

Institute of Geometry

Institut für Geometrie, TU Graz  
[www.geometrie.tugraz.at](http://www.geometrie.tugraz.at)



# Freiformkurven

**Anton Gfrerrer**

Institut für Geometrie, TU Graz  
e-mail: [gfrerrer@tugraz.at](mailto:gfrerrer@tugraz.at)



# 1 Bézierkurven

Es sei eine Folge von Punkten  $\vec{p}_0, \dots, \vec{p}_n$  sei gegeben. Ziel ist es, eine Kurve  $\vec{x}(t)$  zu finden, deren Verlauf durch die gegebene Punktfolge  $\vec{p}_0, \dots, \vec{p}_n$  gut vorhergesagt werden kann. Dabei soll das Kurvenstück in einer gewissen Nähe dieser **Basispunkte** bleiben.

Ein erster Ansatz dieser Art stammt aus den Jahren 1950 - 1960: Die beiden Franzosen DeCASTELJAU und P. BÉZIER haben zur Karosseriemodellierung eine spezielle Kurvenklasse verwendet, die heute unter dem Namen Bézierkurven(stücke) bekannt ist. Dabei werden als Gewichtsfunktionen zu den Basispunkte  $\vec{p}_0, \dots, \vec{p}_n$ , welche auch Steuer- oder Kontrollpunkte genannt werden, die **Bernsteinpolynome  $n$ -ten Grades**

$$B_i^n(t) := \binom{n}{i} \cdot (1-t)^{n-i} \cdot t^i \quad (1.1)$$

verwendet. Sie wurden um 1910 vom russischen Mathematiker BERNSTEIN für einen eleganten Beweis des Weierstraßschen Approximationssatzes verwendet.

**Definition 1.1.** Die zu den Basispunkten  $\vec{p}_0, \dots, \vec{p}_n$  vermöge

$$\vec{x}(t) := \sum_{i=0}^n B_i^n(t) \vec{p}_i$$

mit Parameter  $t \in [0, 1]$  definierte (ganzrationale) Kurve  $n$ -ter Ordnung wird (ganzrationale oder integrale) **Bézierkurve** ( $n$ -ter Ordnung) zu den Basispunkten  $\vec{p}_0, \dots, \vec{p}_n$  genannt.

**Eigenschaften:**

**A) Bernsteinpolynome:**

- 1.) Der Grad der Bernsteinpolynome ist  $n$ .
- 2.) Nach dem binomischen Lehrsatz gilt:

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

Damit erfüllen die Bernsteinpolynome  $n$ -ten Grades die sogenannte **Teilung der 1**.

- 3.)  $t = 0 : B_i^n(0) = \begin{cases} 0 & \text{für } i \neq 0 \\ 1 & \text{für } i = 0 \end{cases}$
- 4.)  $t = 1 : B_i^n(1) = \begin{cases} 0 & \text{für } i \neq n \\ 1 & \text{für } i = n \end{cases}$
- 5.) Es ist  $B_i^n(1-t) = \binom{n}{i} [1-(1-t)]^{n-i} (1-t)^i = \binom{n}{n-i} t^{n-i} (1-t)^i = B_{n-i}^n(t) \quad \forall t \in \mathbb{R}, i = 0, \dots, n$ .  
Damit herrscht eine gewisse Symmetrie zwischen den Intervallgrenzen  $t = 0$  und  $t = 1$ .

**B) Bézierkurven:**

- 1.) Bézierkurven sind mit den Basispunkten affin invariant verbunden (Beweis siehe unten).
- 2.)  $t = 0$  : Auswertung der Bernsteinpolynome an der Stelle  $t = 0$  liefert  $\vec{x}(0) = \vec{p}_0$ .
- 3.)  $t = 1$  : Analog gilt  $\vec{x}(1) = \vec{p}_n$ .
- 4.) Die Koordinatenfunktionen der Parameterdarstellung unserer Bézierkurven sind Polynome  $n$ -ter Ordnung - die Bézierkurvenstücke  $n$ -ter Ordnung sind daher im geometrischen Sinne integrale (ganzrationale) Kurven der Ordnung  $n$ . Technisch bedeutsam sind vor allem die Fälle  $n = 2, 3, 4$ .

- 5.) Wir betrachten neben der Bézierkurve  $\vec{x}(t)$  zu den Basispunkten  $\vec{p}_0, \dots, \vec{p}_n$  auch die mit den Basispunkten  $\vec{p}_0^* = \vec{p}_n, \dots, \vec{p}_i^* = \vec{p}_{n-i}, \dots, \vec{p}_n^* = \vec{p}_0$  (vertauschte Reihenfolge) - wir werden sie mit  $\vec{x}^*(t)$  bezeichnen: Dann gilt nach **A**, 5):

$$\vec{x}^*(1-t) = \sum_{i=0}^n B_i^n(1-t) \cdot \vec{p}_i^* = \sum_{i=0}^n B_{n-i}^n(t) \cdot \vec{p}_{n-i} = \vec{x}(t).$$

Damit ändert das Vertauschen des Durchlaufsinnes der Basispunkte bloß den Durchlaufsin der zugehörigen Bézierkurve, nicht jedoch ihr geometrisches Aussehen.

- 6.) Durch Differentiation erhält man: Der  $j$ -te Ableitungsvektor der Kurve an der Stelle  $t = 0$  bzw.  $t = 1$  hängt nur von den Basispunkten  $\vec{p}_0, \dots, \vec{p}_j$  bzw.  $\vec{p}_{n-j}, \dots, \vec{p}_n$  ab. Damit ergibt sich z.B., dass die erste (letzte) Seite des Basispunktepolygons die Bézierkurve im Punkt  $\vec{p}_0$  ( $\vec{p}_n$ ) berührt.

## 1.1 Der Algorithmus von DeCasteljau

Wir wollen für das folgende die Bézierkurve zu den Basispunkten  $\vec{p}_0, \dots, \vec{p}_n$  mit  $\vec{x}_{0,\dots,n}(t)$  bezeichnen. Dann gilt:

$$\vec{x}_{0,\dots,n}(t) = (1-t) \cdot \vec{x}_{0,\dots,n-1}(t) + t \cdot \vec{x}_{1,\dots,n}(t) \quad (1.2)$$

D.h., dass sich die Bézierkurve zu den Basispunkten  $\vec{p}_0, \dots, \vec{p}_n$  durch **lineares Blending** aus den beiden Bézierkurven zu den Basispunkten  $\vec{p}_0, \dots, \vec{p}_{n-1}$  bzw.  $\vec{p}_1, \dots, \vec{p}_n$  gewinnen lässt. Der Beweis von (1.2) folgt durch direktes Nachrechnen unter Berücksichtigung der Identität

$$\forall i = 1, \dots, n-1: \binom{n}{i} = \binom{n-1}{i-1} + \binom{n-1}{i} \quad (1.3)$$

Durch sukzessives Anwenden kann man so die Bézierkurve  $\vec{x}_{0,\dots,n}(t)$  auf die Bézierkurven  $\vec{x}_{0,1}(t), \dots, \vec{x}_{n-1,n}(t)$  zurückspielen. Im Fall  $n = 3$  sieht das etwa so aus:

$$\begin{aligned} \vec{x}_{0,1,2,3}(t) &= (1-t) \cdot \vec{x}_{0,1,2}(t) + t \cdot \vec{x}_{1,2,3}(t) \\ &= (1-t) \cdot [(1-t) \cdot \vec{x}_{0,1}(t) + t \cdot \vec{x}_{1,2}(t)] + t \cdot [(1-t) \cdot \vec{x}_{1,2}(t) + t \cdot \vec{x}_{2,3}(t)] \end{aligned}$$

Rollt man diese Vorgangsweise von hinten auf, so ergibt sich der

### Algorithmus 1.1. DeCasteljau.

**Geg.:** Basispunkte  $\vec{p}_0, \dots, \vec{p}_n$

**Ges.:** Kurvenpunkt  $\vec{x}(t)$  der zugehörigen integralen Bézierkurve  $n$ -ter Ordnung.

1) **FOR**  $i := 0$  **TO**  $n$  **DO**  $\vec{q}_{i,0}(t) := \vec{p}_i$ .

2) **FOR**  $k := 1$  **TO**  $n$  **DO**

**FOR**  $i := 0$  **TO**  $n-k$  **DO**

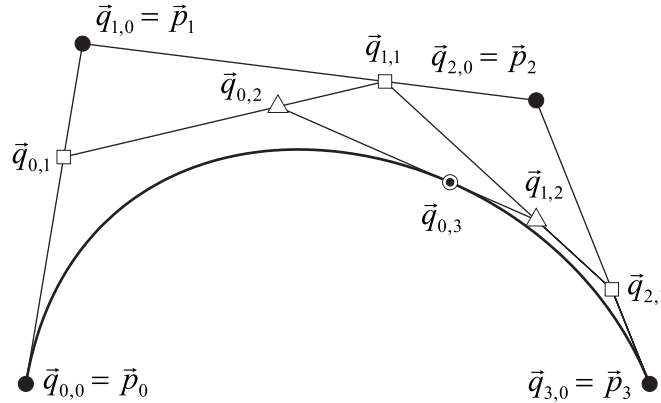
$\vec{q}_{i,k}(t) := (1-t) \cdot \vec{q}_{i,k-1}(t) + t \cdot \vec{q}_{i+1,k-1}(t)$ .

3)  $\vec{x}(t) = \vec{q}_{0,n}(t)$ .

Aus geometrischer Sicht besteht der Algorithmus aus fortgesetztem Bestimmen von Teilungspunkten von Strecken mit konstantem Teilverhältnis. Das Teilen von Strecken ist affin invariant. Damit haben wir nachgewiesen, dass **ganzrationale (integrale) Bézierkurven affin invariant mit den Basispunkten verbunden sind**.

Der DeCasteljau-Algorithmus ist zudem numerisch wesentlich stabiler als die Berechnung von Kurvenpunkten mit Hilfe der Bernsteinpolynome.

Figur 1.1 zeigt als Beispiel den Ablauf des Algorithmus im Fall  $n = 3$ .



Figur 1.1. Algorithmus von DeCasteljau.

## 2 Interpolation von Punkten mittels ganzrationaler Kurven

Ein altes Problem ist die Bestimmung eines Kurvenstückes durch eine gegebene Serie von Punkten  $\vec{p}_0, \dots, \vec{p}_n$ , wobei diese Punkte zu den Parameterwerten  $u_0 < \dots < u_n$  gehören sollen. Die Lösung wird als *Interpolation der Punkte durch eine Kurve* angesprochen.

Es lässt sich nachweisen, dass diese Aufgabe i.a. genau von einer ganzrationalen (integralen) Kurve  $n$ -ter Ordnung gelöst wird.

Diese Lösung lässt sich mit Hilfe der sogenannten *LAGRANGEschen Polynome*

$$L_j^n(t) := \frac{\prod_{i \neq j}^{i=0, \dots, n} (t - u_i)}{\prod_{i \neq j}^{i=0, \dots, n} (u_j - u_i)} = \frac{(t - u_0) \cdot (t - u_1) \cdots (t - u_{j-1}) \cdot (t - u_{j+1}) \cdots (t - u_n)}{(u_j - u_0) \cdot (u_j - u_1) \cdots (u_j - u_{j-1}) \cdot (u_j - u_{j+1}) \cdots (u_j - u_n)} \quad (2.4)$$

$n$ -ten Grades angeben.

Es gilt:

$$1.) \quad L_j^n(u_i) = \begin{cases} 0 & \text{für } i \neq j \\ 1 & \text{für } i = j \end{cases}$$

Daher löst das durch die Parameterdarstellung

$$\vec{x}(t) = \sum_{i=0}^n L_i^n(t) \cdot \vec{p}_i \quad (t \in [u_0, u_n]) \quad (2.5)$$

beschriebene Kurvenstück unsere Interpolationsaufgabe.

2.)  $\sum_{i=0}^n L_i^n(t) = 1 \quad (\forall t \in \mathbb{R}):$

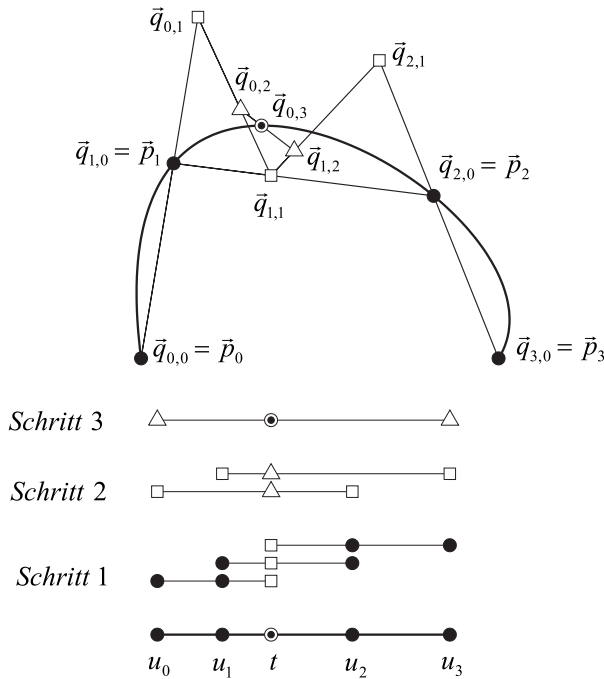
**Beweis:** Wir bilden  $f(t) := 1 - \sum_{i=0}^n L_i^n(t)$ . Es handelt sich dabei um ein Polynom  $n$ -ten Grades, das nach unseren Überlegungen sicher die  $n+1$  Nullstellen  $u_i$  besitzt. Wenn aber ein Polynom  $n$ -ten Grades  $n+1$  verschiedene Nullstellen besitzt, muß  $f(t) = 0 \quad \forall t \in \mathbb{R}$  gelten, womit obige Behauptung bewiesen ist. Somit erfüllen auch die Lagrange-Polynome die Teilung der 1.

**Definition 2.1.** Die zu den Basispunkten  $\vec{p}_0, \dots, \vec{p}_n$  und dem Knotenvektor  $U := (u_0 < u_1 < \dots < u_n)$  vermöge

$$\vec{x}(t) := \sum_{i=0}^n L_i^n(t) \vec{p}_i$$

mit Parameter  $t \in [u_0, u_n]$  definierte Kurve heißt **Lagrange-Interpolant** zu den Punkten  $\vec{p}_i$  und den Parameterstützstellen  $u_i$ .

Weitere Möglichkeiten, diese und verwandte Interpolationsaufgaben (mit gegebenen Tangenten usw.) zu lösen, sind etwa in [2] beschrieben.



**Figur 2.1.** Algorithmus von Aitken.

Für die Konstruktion von Punkten der Kurve (2.5) steht wie bei den Bézierkurven ein einfacher Algorithmus zur Verfügung:

**Algorithmus 2.1. Aitken.****Geg.:** Punkte  $\vec{p}_0, \dots, \vec{p}_n$  zu den Parameterwerten  $u_0 < \dots < u_n$  und ein Parameterwert  $t \in [u_0, u_n]$ **Ges.:** Kurvenpunkt  $\vec{x}(t)$  der interpolierenden integralen Kurve  $n$ -ter Ordnung.**0) FUNCTION**  $\alpha(i, k : \text{INTEGER}; t : \text{REAL})$ : **REAL**;  $\alpha := \frac{t - u_i}{u_{i+k} - u_i}$ **1) FOR**  $i := 0$  **TO**  $n$  **DO**  $\vec{q}_{i,0} := \vec{p}_i$ .**2) FOR**  $k := 1$  **TO**  $n$  **DO****FOR**  $i := 0$  **TO**  $n - k$  **DO** $\vec{q}_{i,k} := (1 - \alpha(i, k, t)) \cdot \vec{q}_{i,k-1} + \alpha(i, k, t) \cdot \vec{q}_{i+1,k-1}$ .**3)  $\vec{x}(t) = \vec{q}_{0,n}$ .**

Die Herleitung dieses Algorithmus ergibt sich wieder aus der Tatsache, dass sich der Lagrange-Interpolant  $\vec{x}_{0,\dots,n}(t)$  zu den Punkten  $\vec{p}_0, \dots, \vec{p}_n$  durch **lineares Blending** aus den beiden Lagrange-Interpolanten  $\vec{x}_{0,\dots,n-1}(t)$  bzw.  $\vec{x}_{1,\dots,n}(t)$  zu den Punkten  $\vec{p}_0, \dots, \vec{p}_{n-1}$  bzw.  $\vec{p}_1, \dots, \vec{p}_n$  gewinnen lässt.

Figur 2.1 veranschaulicht den Algorithmus für den Fall  $n = 3$ .

**Bemerkung 2.1.**

- 1.) Die Lösungskurve hat den Grad  $n$ . Raumkurven höherer als 4. Ordnung neigen zum Oszillieren, weshalb Lagrange-Interpolanten für großes  $n$  unbefriedigende Ergebnisse liefern.
- 2.) Folgende Möglichkeiten der sinnvollen Verteilung der Parameterstützstellen  $u_i$  werden verwendet (Figur 2.2):

- *uniforme Parameterverteilung*  $u_i := i \quad (i = 0, \dots, n)$
- *chordale Parametrisierung (berücksichtigt Abstände der Stützpunkte)*  
 $u_0, u_1$  beliebig vorgeben, dann  $u_i \ (i = 2, \dots, n)$  aus der Beziehung

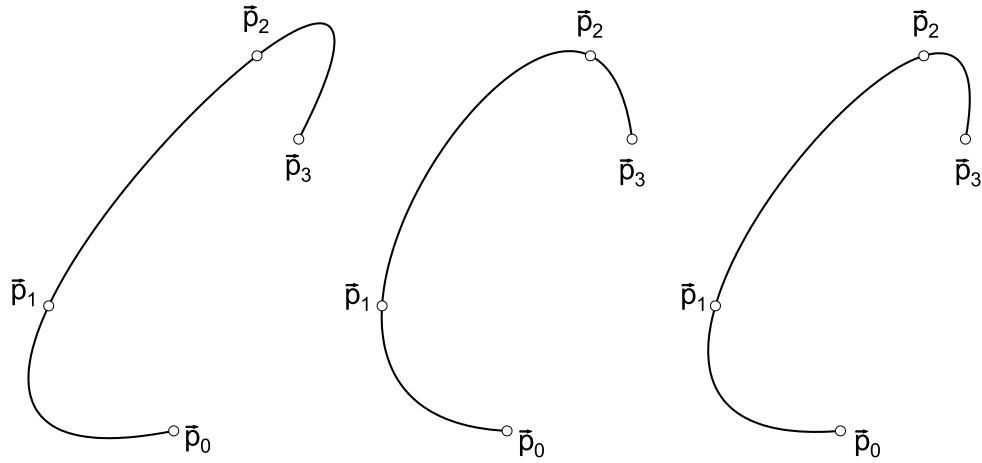
$$\frac{u_i - u_{i-1}}{u_{i-1} - u_{i-2}} = \frac{|\vec{p}_i - \vec{p}_{i-1}|}{|\vec{p}_{i-1} - \vec{p}_{i-2}|}$$

berechnen.

- *zentripetale Parametrisierung (Versuch, die Längenveränderungen auszugleichen):* Vorgehen wie vorhin; nur dass  $u_i \ (i = 2, \dots, n)$  aus

$$\frac{u_i - u_{i-1}}{u_{i-1} - u_{i-2}} = \sqrt{\frac{|\vec{p}_i - \vec{p}_{i-1}|}{|\vec{p}_{i-1} - \vec{p}_{i-2}|}}$$

zu berechnen sind.



**Figur 2.2.** Auswirkung der verschiedenen Parameterverteilungen.

- 3.) Beim beschriebenen AITKEN-Algorithmus werden sukzessive Strecken geteilt. Dieses Teilen ist invariant gegenüber affinen Abbildungen (in dieser Gruppe von Abbildungen sind auch die euklidischen Bewegungen enthalten). Daher gilt: Werden die zu interpolierenden Punkte  $\vec{p}_0, \dots, \vec{p}_n$  einer affinen Abbildung unterworfen und für Ausgangs- und Bildpunkte (samt alter Parameterbelegung) die interpolierenden ganzrationalen Kurven  $n$ -ter Ordnung ermittelt, so entsprechen sich die beiden Kurven in derselben affinen Abbildung wie die Punktmengen. Man sagt auch, die so gefundene interpolierende ganzrationale Kurve  $n$ -ter Ordnung ist affin invariant mit den Ausgangspunkten (mit Parameterbelegung) verbunden.

### 3 B-Splines

Die bisher vorgestellten Freiformkurven (Bézierkurven und Lagrange-Interpolanten) weisen zwei gravierende Nachteile auf:

- Die Änderung eines der Kontrollpunkte  $\vec{p}_i$  bewirkt eine Änderung des gesamten Kurvenverlaufes.
- Die Ordnung  $n$  der Kurve steigt mit der Anzahl der Kontrollpunkte, was bei großer Anzahl zu Oszillationsverhalten führt.

Eine naheliegende Idee ist daher, Approximations- oder Interpolationskurven aus Segmenten niederer algebraischer Ordnung zusammenzusetzen, die an den Trennstellen mit vorgegebener Stetigkeitsordnung ineinander übergehen. Die am häufigsten verwendeten Gewichtsfunktionen, die dieser Forderung genügen, sind die **B-Spline Basisfunktionen**  $N_{i,k}(t)$ . Diese werden wie folgt rekursiv über einem gegebenen Knotenvektor  $U := (u_0 < u_1 < \dots < u_{n+k})$  definiert:

$$\begin{aligned}
 N_{i,1}(t) &:= \begin{cases} 1 & \text{wenn } t \in [u_i, u_{i+1}) \\ 0 & \text{sonst} \end{cases} \\
 N_{i,k}(t) &:= \frac{t - u_i}{u_{i+k-1} - u_i} N_{i,k-1}(t) + \frac{u_{i+k} - t}{u_{i+k} - u_{i+1}} N_{i+1,k-1}(t)
 \end{aligned} \tag{3.6}$$

für  $i = 0, \dots, n$ .

### Eigenschaften der B-Spline-Basispolynome:

- 1.) Die B-Spline Basisfunktionen  $N_{i,k}(t)$  sind vom Grad  $k - 1$ .
- 2.) Ebenso wie die Lagrange-Interpolanten oder die Bernstein-Polynome erfüllen auch die B-Spline-Basisfunktionen die Teilung der Eins:

$$\sum_{i=0}^n N_{i,k}(t) \stackrel{!}{=} 1 \quad (3.7)$$

**Definition 3.1.** Die zu den Basispunkten  $\vec{p}_0, \dots, \vec{p}_n$  und dem Knotenvektor  $U := (u_0 < u_1 < \dots < u_{n+k})$  vermöge

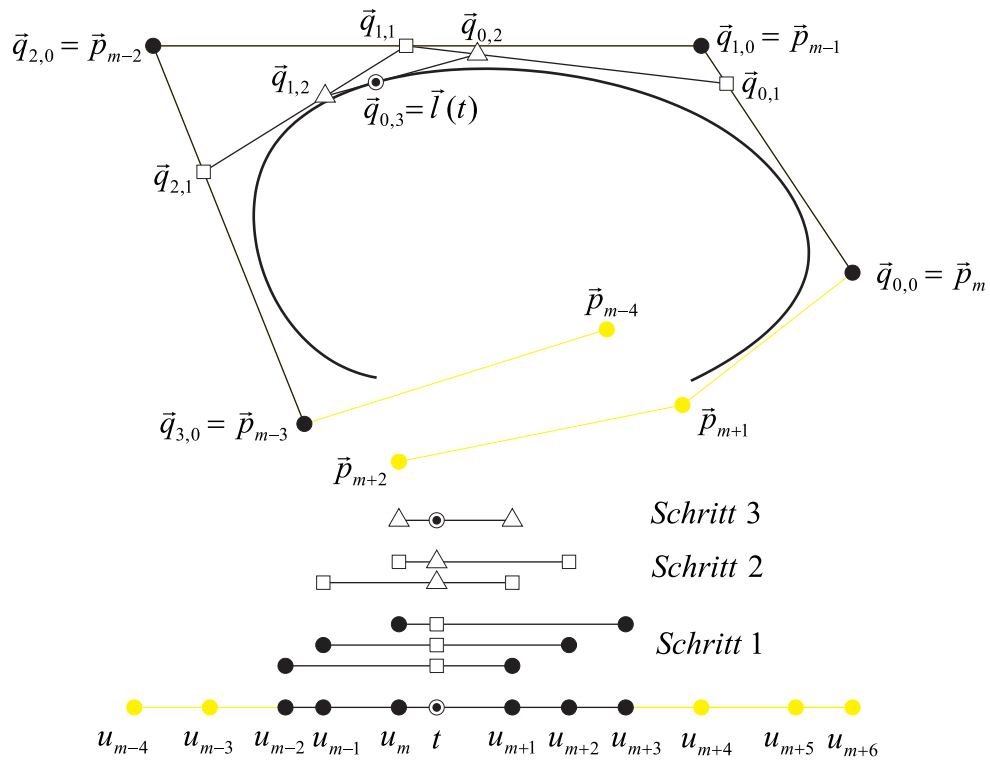
$$\vec{x}(t) := \sum_{i=0}^n N_{i,k}(t) \vec{p}_i$$

definierte Kurve mit Parameter  $t \in [u_{k-1}, u_{n+1}]$  heißt B-Spline-Kurve mit den **Kontrollpunkten**  $\vec{p}_i$  und dem **Knotenvektor**  $U$ .

### Eigenschaften von B-Spline-Kurven:

- 1.) B-Spline-Kurven sind stückweise polynomial vom Grad  $k - 1$ .
- 2.) An den Segmenttrennstellen  $u_i$  schließen die beiden dort zusammentreffenden Kurvensegmente  $C^{k-2}$ -stetig aneinander.
- 3.) Das über dem Intervall  $[u_m, u_{m+1}]$  definierte Kurvenstück wird nur von den  $k$  Kontrollpunkten  $p_{m-k+1}, \dots, p_m$  beeinflusst (**lokale Kontrolle!**).

Ähnlich wie bei den Bézierkurven oder Lagrange-Interpolanten gibt es auch hier einen Teilungsalgorithmus zur Erzeugung dieser Kurven. Er heißt **Algorithmus von Cox-DeBoor** und ist in Figur 3.1 für den Fall  $k = 4$  dargestellt.



**Figur 3.1.** Der Cox-DeBoor Algorithmus.

## Literatur

- [1] Farin, G. (1990), *Curves and surfaces for computer aided design*, 2nd edition, Acad. Press, San Diego.
- [2] Hoschek, J.-Lasser, D. (1992): *Grundlagen der geometrischen Datenverarbeitung*, 2. Auflage, B.G. Teubner Stuttgart.