f_{s,j}(v) \\ & - \sum_{i,j,r,s \in \{0,1\}} q^{(r,s)}(i, j) f_{r,i}(u) f_{s,j}(v) \end{aligned}$$

Schwierigkeit bei diesem Modell:

Man benötigt für $q^{(1,1)}$ die gemischten 2. Ableitungen an den
 Eckpunkten des Patches (die sog. "Twist-Vektoren") –
 schwierig zu ermitteln, müssen i.allg. geschätzt werden.

Wahl der Binfunktionen f :

oft verwendet: kubische Hermite-Interpolationsfunktionen

$$\begin{aligned} f_{0,0}(u) &= H_{0,0}(u) = 2u^3 - 3u^2 + 1 \\ f_{0,1}(u) &= H_{1,0}(u) = -2u^3 + 3u^2 \\ f_{1,0}(u) &= H_{0,1}(u) = u^3 - 2u^2 + u \\ f_{1,1}(u) &= H_{1,1}(u) = u^3 - u^2 \end{aligned}$$

2. Finite Verfahren

Oft will man nur wenige Stütz- oder Kontrollpunkte haben (z.B. beim schnellen interaktiven Editieren von Freiflächen)

Vorgehen analog zur Bézier- oder B-Spline-Modellierung von Kurven

Grundidee:

- ausgehen von Kurvendarstellungen, die wir schon kennen
- "Kurven von Kurven", um auf 2D zu kommen

→ **"Tensorprodukt-Flächen"**

- gegeben eine stückweise polynomiale Kurve $F(u)$ vom Grad n

$$F(u) = \sum_{i=0}^n C_i N_i(u) \quad 0 \leq u \leq 1$$

- gegeben eine zweite stückweise polynomiale Kurve $G(v)$ vom Grad m

$$G(v) = \sum_{j=0}^m C_j N_j(v) \quad 0 \leq v \leq 1$$

- Tensorproduktfläche $S(u, v)$ beschrieben durch:

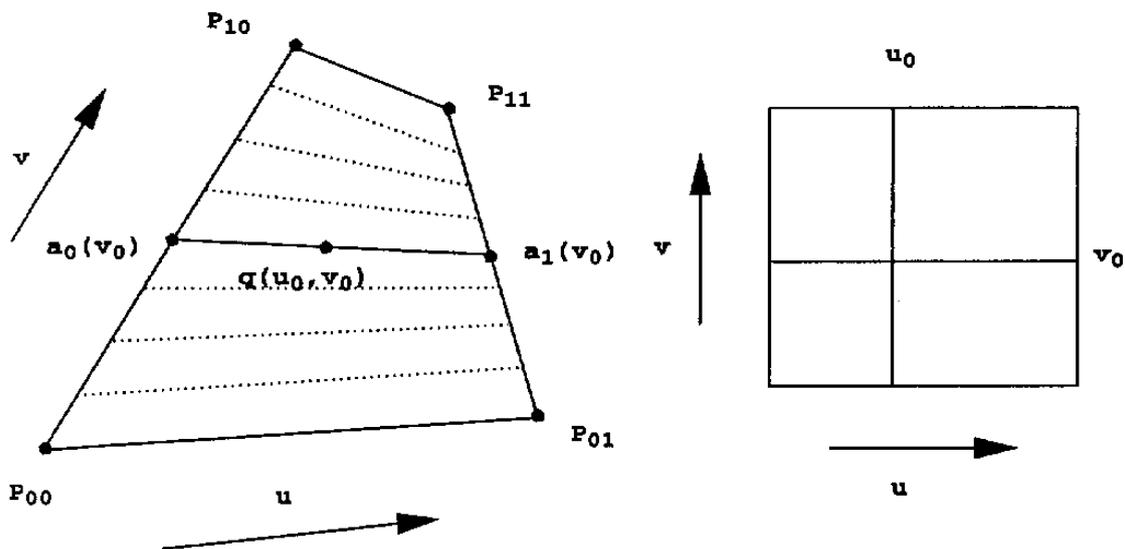
$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m C_{ij} N_i(u) N_j(v) \quad u, v \in [0, 1]$$

- Interpretation als „Kurve von Kurven“

- andere Notation:

$$S(u, v) = \sum_{i=0}^n N_i(u) C_i(v) \quad \text{mit} \quad C_i(v) = \sum_{j=0}^m N_j(v) C_{ij}$$

Konstruktion analog zur bilinearen Interpolation von Geradenstücken:



- statt der Geraden z.B. kubische Splines oder Bézier-Kurven verwenden
- Kontroll- bzw. Stützpunkte auf den Rändern und innerhalb der Fläche

Beispiel: Tensorproduktfläche, basierend auf kubischen Polynomen (häufig verwendet)

$$Q(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 p_{ij} b_i(u) b_j(v)$$

mit $p_{i,j}$ sind die 16 Kontrollpunkte

Ein Punkt $Q = (x, y, z)$ der Fläche im kartesischen Raum ist durch (u, v) im Parameterraum bestimmt.

Komplizierte Flächen werden durch eine Menge von solchen Flächenelemente (Patches) zusammengesetzt.

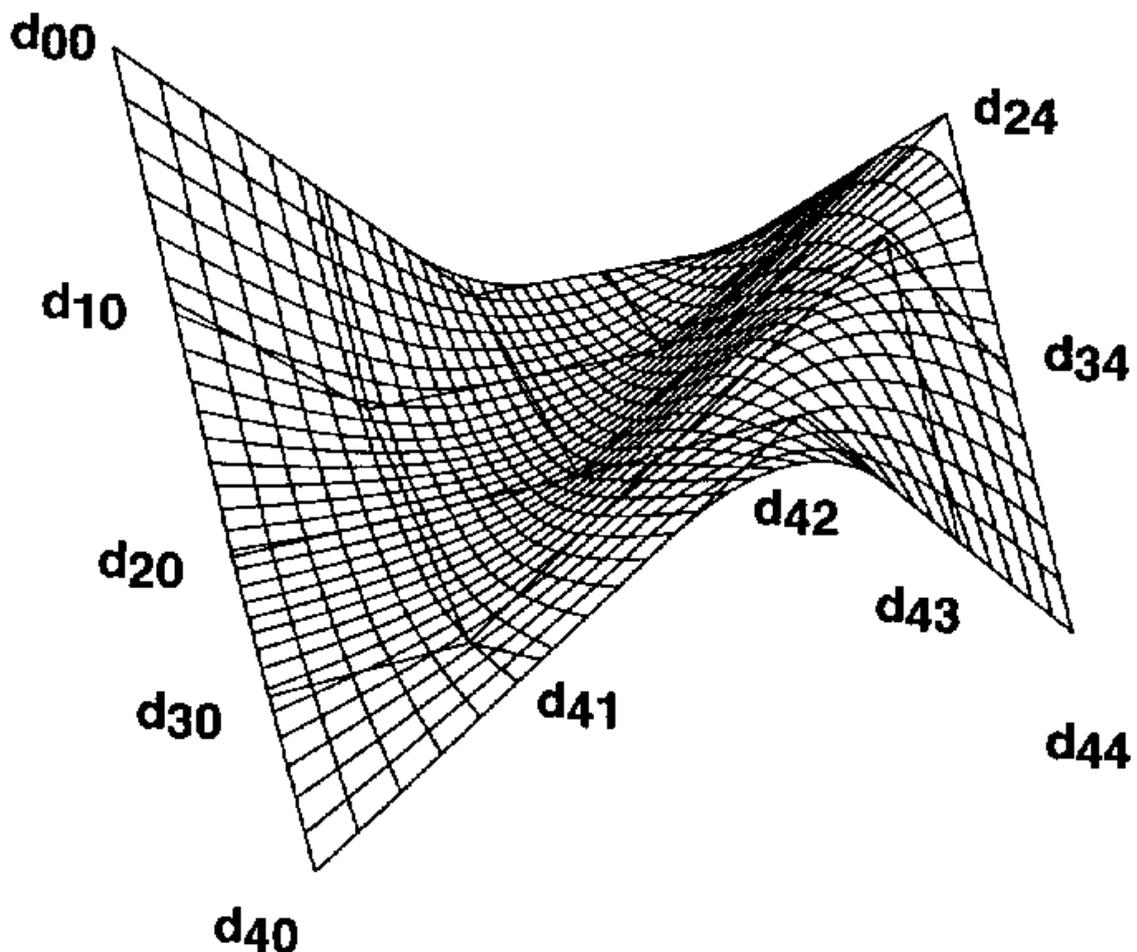
Beispiel: B-Spline-Flächen

Tensorprodukt-Flächen aus B-Spline-Kurven

$$Q(u, v) = \sum_{i=0}^r \sum_{j=0}^s P_{i,j} N_{i,m}(u) N_{j,n}(v)$$

Die Eigenschaften der B-Spline-Kurven übertragen sich, insbes. die konvexe-Hüllen-Eigenschaft.

Beispiel einer kubischen B-Spline-Fläche mit Kontrollnetz (25 Kontrollpunkte):



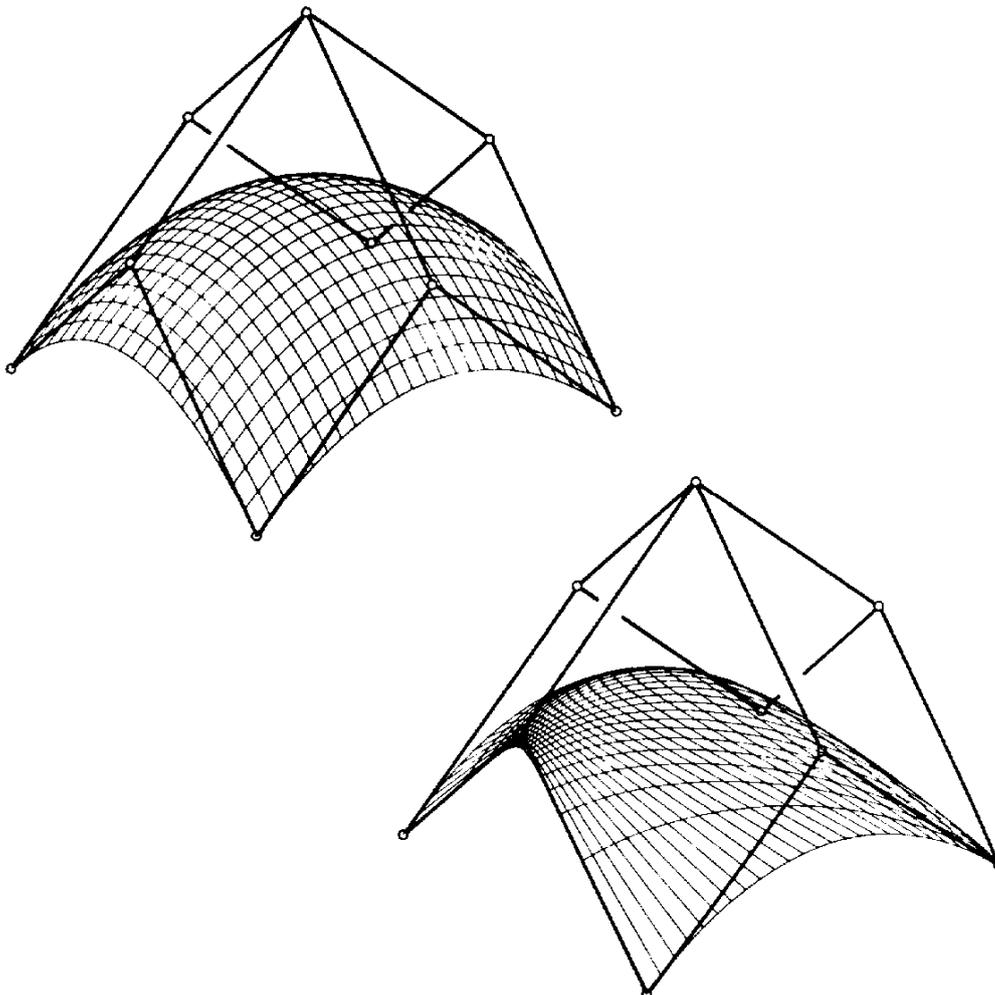
Rationale B-Spline-Flächen (NURBS-Patches) und rationale Bézier-Flächen

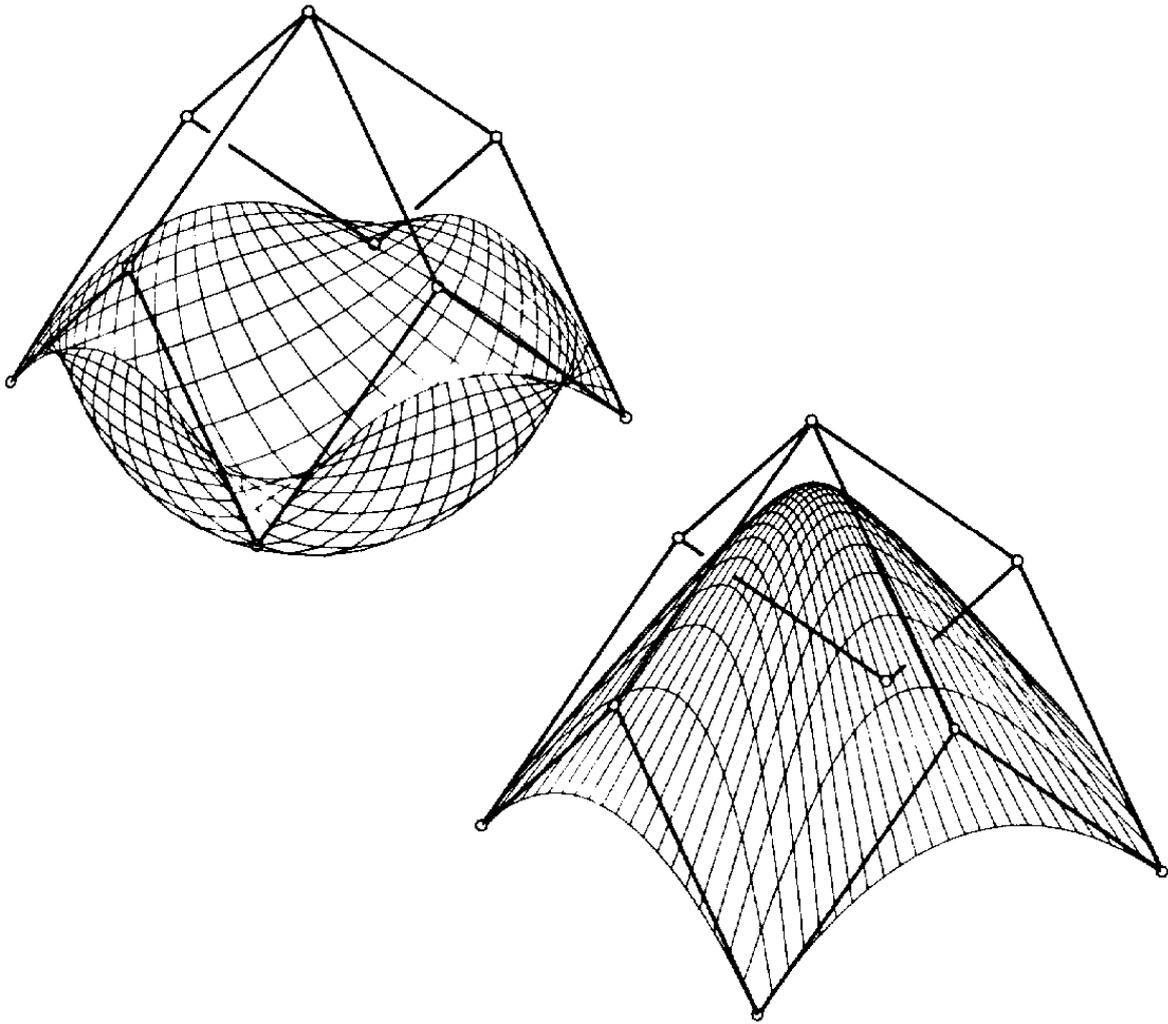
wie im Fall der Kurven: Hinzunahme einer weiteren Funktion in der homogenisierenden, vierten Koordinate, die als "Gewichtsfunktion" weitere Freiheitsgrade eröffnet

Beispiel: rationale Bézier-Flächen mit gleichem Kontrollpunkte-Netz, aber unterschiedlichen Gewichtsfunktionen (aus Hoschek & Lasser 1992):

(1. Bild: alle Gewichte = 1, 2. Bild: linker Rand mit Gewicht 10, rechter Rand mit Gewicht 0,1)

(Fortsetzung nächste Seite)





(oben links: mittlerer Punkt mit Gewicht $-1,5$,
unten rechts: mittl. Punkt mit Gewicht 10 .)