

# **Freiformkurven und -flächen in CAD-Systemen**

**H. Noack  
Fachhochschule Hamburg  
Fachbereich Maschinenbau und Produktion**

## INHALT

1 Einleitung .....	2
2 Grundlagen der Beschreibung von Freiformkurven und -flächen (sculptured curves and surfaces).....	3
2.1 Wie kommt der Konstrukteur zu Freiformkurven ? .....	3
2.2 Approximation .....	5
2.3 Interpolation.....	8
2.3.1 Splines .....	9
2.3.2 Parametrische Splines .....	15
2.3.3 Periodische Splines .....	16
2.4 Darstellung von Kurven mit Hilfe verschiedener polynomialer Basen .....	18
2.4.1 Einführung: Ferguson-Basis und Hermitesche Basis.....	18
2.4.2 Kurvendarstellung nach Bézier.....	20
2.4.3 Kurvendarstellung mit Hilfe von Basis Splines (B-Splines).....	25
2.4.4 Non Uniform Rational B-Splines .....	31
2.5 Flächendarstellungen .....	33
2.5.1 Flächendarstellung nach Ferguson.....	33
2.5.2 Bézier-Patches .....	34
2.5.3 B-Spline Flächen.....	35
2.5.4 Coons-Flächen.....	36

## 1 Einleitung

Oft wird behauptet, daß ein Konstrukteur im Maschinenbau ohne die sogenannten Freiformkurven und -flächen auskommt. In diesem Falle wäre er bei der grafischen Darstellung seiner Konstruktionen auf Formen beschränkt, die sich geschlossen beschreiben lassen:

Geschlossen beschreibbare **Kurven**:

**Kreis, Ellipse, Parabel, Hyperbel, also Kegelschnitte und Geraden**

Geschlossen beschreibbare **Flächen**:

**Kugel, Ellipsoid, Hyperboloid, Kegel, Paraboloid, Zylinder**

Die verschiedenen Arten der mathematischen Darstellung dieser Objekte (z.B. in kartesischen oder in Zylinderkoordinaten) dürften bekannt sein. Besonders einfach wird die Darstellung, wenn sie auf **Normalform** gebracht wird, d.h. daß die Koordinatenachsen mit den Objektachsen zusammenfallen. Dies ist immer durch geeignete Koordinatentransformationen erreichbar. Oft findet man in den Manuals von CAD-Systemen dafür auch den Begriff **kanonische Form**:

Ellipse: 
$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (\text{wird zum Kreis für } a=b)$$

Ellipsoid: 
$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \quad (\text{wird zur Kugel für } a=b=c)$$

a, b und c sind die jeweiligen Abschnitte auf den Hauptachsen der Objekte.

Denkt man jedoch nicht nur an den sogenannten allgemeinen Maschinenbau, sondern auch an die vielen etwas spezielleren Branchen wie:

**Karosseriebau, Flugzeugbau, Turbinenbau, Schiffbau (Außenhaut, Propeller), Gießerei, Gesenkschmieden und letztlich auch an modernes Design, das in vielen Bereichen gefragt ist (siehe Colani),**

so merkt man, daß der Bedarf an Methoden der Darstellung und Handhabung von Freiformkurven und -flächen doch erheblich ist. In allen genannten Bereichen arbeitet keineswegs nur speziell ausgebildetes Personal, sondern sehr oft „ganz normale“ Maschinenbauer.

## 2 Grundlagen der Beschreibung von Freiformkurven und -flächen (sculptured curves and surfaces)

### 2.1 Wie kommt der Konstrukteur zu Freiformkurven ?

Ausgangspunkt der Generierung einer mathematischen Beschreibung von Freiformkurven und/oder -flächen sind in den meisten Fällen Punkte bzw. Punktmengen, die man durch Meß- oder Abtastgeräte aufgenommen hat. Betrachtet man als Beispiel das Karosseriedesign, so liegen nach dem Abtasten eines Stylingmodells (aus Holz oder Kunststoff) große Punktmengen vor. Als Fertigungsunterlage für den Werkzeugbau ist dies nicht direkt verwendbar. Der Konstrukteur muß sich aus den Punktmengen Flächen generieren. Allerdings gibt es bis heute kein CAD-System, daß die gewünschten Flächen direkt aus diesen Punktmengen generieren kann. Neue Funktionen in CAD-Systemen (z.B. CLOUD TO GEOMETRY in CATIA) arbeiten nur dann korrekt, wenn die Punkteverteilung bestimmte Regelmäßigkeiten aufweist und die zu erwartende Flächenform von vier Kanten berandet ist.

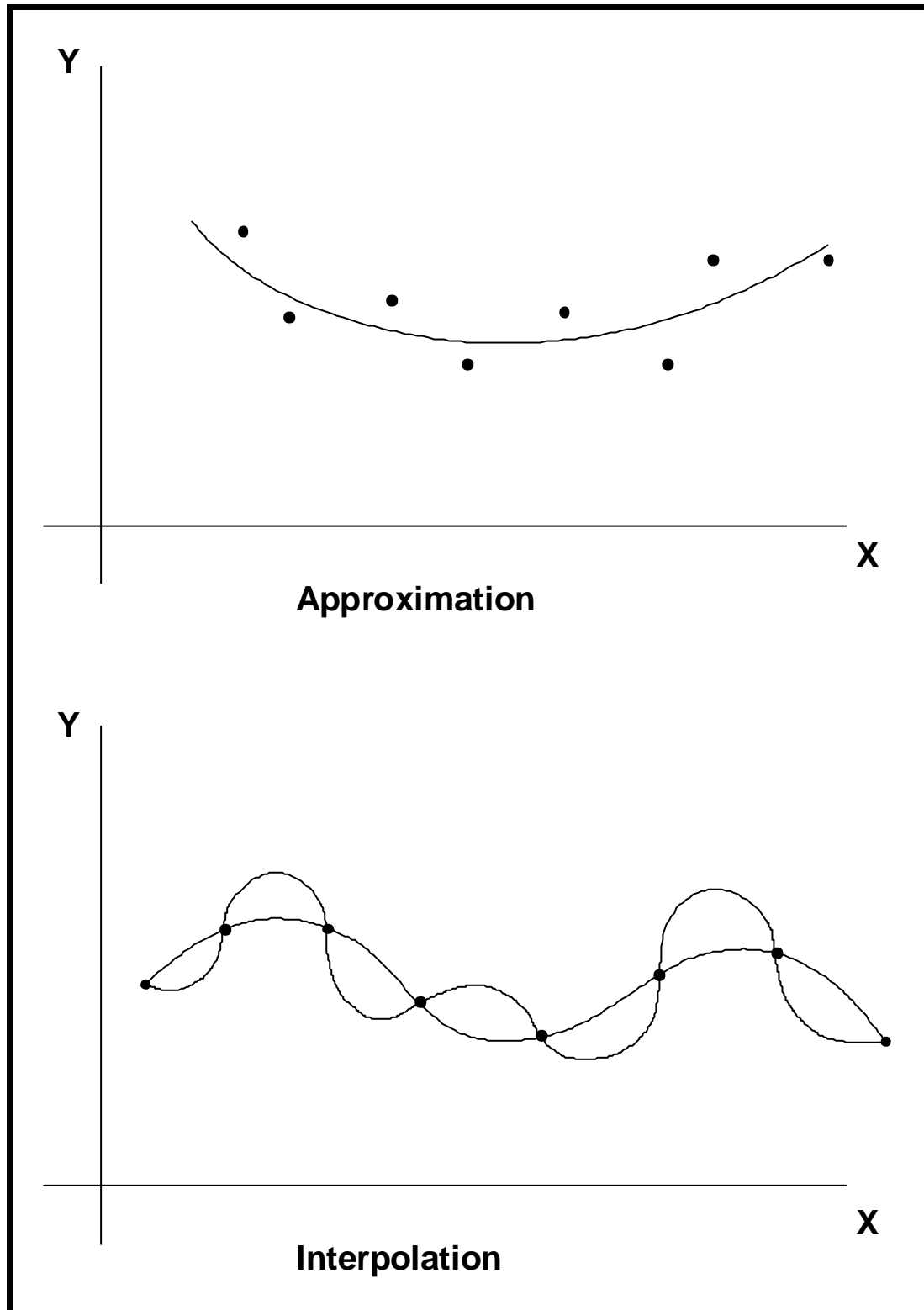
Nach den heute verwendeten praktikablen Algorithmen führt der Weg von den Punkten zur Fläche immer zunächst über Kurven und zwar in der Regel über ebene Kurven. Die mathematischen Methoden, die zur Beschreibung der Kurven dienen, werden mit entsprechenden Erweiterungen auch zur Beschreibung der Flächen verwendet. Darum gilt alles, was im folgenden zu den Kurven gesagt wird, in analoger Weise auch für Flächen.

Grundsätzlich gibt es zwei Arten um von Punkten zu Kurven zu gelangen. Man unterscheidet:

### **Approximation** und **Interpolation**

Welches von beiden richtig ist, hängt von der Bedeutung ab, die den Punkten zukommt. Erhält man eine Folge von Punkten durch Messungen, so weiß man, daß sie mit Fehlern behaftet sind. Eine Kurve, die durch alle Punkte hindurch gelegt würde, würde demnach auch alle Fehler repräsentieren. Insofern entspräche eine ausgleichende Kurve, die die Punkte nicht berührt, sondern nur annähert, vielleicht eher dem gewünschten Ergebnis. In solch einem Fall spricht man von **Approximation**, weil die Punkte nur angenähert (approximiert) werden. Ein sehr bekanntes Beispiel ist der Ausgleich von Messwerten durch eine Gerade nach der Methode der kleinsten Fehlerquadrate. Der Ausgleich in Form einer Geraden wird natürlich vor allem dann gewählt, wenn theoretische Überlegungen einen linearen Zusammenhang nahelegen.

Erhält man dagegen eine Folge von Punkten durch mathematisch exakte Bestimmung, so weiß man, daß eine verbindende Kurve genau durch alle Punkte hindurch verlaufen muß. In diesem Fall spricht man von **Interpolation**.



**Abb. 1: Unterschied zwischen Approximation und Interpolation**

Sowohl zur Approximation als auch zur Interpolation sind Polynome die am häufigsten verwendeten Funktionen. Daher zunächst zur Wiederholung einige Begriffsklärungen, nämlich Grad, Ordnung und Dimension eines Polynoms:

Ein **Polynom** ist eine Funktion der Form  $Y=a_0+a_1x+a_2x^2+a_3x^3+a_4x^4+\dots+a_nx^n$

**Grad:** Der Grad eines Polynoms ist gleich dem höchsten vorkommenden Exponenten (n) der Variablen des Polynoms (x).

**Dimension:** Die Dimension ist gleich Grad+1. Sie bezeichnet die Anzahl der Koeffizienten des Polynoms (Achtung: Nicht vorhandene Koeffizienten haben lediglich den Wert Null)

**Ordnung:** Der Grad des Polynoms ist die Ordnung der Kurve. Beides wird daher oft als Synonym verwendet. Viele Autoren benutzen den Begriff Ordnung aber auch als Grad+1, also im Sinne der Dimension. Wegen des uneinheitlichen Gebrauchs besser nicht verwenden.

## 2.2 Approximation

Wir behandeln hier zunächst die Approximation, da wir dazu an eine in der Regel bekannte Aufgabenstellung anknüpfen können, nämlich den Ausgleich von Punkten durch eine Gerade. Dabei waren die beiden Koeffizienten **m** und **b** in der Geradengleichung  $y(x)=m \cdot x+b$  so zu bestimmen, daß die Summe der quadratischen Abweichungen der Meßpunkte von der Geraden minimal wird.

Möchte man als Ausgleichsfunktion keine Gerade, sondern z.B. eine Parabel erhalten, so sind die Koeffizienten  $a_0$ ,  $a_1$  und  $a_2$  der Kurvengleichung  $y(x)=a_0+a_1x+a_2x^2$  (Parabel) so zu bestimmen, daß die Summe ihrer quadratischen Abweichungen von den Punkten minimal wird. Folglich fordern wir:

$$F = \sum_{i=1}^n [y(x_i) - y_i]^2 = \sum_{i=1}^n [a_0 + a_1x_i + a_2x_i^2 - y_i]^2 \stackrel{!}{=} \min$$

Damit F minimal wird, müssen die drei partiellen Ableitungen von F nach den gesuchten Koeffizienten  $a_0$ ,  $a_1$  und  $a_2$  verschwinden. Wir fordern daher:

$$\frac{\partial F}{\partial a_0} = 2 \sum_{i=1}^n [a_0 + a_1x_i + a_2x_i^2 - y_i] = 0$$

$$\frac{\partial F}{\partial a_1} = 2 \sum_{i=1}^n [a_0 + a_1x_i + a_2x_i^2 - y_i] x_i = 0$$

$$\frac{\partial F}{\partial a_2} = 2 \sum_{i=1}^n [a_0 + a_1x_i + a_2x_i^2 - y_i] x_i^2 = 0$$

Dies ist ein System aus drei Gleichungen für die unbekannten Koeffizienten  $a_0$ ,  $a_1$  und  $a_2$ , das sich natürlich auch in Matrixform schreiben läßt:

$$\begin{bmatrix} n & \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^4 \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n x_i^2 y_i \end{bmatrix}$$

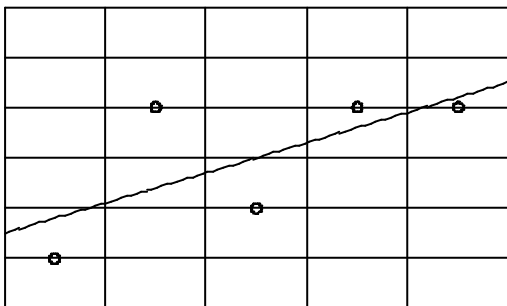
Eine andere Anordnung der Koeffizienten des Gleichungssystems erhält man, wenn man die Reihenfolge der gesuchten Koeffizienten  $a_0$ ,  $a_1$  und  $a_2$  umkehrt. Die Matrix wird dann an der Nebendiagonalen gespiegelt und der Vektor der rechten Seite wird umgekehrt.

Die hier für eine Parabel demonstrierte Vorgehensweise läßt sich natürlich auch für Hyperbeln oder Polynome noch höherer (theoretisch beliebiger) Ordnung durchführen. Entsprechend läßt sich das Ergebnis verallgemeinern bzw. schematisieren:

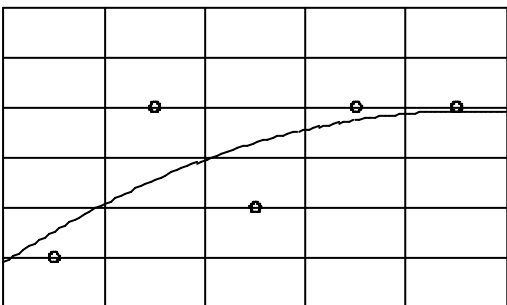
Zur Berechnung eines Ausgleichspolynoms vom Grad  $m$  zur Approximation von  $n$  Meßpunkten ergibt sich (es gilt immer:  $m+1=n$  !!!)

$$\begin{bmatrix} n & \sum_{i=1}^n x_i & \cdot & \cdot & \sum_{i=1}^n x_i^{m-1} & \sum_{i=1}^n x_i^m \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \cdot & \cdot & \cdot & \sum_{i=1}^n x_i^{m+1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \sum_{i=1}^n x_i^{m-1} & \cdot & \cdot & \cdot & \sum_{i=1}^n x_i^{2m-2} & \sum_{i=1}^n x_i^{2m-1} \\ \sum_{i=1}^n x_i^m & \sum_{i=1}^n x_i^{m+1} & \cdot & \cdot & \sum_{i=1}^n x_i^{2m-1} & \sum_{i=1}^n x_i^{2m} \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ \cdot \\ \cdot \\ a_{m-1} \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \cdot \\ \cdot \\ \sum_{i=1}^n x_i^{m-1} y_i \\ \sum_{i=1}^n x_i^m y_i \end{bmatrix}$$

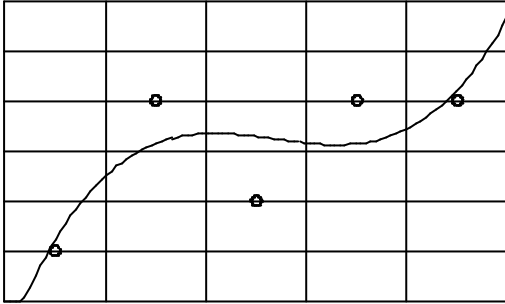
Die folgenden Bilder dokumentieren die Approximation von 5 „Meßpunkten“ durch Polynome vom Grad 1, 2 und 3:



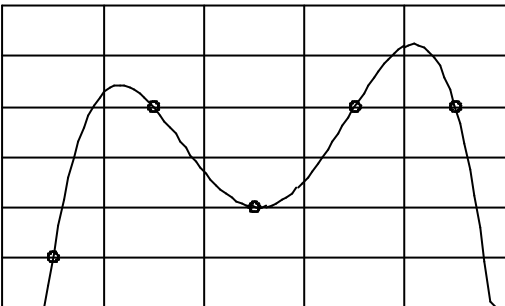
**Abb. 2: Approximation von 5 Punkten durch ein Polynom vom Grad 1**



**Abb. 3: Approximation von 5 Punkten durch ein Polynom vom Grad 2**



**Abb. 4: Approximation von 5 Punkten durch ein Polynom vom Grad 3**



**Abb. 5: "Approximation" (?) von 5 Punkten durch ein Polynom vom Grad 4  
--> Interpolation**

Nach der eingangs gegebenen Definition für Approximation und Interpolation haben wir es hier im letzten Bild mit einem Grenzfall zu tun. Ein Polynom vom Grad 4 hat 5 Koeffizienten, also die Dimension 5 (siehe Definition Grad, Ordnung und Dimension). Mit 5 vorgegebenen Punkten verfügt man gerade über 5 Bedingungen zur Berechnung der 5 unbekannten Koeffizienten. Damit ist klar, daß das Polynom die „Bedingungen“ exakt erfüllt, die Kurve also **durch** die Punkte verläuft. Hat man dagegen mehr Punkte als zu bestimmende Koeffizienten, so hat die Kurve „Spielraum“.

**Ein Programm, mit dem die oben geschilderten Vorgänge nachvollziehbar sind, ist bei mir im Quellcode erhältlich.**

Zur Approximation wurden hier willkürlich Polynome als Funktionen vorgegeben. Im CIM/CA-Umfeld nicht gebräuchlich aber möglich ist die Approximation durch andere Funktionen. Die Handhabung ist jedoch in der Regel weitaus schwieriger, weil das Gleichungssystem zur Bestimmung der unbekannten Koeffizienten meist nichtlinear ist (man weiß schon, warum man gern Polynome nimmt). Hier sei daher nur kurz auf den für den Ingenieur wichtigen Fall der Approximation durch Exponentialfunktionen hingewiesen. Verwendet wird diese Art der Approximation i.d.R. dann, wenn man (z.B. aus theoretischen Überlegungen) schon weiß, daß ein exponentieller Zusammenhang gegeben ist.

Man hilft sich hier dadurch, daß man das Problem durch Logarithmieren auf das Problem der Berechnung der Ausgleichsgeraden zurückführt:

$$y = c e^{rx} \rightarrow \ln y = \ln c + rx$$

Nun kürzt man ab:  $b = \ln c$  und  $z = \ln y$

Damit erhält das Problem die Form



$$z = rx + b$$

ist also reduziert auf die Berechnung der Koeffizienten einer Ausgleichsgeraden.

Für die Berechnung müssen die Werte  $y_i$  der Meßpunkte ersetzt werden durch  $z_i = \ln y_i$ .

Der Koeffizient  $r$  ist direkt Ergebnis der Berechnung, während  $c$  gemäß  $b = \ln c$  noch ausgerechnet werden muß:

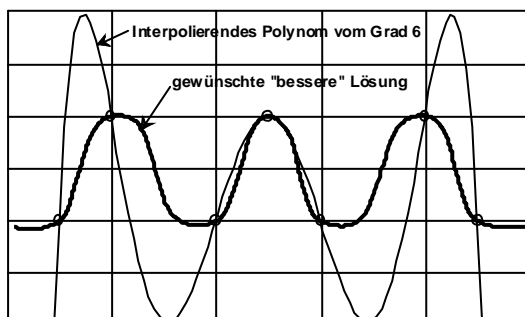
$$c = e^b.$$

## 2.3 Interpolation

Wie im Abschnitt über Approximation gezeigt wurde, ergibt sich die Interpolation von Punkten sozusagen als Sonderfall der Approximation. Man spricht in diesem Fall von der Polynominterpolation. Die Art und Weise, wie dabei verfahren wird, um zu den Koeffizienten zu gelangen (aufstellen und lösen eines linearen Gleichungssystems) ist recht aufwendig. Es gibt auch andere Verfahren, zu interpolierenden Polynomen zu kommen. Bekannt sind besonders das Lagrangesche und das Newtonsche Interpolationspolynom. In beiden Fällen müssen keine Gleichungssysteme gelöst werden, sondern man erhält die Interpolationspolynome durch Auswertung von Formeln, in die natürlich die Koordinaten der Stützpunkte (Meßpunkte) eingehen. Beide Verfahren haben ihre Eignungsschwerpunkte. Eines jedoch ist wichtig:

**Alle Polynomdarstellungen lassen sich durch algebraische Umformungen ineinander überführen.**

Allerdings ist die Polynominterpolation ein nur selten angewendetes Verfahren, da Polynome höheren Grades stark zu Oszillationen neigen, die sich besonders dann auswirken, wenn die Punkte nicht in einer glatten Folge liegen. Das liegt daran, daß ein Polynom vom Grad  $N$  genau  $N-1$  mal die Richtung wechselt, da es genau  $N$  Nullstellen hat. Die folgende Abbildung zeigt eine oszillierende Polynominterpolation für 7 Punkte, also Grad 6:

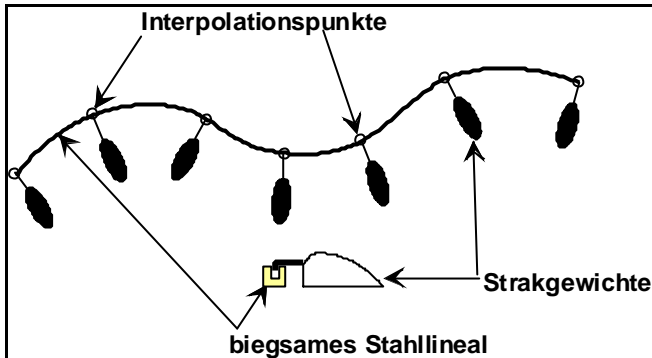


**Abb. 6: Oszillationen bei der Interpolation von 7 Punkten mit einem Polynom vom Grad 6 und ein interpolierender Spline als „bessere“ Lösung**

Das im Bild gezeigte Beispiel ist natürlich extrem. Es zeigt aber das Prinzip: Es ist mit Polynomen sehr schwierig, durch eine Folge von mehr als 7 Punkten eine nicht allzu wellige Kurve zu legen. Die als „bessere“ Lösung bezeichnete Kurve (dick) ist daher auch kein Polynom, sondern ein sogenannter Spline.

### 2.3.1 Splines

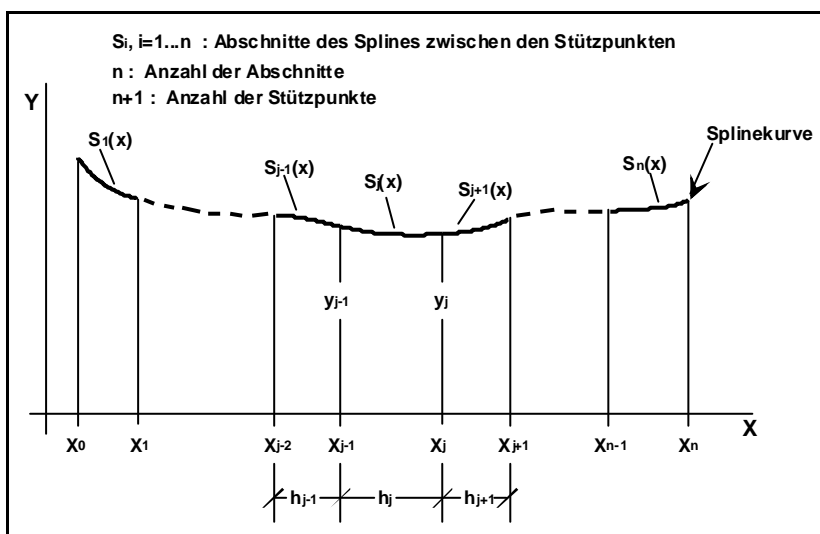
Manchem sind sicherlich biegsame Stahllineale bekannt, die auf der Oberkante eine Rinne haben, in die die sogenannten Strakgewichte eingehängt werden können, um das Lineal in der Stellung festzuhalten, in die man es biegt, um die zu interpolierenden Punkte zu treffen:



**Abb. 7: Interpolation von Punkten mit einem biegsamen Stahllineal (Spline)**

Ein solches Stahllineal heißt englisch SPLINE. Genauso heißen auch die mit einem solchen Lineal erzeugten Kurven. Allerdings bedient man sich heute nicht mehr des Lineals, sondern simuliert das Verfahren mit Hilfe moderner Rechentechnik.

Ein schöner weil für den Ingenieur sehr anschaulicher Überlegungsansatz, das Verhalten des Lineals mathematisch zu beschreiben, stammt von Prof. Dankert, der sich zur Erklärung der (bekannten) Regeln der technischen Mechanik bedient.



**Abb. 8: Eine Splinekurve unterteilt in ihre n Abschnitte, die durch n+1 Stützpunkte definiert sind.**

Da die Belastung des Lineals nur an den Stützstellen (Strakgewichte) eingeleitet wird, wird die Biegelinie des Lineals in jedem der n Bereiche durch eine Polynomfunktion dritten Grades beschrieben (siehe Integration der Gleichung der Biegelinie). Demnach sind also pro Kurvenabschnitt 4 Koeffizienten und für die gesamte Kurve 4n Koeffizienten zu berechnen, um die Kurve zeichnen zu können.

Hier integrieren wir allerdings nicht die Gleichung der Biegelinie, sondern formulieren lediglich die Stetigkeitsbedingungen an den Übergangsstellen, wie wir im folgenden sehen werden:

**Betrachtung der Randbedingungen der Kurvenabschnitte:**

- **Verschiebungsrandbedingung:**

Zwei aneinanderstoßende Kurvenabschnitte dürfen keine Klaffungen aufweisen. Sie müssen also gleiche Funktionswerte haben. Dies bringt  $2n$  Randbedingungen.

- **Tangentenstetigkeit:**

Der Übergang von einem Polynom zum anderen soll tangentialstetig erfolgen. Es wird also die Gleichheit der 1. Ableitungen an den Übergangsstellen gefordert. Dies bringt  $n-1$  Randbedingungen.

- **Krümmungsstetigkeit:**

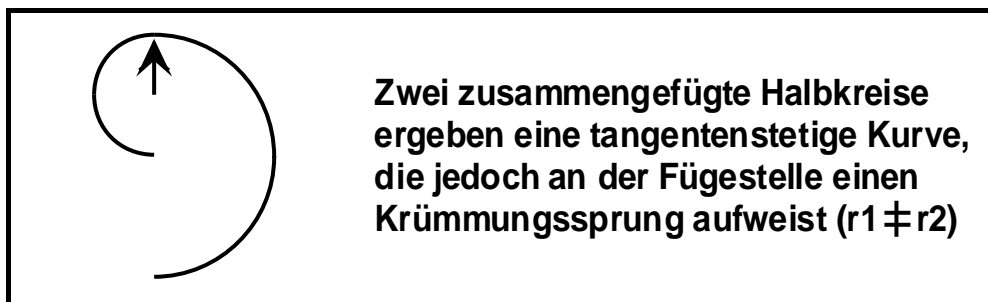
Der Übergang von einem Polynom zum anderen soll krümmungsstetig erfolgen. Es wird also die Gleichheit der 2. Ableitungen an den Übergangsstellen gefordert. Mechanisch bedeutet das die Forderung nach der Gleichheit des rechts- und linksseitigen Biegemomentes an den Übergangsstellen. Dies bringt  $n-1$  Randbedingungen.

- **Freie Ränder:**

An den freien Rändern muß das Biegemoment verschwinden. Das bedeutet, daß dort die 2. Ableitung gleich null ist. Dies bringt 2 Randbedingungen.

**Insgesamt erhalten wir also die benötigten  $4n$  Randbedingungen.**

Zu den so definierten Splines ist zu bemerken, daß es sich um Kurven mit hoher Stetigkeit (Krümmungsstetigkeit) handelt, wie es z.B. im Karosseriedesign gefordert wird. Man will dort möglichst glatte Kurven über große Strecken haben und keine Unstetigkeitsstellen, die vom Betrachter als störend empfunden werden. Designer gehen davon aus, daß bereits tangentialstetige Kurven und Flächen von den meisten Menschen als „glatt“ empfunden werden. Lediglich das geschulte Auge benötigt für das Attribut „glatt“ einen krümmungsstetigen Verlauf.



**Abb. 9: Beispiel für eine nicht krümmungsstetige aber tangentialstetige Kurve**

Unter Bezugnahme auf die Bezeichnungen in Abb. 8 wird im folgenden ein Algorithmus dargestellt, mit dessen Hilfe die  $4n$  Koeffizienten zur Darstellung der Kurve berechnet werden können:

Pro Kurvenabschnitt wird ein Polynomansatz folgender Form gemacht:

$$s_j(x) = y(x) = a_j + b_j(x - x_{j-1}) + c_j(x - x_{j-1})^2 + d_j(x - x_{j-1})^3$$

Der Index  $j$  bezeichnet den jeweiligen Kurvenabschnitt, es gilt also  $j=1, \dots, n$ .

Der Ansatz ist so gestaltet, daß die  $a_j$  für  $x=x_{j-1}$  den Funktionswert am linken Rand des Intervalls annehmen.

Zur Erfüllung der Randbedingungen benötigt man noch die 1. und die 2. Ableitung:

$$s'_j(x) = b_j + 2c_j(x - x_{j-1}) + 3d_j(x - x_{j-1})^2$$

$$s''_j(x) = 2c_j + 6d_j(x - x_{j-1})$$

Zur Bestimmung der  $n$  verschiedenen Koeffizienten  $a$ ,  $b$ ,  $c$  und  $d$  werden zunächst  $a_j$ ,  $b_j$  und  $d_j$  durch die bekannten Größen  $x$  und  $y$  sowie durch die unbekannten Koeffizienten  $c_j$  ausgedrückt. Dabei entsteht ein lineares Gleichungssystem für die  $c_j$ . Damit können dann die  $a_j$ ,  $b_j$  und  $d_j$  berechnet werden:

#### Bestimmung der $a_j$ :

Die Form des Ansatzes gewährleistet, daß die Funktion  $s_j$  am linken Rand den Funktionswert  $a_j$  annimmt. Also:

$$s_j(x_{j-1}) = y_{j-1} \quad \Rightarrow \quad \underline{\underline{a_j = y_{j-1}}}$$

#### Bestimmung der $d_j$ :

Formuliert man die geforderte Gleichheit der 2. Ableitungen für den rechten Rand des Abschnitts  $j$ , so erhält man aus  $s''_j(x_j) = s''_{j+1}(x_j)$  unter Verwendung der oben angegebenen Gleichungen für die 2. Ableitungen des Ansatzes:

$$2c_j + 6d_j(x - x_{j-1}) = 2c_{j+1} + 6d_{j+1}(x_j - x_{j-1}) \quad \nearrow = 0$$

$$\underline{\underline{\Rightarrow d_j = \frac{c_{j+1} - c_j}{3(x_j - x_{j-1})} = \frac{c_{j+1} - c_j}{3h_j}}}$$

dabei ist  $\underline{\underline{h_j = x_j - x_{j-1}}}$

#### Bestimmung der $b_j$ :

Am rechten Rand eines Abschnitts  $j$  (also an der Stelle  $x=x_j$ , siehe Abb. 8) muß die Funktion  $S_j$  den Funktionswert  $y_j$  annehmen, also  $s_j(x_j) = y_j$ . Daraus folgt, wenn man die bereits für  $a_j$ ,  $d_j$  und  $h_j$  ermittelten Beziehungen an der Stelle  $x=x_j$  in den Ansatz einsetzt:

$$y_j = y_{j-1} + b_j(x_j - x_{j-1}) + c_j(x_j - x_{j-1})^2 + \frac{c_{j+1} - c_j}{3h_j}(x_j - x_{j-1})^3$$

$$y_j = y_{j-1} + b_j h_j + c_j h_j^2 + \frac{c_{j+1} - c_j}{3} h_j^2 = y_{j-1} + b_j h_j + \frac{c_{j+1} + 2c_j}{3} h_j^2$$

$$\Rightarrow \frac{y_j - y_{j-1}}{h_j} = b_j + \frac{c_{j+1} + 2c_j}{3} h_j \quad \text{oder:}$$

$$\underline{\underline{b_j = \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{3}(2c_j + c_{j+1})}}$$

### Bestimmung der $c_j$ :

Hierzu wird die Bedingung für die Tangentenstetigkeit, also die Übereinstimmung der 1. Ableitungen an den Übergangsstellen verwendet. Formuliert für den linken Randpunkt, also für  $x=x_{j-1}$  erhält man:  $s'_{j-1}(x_{j-1}) = s'_j(x_{j-1})$ . Setzt man hier die abgeleitete Beziehung für die 1. Ableitung des Ansatzes und die erhaltenen Beziehungen für die  $b_j$ ,  $d_j$  und  $h_j$  ein, so erhält man:

$$s'_{j-1}(x_{j-1}) = s'_j(x_{j-1})$$

einsetzen der Ausdrücke für die Ableitungen:

$$b_{j-1} + 2c_{j-1}h_{j-1} + 3d_{j-1}h_{j-1}^2 = b_j$$

einsetzen der Ausdrücke für  $b_{j-1}$ ,  $d_{j-1}$  und  $b_j$  liefert:

$$\frac{y_{j-1} - y_{j-2}}{h_{j-1}} - \frac{h_{j-1}}{3}(2c_{j-1} + c_j) + 2c_{j-1}h_{j-1} + 3\frac{(c_j - c_{j-1})h_{j-1}^2}{3h_{j-1}} = \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{3}(2c_j + c_{j+1})$$

multiplizieren mit 3, kürzen und umgruppieren liefert:

$$-2c_{j-1}h_{j-1} - c_j h_{j-1} + 6c_{j-1}h_{j-1} + 3c_j h_{j-1} - 3c_{j-1}h_{j-1} + 2c_j h_j + c_{j+1}h_j = 3\left(\frac{y_j - y_{j-1}}{h_j} - \frac{y_{j-1} - y_{j-2}}{h_{j-1}}\right)$$

zusammenfassen der Ausdrücke liefert:

$$c_{j-1}h_{j-1} + 2c_j h_{j-1} + 2c_j h_j + c_{j+1}h_j = 3\left(\frac{y_j - y_{j-1}}{h_j} - \frac{y_{j-1} - y_{j-2}}{h_{j-1}}\right) = g_j$$

Der letzte Teil der erhaltenen Gleichung ist eine Definitionsgleichung für die Koeffizienten  $g$  eines linearen Gleichungssystems, das im folgenden aufgestellt wird. Faßt man den vorderen Teil der Gleichung noch weiter zusammen erhält man:

$$\underline{\underline{h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = g_j}}$$

Durch Variieren der Indices  $j$  von 2 bis  $n$  (niedrigster Index: bei  $h_{j-1}$ , daher kann erst bei 2 begonnen werden, denn 1 ist der kleinste bei  $h$  mögliche Index. Höchster Index bei  $c_{j+1}$  daher kann nur bis  $n$  variiert werden, denn  $n+1$  ist der höchste mögliche Index für  $c$ ) erhält man aus dieser Gleichung  $n-1$  Gleichungen zur Bestimmung der Unbekannten  $c_j$ . Allerdings kommen insgesamt  $c_1, c_2, \dots, c_{n+1}$ , also  $n+1$  Unbekannte vor. Die fehlenden 2 Bedingungen erhält

man durch Auswertung der Randbedingungen (Verschwinden der 2. Ableitungen am linken und rechten Rand):

$$s_1''(x_0) = 0 : \Rightarrow 2c_1 + 6d_1(x_0 - x_0) = 0 \Rightarrow c_1 = 0$$

$$s_n''(x_n) = 0 : \Rightarrow 2c_{n+1} + 6d_{n+1}(x_n - x_n) = 0 \Rightarrow c_{n+1} = 0$$

Damit muß zur Berechnung der verbleibenden unbekannten Koeffizienten  $c_i$  nur noch ein Gleichungssystem aus  $n-1$  Gleichungen aufgestellt werden. Anders betrachtet kann man sich die Zeilen und Spalten als gestrichen vorstellen, auf die die Koeffizienten  $c_1$  und  $c_{n+1}$  Einfluß haben. Das ist ein Verfahren, das auch bei der Finiten Element Methode verwendet wird, um vorgeschriebene Randbedingungen in das globale Gleichungssystem einzuarbeiten.

$$\begin{bmatrix} 2(h_1 + h_2) & h_2 & 0 & 0 & 0 & 0 \\ h_2 & 2(h_2 + h_3) & h_3 & 0 & 0 & 0 \\ 0 & h_3 & 2(h_3 + h_4) & h_4 & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 0 & h_{n-1} & 2(h_{n-1} + h_n) \end{bmatrix} \cdot \begin{bmatrix} c_2 \\ c_3 \\ \dots \\ \dots \\ \dots \\ c_n \end{bmatrix} = \begin{bmatrix} g_2 \\ g_3 \\ \dots \\ \dots \\ \dots \\ g_n \end{bmatrix}$$

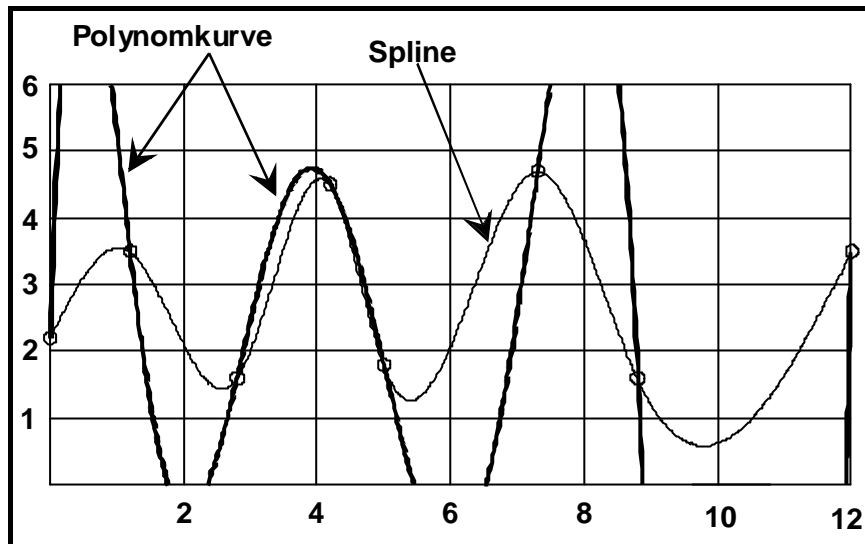
Zur Lösung des Gleichungssystems können zum Beispiel die beiden von Ihnen in Informatik 1b programmierten Routinen (Gausscher Algorithmus) benutzt werden. Allerdings nutzt dieses Verfahren nicht die Tatsache aus, daß die Matrix eine ausgeprägte Bandstruktur aufweist. Bei kleinen Matrizen ist dies nicht weiter wichtig. Bei großen Gleichungssystemen bringen Lösungsverfahren, die die Tatsache ausnutzen, daß die meisten Matrixelemente 0 sind, erhebliche Rechenzeitvorteile. Außerdem sparen diese Verfahren Speicherplatz, da sie lediglich die von null verschiedenen Matrixelemente speichern. Das müßte dann allerdings bereits bei der Erstellung der Matrix, also in den Berechnungsroutinen für die Splinekoeffizienten berücksichtigt werden. Üblich sind diese Verfahren lediglich bei Rechnungen mit sehr großen Matrizen, also z.B. bei Differenzen oder Elementverfahren.

Das in Abbildung 10 dargestellte Beispiel zeigt die Interpolation folgender 8 Punkte

	<u>x:</u>	<u>y :</u>
1.	0.0,	2.2
2.	1.2,	3.5
3.	2.8,	1.6
4.	4.2,	4.5
5.	5.0,	1.8
6.	7.3,	4.7
7.	8.8,	1.6
8.	12.0,	3.5

- durch: 1. ein Polynom vom Grad 7  
2. eine Splinekurve

Das Polynom (dicke Kurve) schwingt so stark durch, daß es den Bereich des Diagrammgitters mehrfach verläßt. Dagegen verbindet die Splinekurve (dünne Kurve) die Punkte in optisch glatter Form ohne überflüssige Schwingungen.



**Abb. 10: Polynomkurve vom Grad 7 (dick) durch 8 Stützpunkte gelegt und interpolierende SPLINE-kurve durch dieselben Punkte**

Eine erhebliche Vereinfachung für das Verfahren ergibt sich, wenn die Stützstellen äquidistant verteilt sind, das heißt  $h_1=h_2=h_3=\dots=h_n$ .

Für diesen Fall vereinfacht sich die Form des Bestimmungsgleichungssystems für die  $c_j$  zu:

$$\begin{bmatrix} 4 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 0 & 1 & 4 \end{bmatrix} \cdot \begin{bmatrix} c_2 \\ c_3 \\ \dots \\ \dots \\ \dots \\ c_n \end{bmatrix} = \frac{3}{h^2} \begin{bmatrix} y_0 - 2y_1 + y_2 \\ y_1 - 2y_2 + y_3 \\ \dots \\ \dots \\ \dots \\ y_{n-2} - 2y_{n-1} + y_n \end{bmatrix}$$

Der Faktor  $h$  aus der Matrix steckt in dieser Form in dem Faktor  $3/h^2$  vor dem Vektor der rechten Seite. Die Terme zur Bestimmung der übrigen Koeffizienten vereinfachen sich ebenfalls etwas zu:

$$c_1 = 0$$

$$c_{n+1} = 0$$

$$a_j = y_{j-1}$$

$$b_j = \frac{1}{h}(y_j - y_{j-1}) - \frac{h}{3}(2c_j + c_{j+1})$$

$$d_j = \frac{1}{3h}(c_{j+1} - c_j)$$

**Ein Programm, das keine äquidistanten Stützstellen benötigt und mit dessen Hilfe Abbildung 10 erzeugt wurde, ist bei mir im Quellcode erhältlich. Fälle bei denen  $h=0$  ist (ggf. mehrfach) und die zu Singularitäten der Matrix führen könnten, sind darin allerdings nicht abgefangen.**

### 2.3.2 Parametrische Splines

Bei den Polynomkurven, die mit der Fehlerquadratmethode ermittelt wurden, spielt es keine Rolle, ob die Punktkoordinaten in aufsteigender Reihenfolge sortiert sind. Der Polynomansatz wird für die ganze Kurve vorgegeben und die Punkte dienen unabhängig von ihrer Reihenfolge zur Bestimmung der Koeffizienten.

Anders dagegen bei den Splines. Ein Polynom wird immer für den Abschnitt zwischen zwei aufeinander folgenden Punkten berechnet. Ist nun  $x_1 < x_2 > x_3$  u.s.w., so muß irgendwo eine senkrechte Tangente, also eine unendliche Steigung auftreten. Zu einem  $x$ -Wert gäbe es mehr als einen  $y$ -Wert. Die Kurve stellt im  $x$ - $y$ -Diagramm keinen eindeutigen Zusammenhang  $y=f(x)$  dar. Sie sähe z.B. so aus:

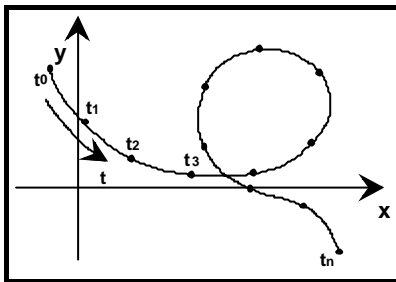


Abb. 11: Kurve in Parameterdarstellung.

Zur Darstellung eines derartigen Kurvenverlaufs beschreibt man die  $x$ - und die  $y$ -Koordinate unabhängig voneinander in Abhängigkeit eines gemeinsamen Parameters. Diese Art der Kurvenrepräsentation ist bekannt. Man spricht von Parameterdarstellung. Für den Kreis lautet sie:

$$x = r \cdot \cos j$$

$$y = r \cdot \sin j$$

Dabei ist  $r$  der Kreisradius und  $j$  der Winkel, der hier der gemeinsame Parameter ist.

Auch Splines sind in parametrischer Form darstellbar. Statt der Wertepaare  $x_i, y_i$  sind jeweils dieselbe Anzahl Wertepaare  $t_j, x_j$  und  $t_j, y_j$  vorzugeben. Die Bedeutung des gemeinsamen Parameters  $t$  ist eine Frage der Interpretation. Mißt man den Ort eines sich bewegendes Punktes in einer Ebene, so kann er z.B. die Zeit in Sekunden sein. Genausogut kann er aber auch als Bogenlänge der Bahnkurve aufgefaßt werden.

Die im vorherigen Kapitel aufgeführten Zusammenhänge gelten für parametrische Splines in analoger Weise. Das Gleichungssystem zur Bestimmung der Koeffizienten muß lediglich zweimal gelöst werden und zwar einmal für die Koeffizienten  $a_{jx}, b_{jx}, c_{jx}, d_{jx}$  und einmal für die Koeffizienten  $a_{jy}, b_{jy}, c_{jy}, d_{jy}$ . Dabei sind einmal die Wertepaare  $t_j, x_j$  und einmal die Wertepaare  $t_j, y_j$  zu verwenden.

Die Gleichungen der Splinefunktionen in Parameterdarstellung lauten dann:

$$s_{jx}(t) = x(t) = a_{jx} + b_{jx}(t - t_{j-1}) + c_{jx}(t - t_{j-1})^2 + d_{jx}(t - t_{j-1})^3$$

$$s_{jy}(t) = y(t) = a_{jy} + b_{jy}(t - t_{j-1}) + c_{jy}(t - t_{j-1})^2 + d_{jy}(t - t_{j-1})^3$$

Es ist offensichtlich, daß sich diese Art der Kurvendarstellung leicht auf räumliche Darstellungen erweitern läßt. Dazu ist lediglich die Parameterdarstellung für die  $z$ -Koordinate in analoger Form hinzuzufügen.



### 2.3.3 Periodische Splines

Die bisher behandelte Art der Splines mit freien Kurvenenden nennt man natürliche Splines. Daneben gibt es noch die periodischen Splines, die geschlossene Kurven beschreiben. Für diese Splines gilt ebenfalls das Gleichungssystem

$$\begin{bmatrix} 2(h_1 + h_2) & h_2 & 0 & 0 & 0 & 0 \\ h_2 & 2(h_2 + h_3) & h_3 & 0 & 0 & 0 \\ 0 & h_3 & 2(h_3 + h_4) & h_4 & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 0 & h_{n-1} & 2(h_{n-1} + h_n) \end{bmatrix} * \begin{bmatrix} c_2 \\ c_3 \\ \dots \\ \dots \\ \dots \\ c_n \end{bmatrix} = \begin{bmatrix} g_2 \\ g_3 \\ \dots \\ \dots \\ \dots \\ g_n \end{bmatrix}$$

zur Bestimmung der Koeffizienten  $c_2$  bis  $c_n$ . Auch die Substitutionsformeln zur Bestimmung der Koeffizienten  $a_i$ ,  $b_i$  und  $d_i$  gelten analog. Die Bestimmung der Koeffizienten  $c_1$  und  $c_{n+1}$  allerdings muß anders ablaufen, weil es keine freien Ränder gibt. Statt dessen wird Gebrauch gemacht von der Bedingung, daß Steigung und Krümmung an den Enden übereinstimmen müssen, also die beiden ersten Ableitungen. Es gilt also:

$$s'_1(x_0) = s'_n(x_n) \quad \text{und} \quad s''_1(x_0) = s''_n(x_n)$$

Aus der Bedingung der Übereinstimmung in den 2. Ableitungen folgt:

$$2c_1 + 6d_1(x_0 - x_0) = 2c_n + 6d_n(x_n - x_{n-1})$$

mit  $x_0 - x_0 = 0$ ,  $x_n - x_{n-1} = h_n$  und der Substitutionsformel für  $d_n$  folgt:

$$2c_1 = 2c_n + 6h_n \frac{c_{n-1} - c_n}{3h_n} \Rightarrow c_1 = c_n + c_{n+1} - c_n$$

und damit:

$$\underline{c_1 = c_{n+1}}$$

Aus der Bedingung der Übereinstimmung in den 1. Ableitungen folgt:

$$b_1 + 2c_1(x_0 - x_0) + 3d_1(x_0 - x_0)^2 = b_n + 2c_n(x_n - x_{n-1}) + 3d_n(x_n - x_{n-1})^2$$

bzw. mit den oben gemachten Ersetzungen:

$$b_1 = b_n + 2c_n h_n + 3d_n h_n^2$$

einsetzen der Beziehungen für  $b_1$ ,  $b_n$  und  $d_n$  liefert:

$$\frac{y_1 - y_0}{h_1} - \frac{h_1}{3}(2c_1 + c_2) = \frac{y_n - y_{n-1}}{h_n} - \frac{h_n}{3}(2c_n + c_{n+1}) + 2c_n h_n + \frac{3(c_{n+1} - c_n)h_n^2}{3h_n}$$

multiplizieren mit 3, kürzen des letzten Ausdrucks und Umstellen liefert:

$$\begin{aligned} \frac{y_1 - y_0}{h_1} - \frac{h_1}{3}(2c_1 + c_2) &= \frac{y_n - y_{n-1}}{h_n} - \frac{h_n}{3}(2c_n + c_{n+1}) + 2c_n h_n + \frac{3(c_{n+1} - c_n)h_n^2}{3h_n} \\ 3\left(\frac{y_1 - y_0}{h_1} - \frac{y_n - y_{n-1}}{h_n}\right) &= h_1(2c_1 + c_2) - h_n(2c_n + c_{n+1}) + 6c_n h_n + 3c_{n+1} h_n - 3c_n h_n \end{aligned}$$

die weitere Zusammenfassung liefert unter Ausnutzung der Beziehung

$$c_1 = c_{n+1}$$

$$\underline{\underline{3 \left( \frac{y_1 - y_0}{h_1} - \frac{y_n - y_{n-1}}{h_n} \right) = c_n h_n + 2c_1 (h_1 + h_n) + c_2 h_1}}$$

Diese Beziehung entspricht genau der Form der Systemgleichungen zur Berechnung der  $c_j$ . Der Ausdruck der linken Seite entspricht dabei dem  $g=g_{\bar{u}}$  für die Übergangsstelle des Splines. Damit folgt für  $c_1$ :

$$\underline{\underline{c_1 = \frac{g_{\bar{u}} - c_n h_n - c_2 h_1}{2(h_1 + h_n)}}}$$

Numerisch auswertbar wird dieser Ausdruck nach Lösung des Gleichungssystems für die  $c_j$ , denn dann stehen die Werte für  $c_2$  und  $c_n$  zur Verfügung. Mit  $c_1$  ist dann auch  $c_{n+1}$  bekannt.

## 2.4 Darstellung von Kurven mit Hilfe verschiedener polynomialer Basen

### 2.4.1 Einführung: Ferguson-Basis und Hermite'sche Basis

Konstrukteure, die Formen gestalten müssen, die nicht kanonisch beschreibbar sind, haben in der Regel den Wunsch, die von Ihnen gestalteten Kurven noch verändern zu können, um sich „iterativ“ an das gewünschte Ergebnis herantasten zu können. In den seltensten Fällen wird ein Konstrukteur ein Polynom angeben können, die entsprechende Kurve zeichnen lassen und danach sagen können, „so wollte ich es haben“. Wenn er dann einen der Koeffizienten des Polynoms verändert, ändert sich die Form der Kurve. Aber ändert sie sich so wie er es haben wollte? Was muß man tun, wenn man bei einem Polynom der Form

$$y=ax^3+bx^2+cx+d$$

lediglich die Steigung am Anfangspunkt verändern möchte ?

Eine erste Antwort auf diese Frage stammt von einem britischen Ingenieur namens Ferguson. Er betrachtete ein Polynom 3. Grades in Parameterdarstellung:

$$p_x(u) = a_{0x} + a_{1x}u + a_{2x}u^2 + a_{3x}u^3$$

$$p_y(u) = a_{0y} + a_{1y}u + a_{2y}u^2 + a_{3y}u^3 \quad (\text{Der Parameter } u \text{ variiert zwischen 0 und 1})$$

$$p_z(u) = a_{0z} + a_{1z}u + a_{2z}u^2 + a_{3z}u^3$$

Um die Koeffizienten  $a_{ij}$   $i=0,\dots,3$ ;  $j=x,y,z$  bestimmen zu können, gab Ferguson 4 Randbedingungen vor, und zwar für jede Koordinate (x,y,z) die Anfangs- und Endpunkte der Kurve sowie die Steigungen in den Anfangs- und Endpunkten. Damit ergaben sich pro Koordinate die folgenden 4 Gleichungen zur Bestimmung der insgesamt 12 unbekannten Koeffizienten:

$$p(0)=a_0$$

$$p(1)=a_0+a_1+a_2+a_3$$

$$p'(0)=a_1$$

$$p'(1)=a_1+2a_2+3a_3$$

Löst man dieses Gleichungssystem nach den Koeffizienten  $a_0$ ,  $a_1$ ,  $a_2$  und  $a_3$  auf und substituiert die Koeffizienten in  $p(u)$ , so ergibt sich für jede Koordinate:

$$\begin{aligned} p(u) &= p(0) + p'(0)u \\ &\quad + \{3[p(1) - p(0)] - 2p'(0) - p'(1)\}u^2 \\ &\quad + \{2[p(0) - p(1)] + p'(0) + p'(1)\}u^3 \end{aligned}$$

umgeformt:

$$p(u) = p(0) * [2u^3 - 3u^2 + 1] + p'(0) * [u^3 - 2u^2 + u] + p(1) * [-2u^3 + 3u^2] + p'(1) * [u^3 - u^2]$$

oder in Matrixform geschrieben:

$$p(u) = \underbrace{\{u^3 \ u^2 \ u \ 1\}}_{\text{Ferguson-Basis}} * \underbrace{\begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{\text{Koeffizienten}} * \underbrace{\begin{Bmatrix} p(0) \\ p'(0) \\ p(1) \\ p'(1) \end{Bmatrix}}_{\text{Koeffizienten}}$$
  

$$\underbrace{\hspace{10em}}_{\text{Hermite'sche Basis}} \quad \underbrace{\hspace{10em}}_{\text{Koeffizienten}}$$

Man erkennt hier, daß das gesamte Polynom eine Produktsumme ist, deren Produkte aus „Unterpolyomen“ mit den Randbedingungen ( $p(0)$  usw.) zusammengesetzt sind. Diese Unterpolyome nennt man **Basispolynome**, **polynomiale Basis** oder auch kurz nur **Basis**.

Ob man in diesem Fall lediglich den Term links von der Matrix als Basis bezeichnet (sogenannte Ferguson Basis, siehe oben), oder das Produkt dieses Terms mit der Matrix (sogenannte Hermite'sche Basis, siehe oben), ist willkürlich. In der Literatur findet man, benannt nach den Autoren, beide Möglichkeiten. Einmal die sogenannte Ferguson Basis und einmal die sogenannte Hermite'sche Basis.

An dieser Aufspaltung der Polynome wird deutlich, wie verflochten die anschaulichen Randbedingungen ( $p(0), \dots$ ) in den Koeffizienten des Polynoms stecken. Hier hat man eine Möglichkeit, gezielt die Randbedingungen zu verändern, was durch Änderung eines Polynomkoeffizienten nicht möglich gewesen wäre.

Wer aber sagt, daß als polynomiale Basis nur die reinen Potenzen des Parameters  $u$  bezeichnet werden dürfen? Multipliziert man bei der Auswertung des Polynoms zuerst den linken Term mit der Matrix, so erhält man wiederum Polynome, die in einem Vektor angeordnet sind, die sogenannte Hermite'sche Basis. Diese Basispolynome haben einen etwas komplizierteren Aufbau als die Ferguson Basis, aber was als Koeffizienten übrigbleibt hat eine sehr anschauliche Bedeutung. Die Hermitepolynome werden in der Finite Element Technik zur Interpolation der Zustandsgrößen in Balkenelementen verwendet, da dabei die Randbedingungen eine gut angebbare mechanische Bedeutung haben (Durchbiegung und Neigung).

Den Gedanken der Aufspaltung von Polynomen in Basispolynome und Koeffizienten mit möglichst anschaulicher Bedeutung haben auch andere Mathematiker aufgegriffen. Die Fragestellung, die dabei zu beantworten ist, könnte etwa folgendermaßen lauten:

**Welche (anschauliche) Bedeutung sollen die Koeffizienten meiner Polynomdarstellung haben und wie müssen demzufolge meine Basispolynome aussehen ???**

Für die Gestaltung von Kurven in CAD-Systemen haben sich in den letzten Jahrzehnten Verfahren durchgesetzt, die als Basispolynome die sogenannten Bernsteinpolynome oder die Basis-Spline-Funktionen (abgekürzt B-Spline-Funktionen) benutzen und als Koeffizienten die Stützpunkte. Allerdings verlaufen die Kurven nicht durch alle Stützpunkte sondern nur durch einige. Insofern ist der Begriff Stützpunkt hier nicht mehr korrekt. Man spricht richtigerweise von sogenannten Kontrollpunkten, da sie den Verlauf der Kurven zwar kontrollieren, die Kurve

jedoch nicht zwangsläufig hindurchgeht. Insofern haben die Kurven bezogen auf die Kontrollpunkte teils approximierenden teils interpolierenden Charakter. Wie die Kontrollpunkte die Kurve in den nicht interpolierenden Bereichen beeinflussen, wird noch besprochen.

## 2.4.2 Kurvendarstellung nach Bézier

Die vielleicht bekannteste Art der Kurvendarstellung unter Verwendung von Basispolynomen geht auf den französischen Mathematiker Bézier zurück. Er ist damit der Urheber des Freiform-CAD-Paketes „UNISURF“, das seit vielen Jahren in immer verbesserten Versionen als Freiform-Kern im System EUCLID läuft. Auch als spezielles und individuell verbessertes Einzelsystem war und ist es z.B. bei Renault und bei VW im Einsatz. Nach der Methode von Bézier dargestellte Kurven und Flächen werden aber auch in vielen anderen CAD-Systemen verwendet. So z.B. auch in dem in der FH neu beschafften System CATIA.

Bézier verwendet als polynomiale Basis die sogenannten Bernstein Polynome. Man bezeichnet die damit verbundene Art der Kurvenrepräsentation deshalb auch als Bézier-Bernstein-Approximation (im Gegensatz zur Bézier-B-Spline-Approximation, die später besprochen wird).

Die Bernstein-Polynome sind folgendermaßen definiert:

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \quad 0 \leq t \leq 1 \quad \text{und} \quad 0 \leq i \leq n$$

Dabei repräsentiert der Index n den Grad der Polynome. Für den oft benutzten Fall kubischer Polynome (n=3) sehen die Bernsteinpolynome folgendermaßen aus:

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

$$B_{1,3}(t) = 3t - 6t^2 + 3t^3$$

$$B_{2,3}(t) = 3t^2 - 3t^3$$

$$B_{3,3}(t) = t^3$$

Die Koeffizienten der Kurvendarstellung sind die Koordinaten der Kontrollpunkte. Damit sieht die Kurvendarstellung folgendermaßen aus (hier nur für die x-Koordinate):

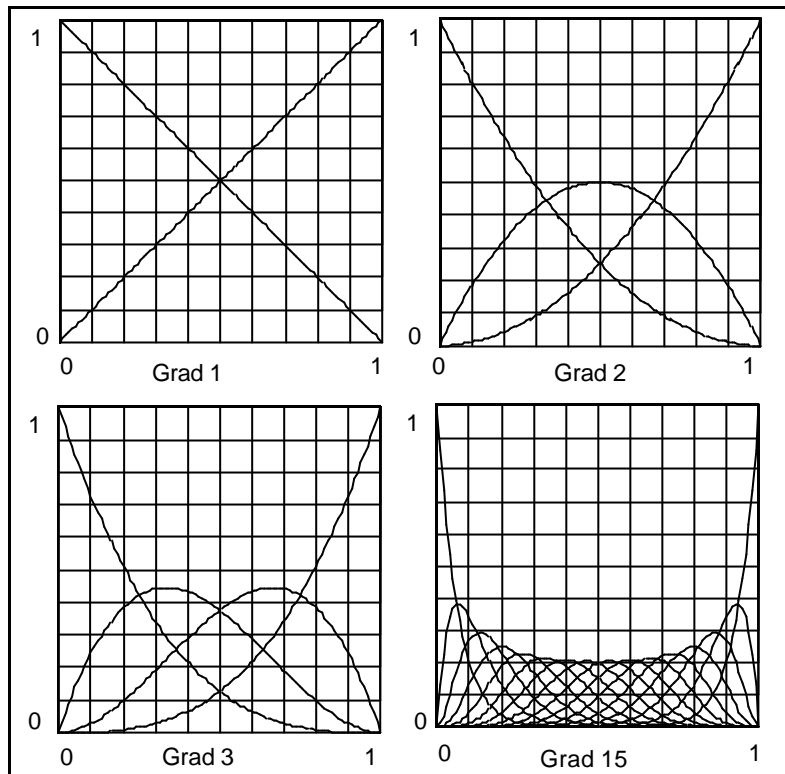
$$x(t) = x_0(1 - 3t + 3t^2 - t^3) + x_1(3t - 6t^2 + 3t^3) + x_2(3t^2 - 3t^3) + x_3 t^3$$

oder in Matrixdarstellung (in Anlehnung an das vorausgegangene Kapitel):

$$x(t) = \begin{Bmatrix} 1 & t & t^2 & t^3 \end{Bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} * \begin{Bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

Analog sehen die Darstellungen für y(t) und bei 3D-Kurven auch für z(t) aus.

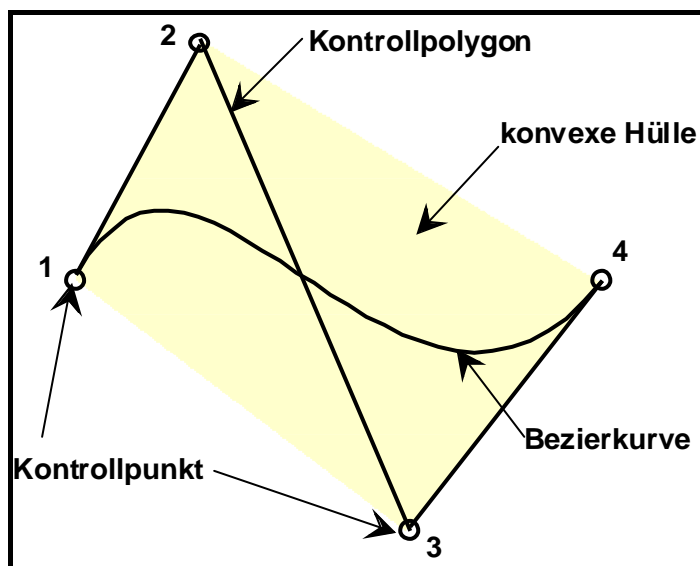
Das Aussehen der Bernstein-Polynome ist in Abbildung 12 für verschiedene Polynomgrade dargestellt. Die Zahl der Kurven entspricht jeweils der Zahl der Stützstellen. Diese ist um eins größer als die durch sie definierten Kurvenabschnitte. Die Zahl der Abschnitte wiederum entspricht dem Grad der Bezierkurve und damit dem Grad der Bernstein-Polynome.



**Abb. 12:**  
Die Bernsteinpolynome für  
die Polynomgrade 1, 2, 3  
und 15

Man erkennt einige Eigenschaften der Bernsteinpolynome: Die Summe aller Kurven ist an jeder Stelle des Intervalls immer 1. Besonders deutlich wird das, wenn man sich die jeweils erste und letzte Kurve anschaut. Diese nehmen an den Rändern immer den Wert 1 an, während alle anderen Kurven an den Rändern verschwinden.

Es bleibt festzustellen, welche Bedeutung den Koeffizienten einer Bezierkurve, also den Kontrollpunkten, zukommt. Dazu betrachte man Abbildung 13.



**Abb. 13:** Einzelne Bezierkurve mit  
Kontrollpolygon  
und konvexer Hülle

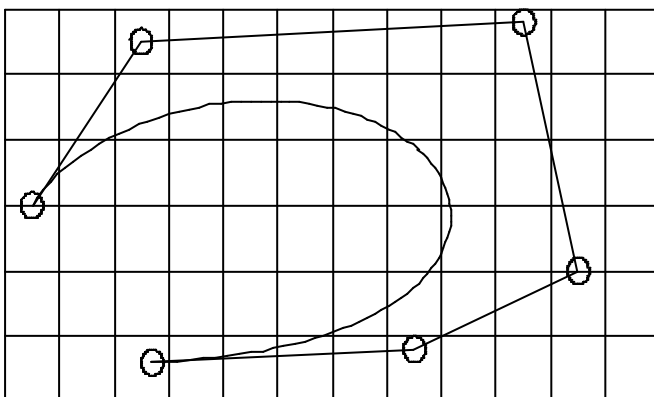
Das Kontrollpolygon einer Bezierkurve hat (Abb. 13) folgende Eigenschaften:

- Der erste und der letzte Punkt des Kontrollpolygons fallen mit Anfangs- und Endpunkt der Kurve zusammen. Dies wird klar, wenn man sich vergegenwärtigt, daß jeweils nur die

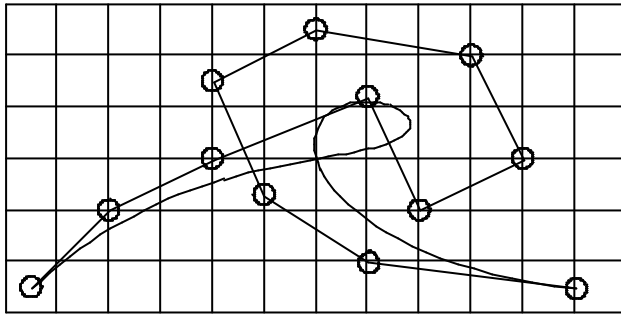
erste und die letzte Kurve der Bernsteinpolynome am Rand den Wert 1 annehmen, die anderen Kurven dagegen gegen 0 gehen. Da der erste Kontrollpunkt mit der ersten Kurve multipliziert wird und der letzte mit der letzten, bleiben nur diese beiden Werte an den Rändern übrig. Man kann sich davon aber auch dadurch überzeugen, daß man in die Gleichung der Bezierkurve die Parameterwerte  $t=0$  und  $t=1$  einsetzt.

- Das Kontrollpolygon spannt eine „konvexe Hülle“ auf, in der die Kurve vollständig eingebettet ist. Die konvexe Hülle kann man sich dadurch erzeugt denken, daß man ein Seil um das Kontrollpolygon spannt.
- Die Verbindungslinie zwischen erstem und zweitem Kontrollpunkt sowie die zwischen letztem und vorletztem Kontrollpunkt fallen mit der Tangentenrichtung der Kurve am Anfangs- bzw. am Endpunkt zusammen.
- Zählt man die Kontrollpunkte von 0 bis  $n$ , so kann man (anschaulich) sagen, daß die Punkte 1 bis  $n-1$  auf die Kurve eine „Anziehungskraft ausüben“. Am deutlichsten sieht man dies an den Punkten 1 und  $n-1$ . Wenn man die verschiebt, ändert man die Tangentenrichtung. Vergrößert man den Abstand zwischen den ersten beiden Randpunkten, verstärkt man den Einfluß der Tangente auf den Kurvenverlauf und damit das Schmiegeverhalten der Kurve an den Enden. Verschiebt man die übrigen Punkte, wird man feststellen, daß die Kurve der Verschiebung folgt. Als Analogon stelle man sich vor, daß man das Fadenkreuz einer Marionette bewegt, bei der die äußeren Fäden aus Zwirn, die inneren dagegen aus Gummi sind.

Das Schöne an den Bezierkurven ist, daß sowohl Basispolynome als auch Koeffizienten (=Kontrollpunkte) vorgegeben werden. Es sind also keine Gleichungssysteme zu lösen um Koeffizienten zu bestimmen. Die Kontrollpunkte werden zur Kurvengestaltung vorgegeben (und können im CAD-System interaktiv verändert werden) und die Gleichungen für den Kurvenverlauf stehen fest. Was numerisch noch zu tun ist, ist lediglich das punktweise Ausrechnen und Darstellen der Kurven. Will man dies so tun, daß das Programm für möglichst alle (vernünftigen) Vorgaben funktionieren soll, so ist das immerhin auch etwas Arbeit. Abbildung 14 zeigt als Beispiel eine Bezierkurve nach einem Polygon aus 6 Punkten.



**Abb. 14: Bezierkurve mit  
Kontrollpolygon aus  
6 Stützstellen (Grad 5)**

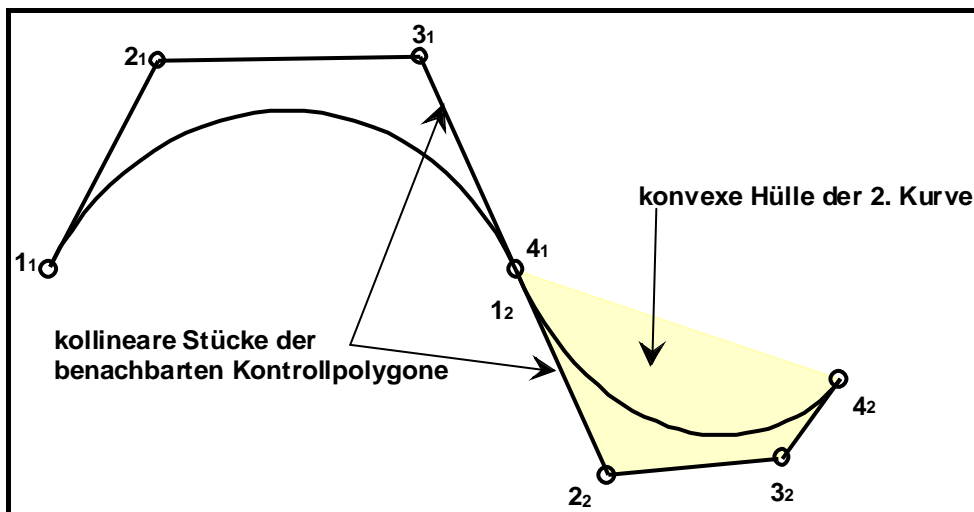


**Abb. 15: Bezierkurve mit  
Kontrollpolygon aus  
12 Stützstellen (Grad 11)**

Wie man vor allem in Abbildung 15 sieht, weichen die Kurven unter Umständen erheblich vom Kontrollpolygon ab. Die „Marionettengummibänder“ (siehe oben) sind also weich.

Arbeitet man an Bezierkurven hohen Grades, also mit vielen Kontrollpunkten, so wird man beim Verschieben feststellen, daß der Kurvenverlauf nur schwer glatt zu halten ist. Die Kurve wird leicht wellig. Das entspricht natürlich dem Verhalten von Polygonen hohen Grades. Aber beim Modellieren von Freiformkurven ist meist ein glatter Verlauf über weite Strecken erwünscht. Hier wäre eine Kurve niedrigen Grades unter Umständen vorteilhafter. Da man jedoch zur Beschreibung großer Strecken (z.B. Autodach) meist viele Punkte vorgegeben bekommt und der Grad von Bezierkurven an die Kontrollpunktzahl gebunden ist, erhält man auch Kurven hohen Grades.

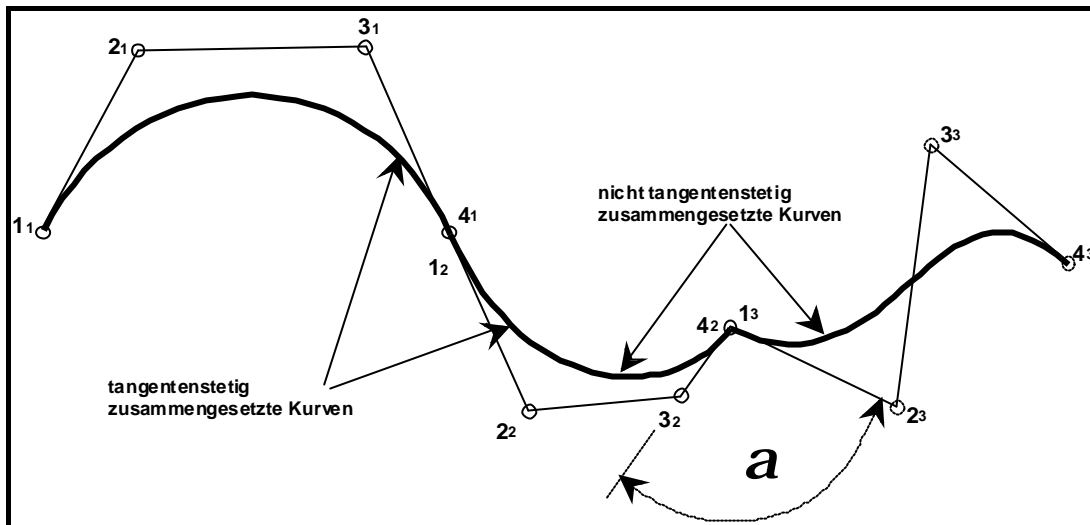
Eine Alternative wäre das Zusammenfügen von mehreren Kurven niedrigeren Grades. Aber auch das ist nicht unproblematisch. Setzt man zwei Bezierkurven zusammen, so will man das meist unter Beachtung bestimmter Stetigkeitsanforderungen tun. In der Regel genügt Tangentenstetigkeit. Das ist bei Bezierkurven gut kontrollierbar. Man muß lediglich dafür sorgen, daß der gemeinsame Kontrollpunkt an der Fügestelle kollinear mit den beiden benachbarten Punkten liegt, da ja die beiden Endpunkte einer Kurve die Tangentenrichtung am End- bzw. Anfangspunkt bestimmen, vergl. Abbildung 15.



**Abb. 15: tangentenstetig zusammengesetzte Bezierkurven  
(Kollinearität der benachbarten Stücke des Kontrollpolygons)**



Andererseits kann es vorkommen, daß Kurven ganz gezielt unstetig aneinandergefügt werden sollen, wobei der Winkel zwischen den Randtangenten relevant sein kann. Auch das ist bei Bezierkurven über den Winkel zwischen den Kontrollpolygonen gut steuerbar, vergl. Abbildung 16.



**Abb. 16: zusammengesetzte Bezierkurven:**  
**erste Fügung tangentenstetig,**  
**zweite Fügung mit definiertem Tangentenwinkel  $a$**

Bei Einhaltung von Stetigkeitsanforderungen an den Fügstellen kann neben Tangentenstetigkeit auch Krümmungsstetigkeit gefragt sein. Dabei ist folgendes zu beachten: Bildet man die 1. und die 2. Ableitung einer Bezierkurve (siehe Gleichung Seite 20) und setzt die Parameterwerte für die Randpunkte der Kurve ein ( $t=0$  und  $t=1$ ) so läßt sich damit folgendes verifizieren:

$$B'(t=0) = n(k_1 - k_0)$$

$$B'(t=1) = n(k_n - k_{n-1})$$

$$B''(t=0) = n(n-1)(k_2 - 2k_1 + k_0)$$

$$B''(t=1) = n(n-1)(k_n - 2k_{n-1} + k_{n-2})$$

Hier steht  $B'$  bzw  $B''$  für die 1. und 2. Ableitung einer beliebigen Bezierkurve, die  $k_j$  sind die Kontrollpunkte und  $n$  ist der Grad der Polynome.

Aus diesen Beziehungen läßt sich folgendes ablesen:

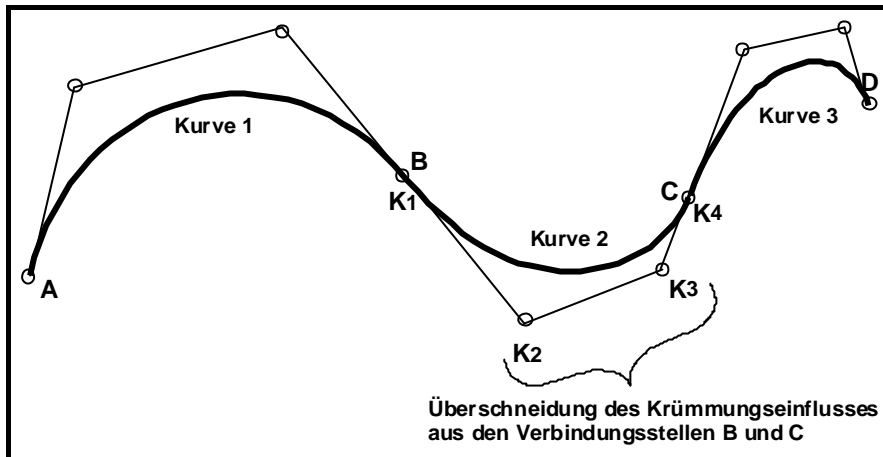
- Die 1. Ableitungen (Steigungen) sind durch die ersten beiden Polygonpunkte bestimmt (siehe auch Eigenschaften des Kontrollpolygons).
- Ein weiterer Polygonpunkt bestimmt die 2. Ableitungen (Krümmungen).

Fügt man also Bezierkurven niedrigen Grades, z.B. vom Grad 3, zusammen, um glatte Kurven zu erhalten, so überlappen sich die aus der Erfüllung der Krümmungsstetigkeit folgenden Einflüsse der Kontrollpolygone in jeder Kurve.

Für den in Abbildung 17 dargestellten Fall bedeutet dies:

- Krümmungsstetiger Übergang am **Knoten B**:  
Die Krümmung ist beeinflusst durch Knoten **K1, K2 und K3**
- Krümmungsstetiger Übergang am **Knoten C**:  
Die Krümmung ist beeinflusst durch Knoten **K4, K3 und K2**

=> Überlappung des Einflusses der Krümmungsstetigkeit in den **Knoten K2 und K3**



**Abb. 17: Überschneidung des Einflusses krümmungsstetiger Kurvenübergänge**

Krümmungsstetigkeit bedeutet gleiche Krümmung auf beiden Seiten des Verbindungspunktes. Folglich sind an den Intervallgrenzen 5 Punkte voneinander abhängig. Wenn alle Verbindungsstellen zwischen den Einzelkurven krümmungsstetig sind bedeutet das, daß sich der überschneidende Einfluß der Krümmungsstetigkeit über die gesamte Kurve erstreckt. Das wiederum bedeutet, daß sich jede lokale Änderung an einer Einzelkurve auf den Verlauf der Gesamtkurve auswirkt. Mit anderen Worten sind keine lokalen Kurvenänderungen möglich. Es sei denn, man erhöht den Polynomgrad der Einzelkurven, so daß sich die Einflüsse aus den Krümmungen an den Übergangsstellen nicht mehr überschneiden. Damit gibt man jedoch den Vorteil des niedrigen Grades und der daraus resultierenden glatten Kurven auf.

In der Praxis verzichtet man sehr oft auf krümmungsstetige Übergänge. Zusätzlich werden vielfach Kurven mit einem Polynomgrad größer 3 verwendet. Sehr gebräuchlich ist der Grad 5. Vielfach werden Kurven zusammengesetzt, deren Teilstücke unterschiedliche Polynomgrade haben, so daß die Gesamtkurve lokalen Gegebenheiten angepaßt werden kann.

**Ein Programm, mit dem die hier abgebildeten Bézier-Kurven erzeugt wurden, ist bei mir im Quellcode erhältlich.**

### 2.4.3 Kurvendarstellung mit Hilfe von Basis Splines (B-Splines)

Alle bisher besprochenen Kurvendarstellungen, nämlich interpolierende Polynome, Kurven mit Hermitescher oder Ferguson Basis, Splines und Bezierkurven haben den Nachteil, daß sich bei Veränderung einer Stützstelle die gesamte Kurve ändert. Dieser Nachteil ist bei den sogenannten B-Splines nicht mehr gegeben. Die Kopplung des Polynomgrades an die Anzahl der Stützstellen ist aufgehoben und es kommt der Vorteil lokaler Änderbarkeit hinzu. Im

Prinzip kann man die Kurvendarstellung durch B-Splines als Bézierkurvendarstellung auffassen, bei der die Bernsteinpolynome durch die Basis-(B)-Spline-Kurven ersetzt werden.

Die Definition der B-Spline-Basisfunktionen ist bekannt als Cox-deBoor Algorithmus. Es handelt sich um eine rekursive Definition, die gegenüber den sonst benutzten geschlossen darstellbaren Funktionen gewöhnungsbedürftig ist:

**Definition der B-Spline-Basisfunktionen:**

$$B_{i,0}(t) = \begin{cases} 1 & \text{für } t_i \leq t \leq t_{i+1} \\ 0 & \text{anderenfalls} \end{cases} \quad (B1)$$

$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} B_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(t) \quad (B2)$$

mit:

$T = \{t_0, t_1, \dots, t_i, t_{i+1}, \dots, t_m\}$  Knotenvektor (hat nichts mit den Stützpunkten zu tun)

und:

$$t \in [t_0, t_m]$$

außerdem gilt:  $\frac{0}{0} = 0$

$i$  := laufende Nummer der jeweils berechneten Basisfunktion

$0 \leq i \leq n$ ,  $n+1$  = Anzahl der Kontrollpunkte (Zählung beginnt bei 0 !!!)

$k$  := Polynomgrad der jeweils berechneten Basisfunktion

$0 \leq k \leq p$   $p$  := gewünschter Polynomgrad (vorzugeben)

$m$  := Anzahl Elementes des Knotenvektors:  $m = n + p + 1$

Mit den so definierten Basisfunktionen läßt sich die Gleichung der B-Spline-Kurve in Parameterdarstellung schreiben. Hier stellvertretend für alle Koordinaten lediglich für  $x$ :

$$x(t) = x_0 B_{0,p}(t) + x_1 B_{1,p}(t) + \dots + x_n B_{n,p}(t) = \sum_{i=0}^n x_i B_{i,p}(t)$$

wie man es der Form nach auch von den Bézier-Kurven kennt.

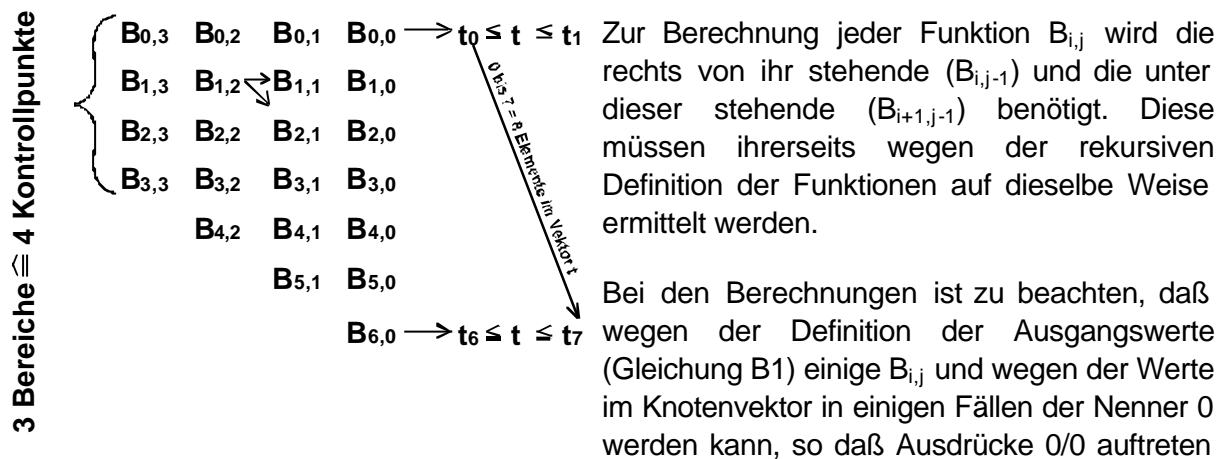
Beim Betrachten der Definition der Basisfunktionen wird klar, daß es sich um ein rechnerorientiertes Vorgehen handelt. Allerdings hat es gegenüber anderen Verfahren erst relativ spät (ab 1973 durch R. Riesenfeld) Einzug in den Bereich CAD/CAM gehalten. Seitdem wurde es in diversen Varianten mehr und mehr zur bevorzugten Art der Kurven- und Flächendarstellung.

Das Berechnen einer einzelnen Basisfunktion läuft nach obiger Definition folgendermaßen ab:

1. Zunächst definiert man sich neben dem Knotenvektor  $T$  (siehe Definition) einen weiteren Vektor (eindimensionales array), den Basisvektor. Dieser Begriff ist keine standardisierte Bezeichnung. Das Feld dient bei der rekursiven Berechnung zur Speicherung der im vorherigen Schritt berechneten Werte der  $B_{i,k}$ .

- Man beginnt mit der Berechnung der Basiswerte nach Gleichung (B1), für den Polynomgrad  $k=0$  ( $p+1$  Werte).  
Beispiel: Für die Berechnung der 3. Basisfunktion vom Grad 3 (also  $B_{3,3}$ ) müssen in diesem Schritt die Basisfunktionen  $B_{3,0}$ ,  $B_{4,0}$ ,  $B_{5,0}$ , und  $B_{6,0}$  berechnet werden, da sie in den weiteren Berechnungsschritten für die Berechnung der Basisfunktion  $B_{3,3}$  benötigt werden.
- In den weiteren Schritten ermittelt man nun die Basisfunktionen höheren Grades nach Gleichung (B2), wobei jeweils eine Funktion weniger zu ermitteln ist als bei den Funktionen niedrigeren Grades. Im Beispiel also  $B_{3,1}$ ,  $B_{4,1}$ ,  $B_{5,1}$ , dann  $B_{3,2}$ ,  $B_{4,2}$ , und dann  $B_{3,3}$ .

Die Berechnung der Basisfunktionen wird durch folgendes Schema veranschaulicht:



Ein besonderes Kapitel ist der Inhalt des Knotenvektors, also die Verteilung der Knoten auf den gesamten Wertebereich (0 bis 1) des Parameters  $t$  (Achtung: die Knoten des Knotenvektors sind nicht mit den Kontrollpunkten zu verwechseln, noch stimmt deren Anzahl mit der Anzahl der Knoten überein!).

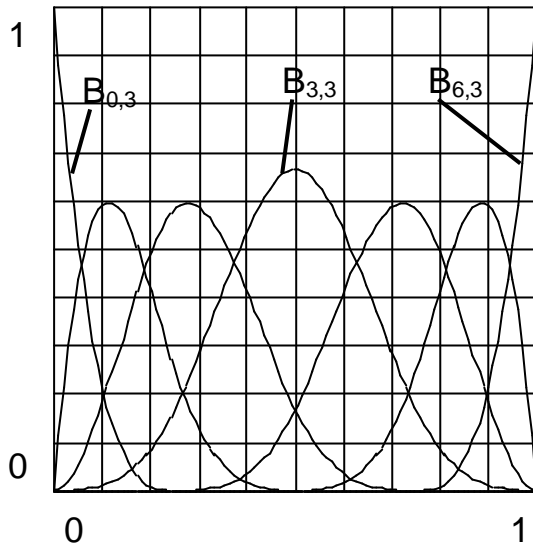
Der Knotenvektor hat  $m=n+p+1$  Elemente ( $n$ =Anzahl der Kontrollpunkte,  $p$ =gewählter Grad der Basispolynome). Die ersten  $p+1$  und die letzten  $p+1$  Knoten haben jeweils den gleichen Wert (dann spricht man von nicht periodischen Splines). In der Regel benutzt man normierte Knotenvektoren, deren Werte von 0 bis 1 variieren, also  $t_0=0$ ,  $t_m=1$ . Wenn die Knotenwerte außerdem gleichmäßig verteilt sind (sogenannte **uniform splines**) sieht ein Knotenvektor folgendermaßen aus:

$$\underbrace{\{0, \dots, 0, t_{p+1}, \dots, t_{m-p-1}, 1, \dots, 1\}}_{p+1} \quad \underbrace{\hspace{10em}}_{p+1}$$

für  $n=7$  Kontrollpunkte und einem Polynomgrad von  $p=3$ :

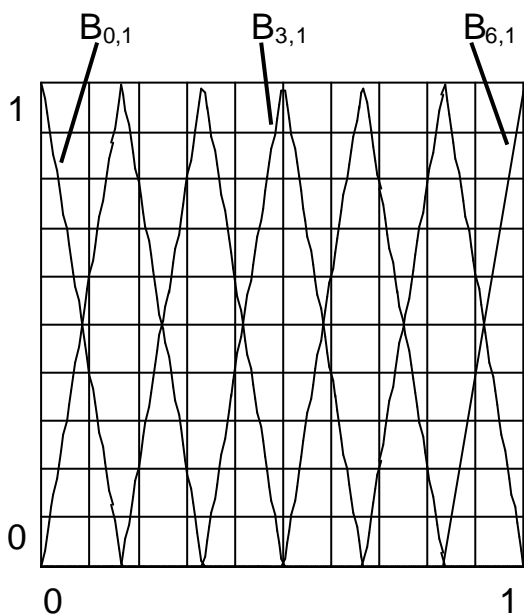
$$\{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$$

Wie sehen nun die B-Spline Basisfunktionen aus? Abbildung 18 zeigt die B-Spline Basisfunktionen für den Polynomgrad 3 bei 7 Kontrollpunkten, also für einen Knotenvektor  $\{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$  (uniform b-spline).



**Abb. 18: B-Spline-Basisfunktionen für den Polynomgrad 3 (7 Kontrollpunkte)**

Abbildung 19 zeigt die Basisfunktionen vom Grad 1 für 7 Kontrollpunkte. Der Knotenvektor lautet:  $\{0,0,1/6,2/6,3/6,4/6,5/6,1,1\}$ .



**Abb. 19: B-Spline-Basisfunktionen für den Polynomgrad 1 (7 Kontrollpunkte)**

Würde man die Basisfunktionen für den Polynomgrad 0 aufzeichnen, wäre in dieser Darstellung nichts zu sehen. Jede Basisfunktion wäre eine Sprungfunktion mit dem Wert null oder dem Wert 1. Der Verlauf fiel also bis auf die senkrechten Linien mit der Unter- bzw. der Oberkante des Diagramms zusammen. Die Berechnung dieses Satzes von Basisfunktionen entspräche dem Vollzug des ersten Berechnungsschrittes höherer Basisfunktionen (vergl. vorhergehende Seite).

Ein Programm zur Berechnung und Darstellung der B-Spline-Basisfunktionen ist bei mir im Quellcode erhältlich.

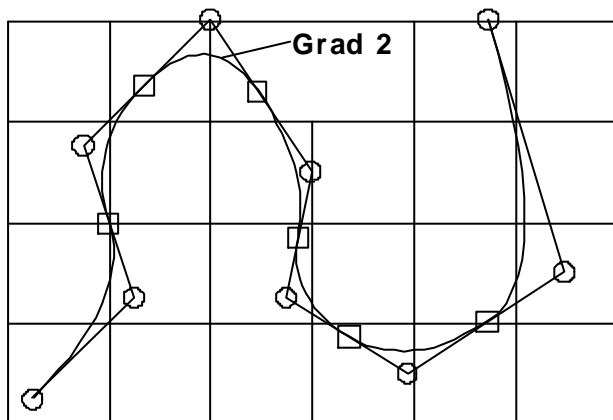
### **B-Spline-Basisfunktionen haben folgende wichtige Eigenschaften:**

- Sie sind stückweise polynomiale Funktionen.
- Sie sind nie negativ:  $B_{i,j} = 0$  für alle  $i, p$  und  $t$

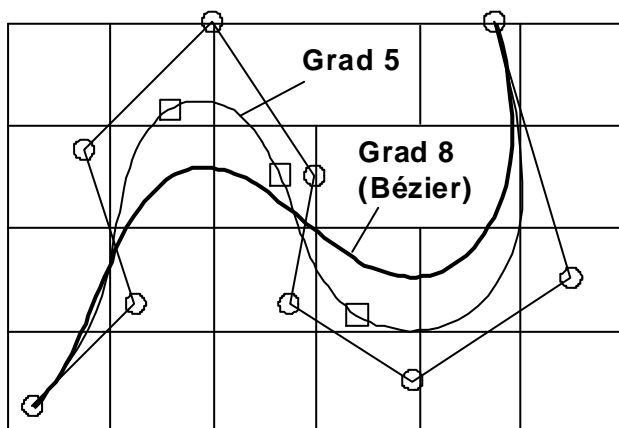
- Ihre Summe ist 1 für jeden Wert  $t$  aus dem Bereich des Knotenvektors:  $\sum_{i=0}^n B_{i,p}(t) = 1$
- sie haben lokal begrenzten Einfluß:  $B_{i,p}(t)=0$  für jeden Wert von  $t$  außerhalb des Intervalls  $[t_i, t_{i+p+1}]$ . Weiterhin sind in jedem beliebigen Intervall des Knotenvektors höchstens  $p+1$  Basispolynome ungleich 0. Dies ist der Grund für die lokale Änderbarkeit. Der Einfluß eines Kontrollpunktes ist damit auf wenige Basispolynome beschränkt.
- Im Bereich einer Knotenspanne (in dem die Funktion ein Polynom ist) existieren alle Ableitungen. An einem Knoten ist  $B_{i,p}(t)$   $p-j$  mal differenzierbar, wobei  $j$  die Vielfachheit eines Knotens ist (ist dann nach obiger Definition kein „uniform spline“ mehr).
- Außer im Fall  $p=0$  hat jede Basisfunktion genau ein Maximum

Nun aber zu den eigentlichen Spline-Kurven. Die Abbildungen 20 bis 22 zeigen B-Splines mit folgenden Merkmalen:

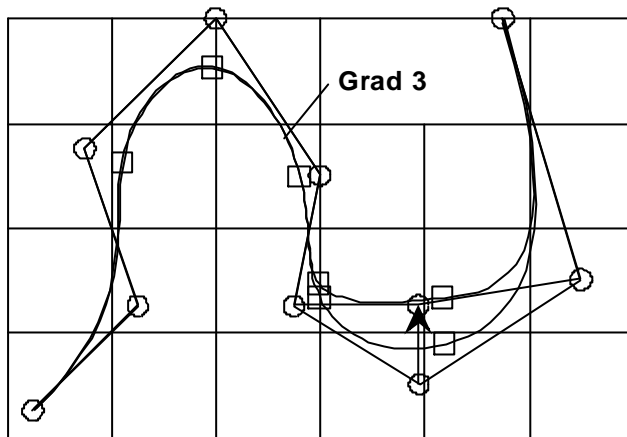
- Alle haben ein Kontrollpolygon aus 9 Punkten. die runden Kreise markieren die Punkte des Kontrollpolygons.
- Die kleinen Quadrate markieren diejenigen Parameterwerte, die den von 0 und 1 verschiedenen Knoten des Knotenvektors entsprechen.
- Der gewählte Polynomgrad ist im Bild angegeben.



**Abb. 20: B-Spline vom Grad 2 mit Kontrollpolygon (Kreise) und Knoten (Quadrate)**



**Abb. 21: B-Spline vom Grad 5 mit Kontrollpolygon (Kreise) und Knoten (Quadrate), sowie vom Grad 8 (entspricht der zum Kontrollpolygon errechneten Bézier-Kurve)**



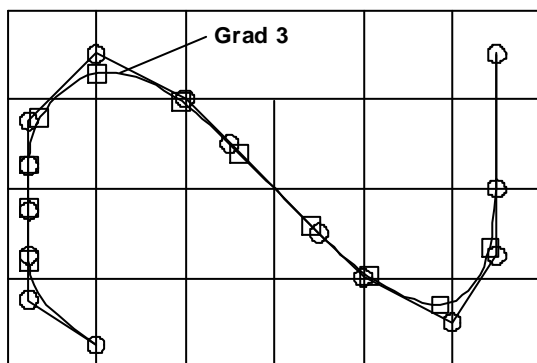
**Abb. 22: Auswirkung der Verschiebung eines Kontrollpunktes (Pfeil) an einem B-Spline vom Grad 3.**

**Kontrollpolygon: Kreise**  
**Knoten: Quadrate**

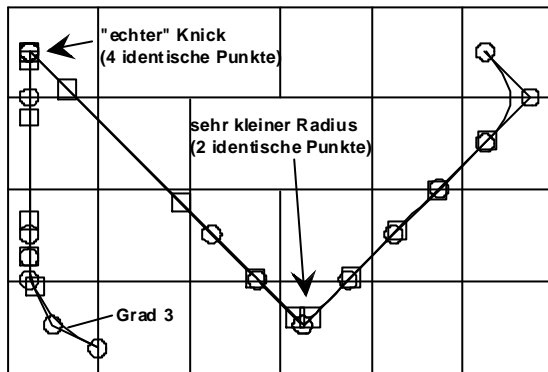
Beim Betrachten der Abbildungen werden folgende Eigenschaften der B-Splines deutlich:

- Wie bei den Bézier-Kurven liegt auch die B-Spline-Kurve innerhalb der konvexen Hülle des durch die Kontrollpunkte aufgespannten Polygons.
- Anfangs- und Endpunkt der Kurven fallen mit Anfangs- und Endpunkt des Kontrollpolygons zusammen.
- Die Tangentenrichtung der Kurve an den Endpunkten stimmt mit der Richtung des Polygons an den Endpunkten überein.
- Je niedriger der Grad der B-Spline-Kurve, desto enger folgt sie dem Kontrollpolygon (vergl. Abb. 20 und Abb. 21). Wählt man den Grad 1 erhält man als Ergebnis das Kontrollpolygon. In diesem Fall fallen Kontrollpunkte und Knotenpunkte zusammen.
- Wählt man den Grad der B-Spline-Basispolynome um eins kleiner als die Anzahl der Kontrollpunkte, so ergibt sich genau die Bézierkurve (vergl. Abb. 22). Das lässt sich natürlich auch theoretisch begründen. Der Knotenvektor sieht dann so aus:  $\{0, \dots, 0, 1, \dots, 1\}$ .
- Die Verschiebung eines Kontrollpunktes wirkt sich nur auf  $p+1$  Bereiche der Kurve aus, wobei  $p$  der gewählte Polynomgrad ist. Daraus resultiert die lokale Änderbarkeit der B-Spline-Kurven. Je kleiner der Grad, desto enger der Bereich, in dem sich Änderungen auswirken. Bei der Bezierkurve ( $p+1$ =Anzahl der Stützstellen) kann also keine lokal begrenzte Änderung mehr möglich sein.

Weitere Eigenschaften der B-Splines sind in den Abbildungen 23 und 24 dokumentiert.



**Abb. 23: Gerade Bereiche in einem B-Spline durch kollineare Kontrollpunkte**



**Abb. 24: Knicke in einem B-Spline durch identische Kontrollpunkte**

Man kann in einer B-Splinekurve gerade Bereiche und/oder Knicke erzeugen. Da die Kurve den Punkten des Kontrollpolygons folgt, kann man sich vorstellen, daß man ihr einen sehr engen Radius aufzwingt, wenn man zwei identische Punkte setzt. Genau genommen gilt:

- Liegen mindestens  $p+1$  Punkte des Kontrollpolygons kollinear, so weist die Kurve in diesem Bereich ein gerades Teilstück auf.
- Liegen mindestens  $p+1$  Punkte übereinander (identische Punkte), so verläuft die Kurve **durch** diese Punkte. Wenn dann die Nachbarpunkte nicht kollinear liegen, erzeugt man dadurch einen Knick.

Ein Programm, mit dem die hier dargestellten B-Spline-Kurven erzeugt wurden, ist bei mir im Quellcode erhältlich.

#### 2.4.4 Non Uniform Rational B-Splines

Die bisher besprochenen Splines waren die sogenannten "uniform splines", bei denen die Knoten des Knotenvektors (mit Ausnahme der Mehrfachknoten am Rand) gleichmäßig verteilt sind. Allerdings gibt es auch Typen von Knotenvektoren bei denen das nicht der Fall ist und mit denen Varianten des Verfahrens abgeleitet wurden. "Non Uniform" Knotenvektoren beeinflussen natürlich die Basisfunktionen und erlauben eine bessere Formsteuerung, so daß die Anpaßbarkeit der Splinekurve an bestimmte Vorgaben verbessert wird. So haben sich heute Verfahrensvarianten herausgebildet, die unter der Bezeichnung NUBS (non uniform b-splines) oder NURBS (non uniform rational b-splines) bekannt sind.

Prinzipiell lassen sich alle hier besprochenen Approximationsverfahren auch in rationaler Form, also aus einem Zähler- und einem Nennerpolynom, definieren:

$$Q(t) = \frac{a_0 + a_1 t + a_2 t^2 + \dots + a_n t^n}{b_0 + b_1 t + b_2 t^2 + \dots + b_n t^n}$$

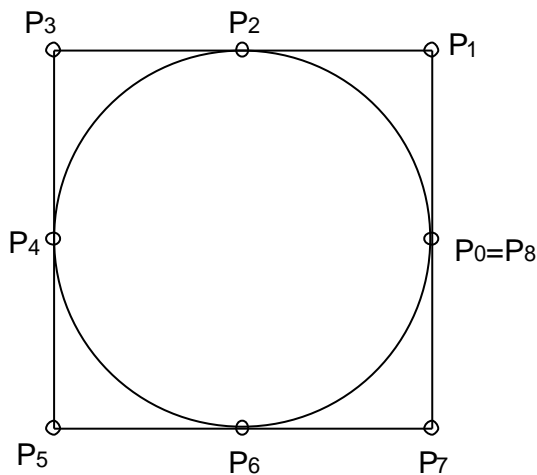
Die rationale Darstellung bietet den Vorteil einer größeren Flexibilität der Kurven bei gleichem Grad. Kombiniert man dies mit den Vorteilen der B-Splines, so läßt sich das rationale B-Spline formulieren (hier nur für die x-Koordinate):



$$Q(t) = \frac{w_0 x_0 B_{0,p}(t) + w_1 x_1 B_{1,p}(t) + \dots + w_n x_n B_{n,p}(t)}{w_0 B_{0,p}(t) + w_1 B_{1,p}(t) + \dots + w_n B_{n,p}(t)} = \frac{\sum_{i=0}^n w_i x_i B_{i,p}(t)}{\sum_{i=0}^n w_i B_{i,p}(t)}$$

Die  $w_i$  sind die sogenannten "Gewichte", die je nach ihrem Betrag die Kontrollpunkte mehr oder weniger betonen, also die Kurve mehr oder weniger an das Kontrollpolygon „heranziehen“.

NURBS werden in zunehmendem Maße Grundlage der (einheitlichen) Geometriebeschreibung in CAD-Systemen. Sie bieten den Vorteil, daß nicht mehr zwischen kanonisch beschreibbaren Geometrieobjekten und Freiform-Geometrien unterschieden werden muß, da man mit Hilfe von NURBS auch die kanonischen Formen bei geeigneter Wahl von Knotenvektoren und Gewichten exakt abbilden kann.



**Abb. 25: Kreisdarstellung durch NURBS**

Abbildung 25 zeigt die Darstellung eines Kreises mit Hilfe von non uniform rational b-splines vom Grad 2. Das Kontrollpolygon hat 9 Knoten. Folglich lautet die Darstellung:

$$Q(t) = \frac{\sum_{i=0}^8 w_i P_i B_{i,p}(t)}{\sum_{i=0}^8 w_i B_{i,p}(t)} \quad \text{wobei } P_i \text{ für die jeweilige Koordinate eines Kontrollpunktes steht.}$$

Der Knotenvektor ist non uniform, da er Mehrfachpunkte aufweist:

$$\left\{ 0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1 \right\} \quad \text{Knotenvektor}$$

$$\left( 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1 \right) \quad \text{Gewichte}$$

Der Faktor  $\frac{\sqrt{2}}{2}$  resultiert aus dem Wert für  $\cos(45^\circ)$ .

## 2.5 Flächendarstellungen

### 2.5.1 Flächendarstellung nach Ferguson

Ausgehend von der parametrischen Standarddarstellung einer Kurve  $x=f_1(t)$ ,  $y=f_2(t)$  kann man sich die Koeffizienten  $a_i$  als Funktionen eines zweiten Parameters  $s$  vorstellen:

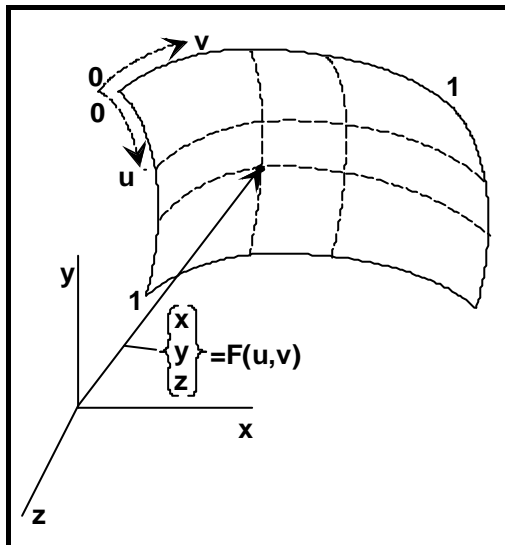
$$x(t) = \sum_{i=0}^n a_i t^i \quad \text{ebenso für } y(t) \text{ und ggf. } z(t). \quad \text{Für } a_i \text{ soll nun gelten: } a_i(s) = \sum_{k=0}^m b_{i,k} s^k$$

Damit hat man bereits die parametrische Form einer polynomialen Flächendarstellung gefunden. Allerdings hat sich bei Flächen für die Parameter die Bezeichnung  $u$  und  $v$  eingebürgert:

$$x(u, v) = \sum_{i=0}^n \sum_{k=0}^m b_{i,k} u^i v^k$$

$$y(u, v) = \sum_{i=0}^n \sum_{k=0}^m c_{i,k} u^i v^k$$

$$z(u, v) = \sum_{i=0}^n \sum_{k=0}^m d_{i,k} u^i v^k$$



**Abb. 26: Flächenstück im Raum  
beschrieben durch die beiden  
unabhängigen Parameter u und v**

Setzt man in obiger Gleichung  $n=m=3$ , so erhält man die Flächendarstellung nach Ferguson. Analog zur Kurvendarstellung bestimmte Ferguson die Koeffizienten durch Vorgabe der 4 Eckpunkte und der Ableitungen an den Eckpunkten in  $u$ - und in  $v$ -Richtung. Das sind 12 Bedingungen. Benötigt werden jedoch 16 Bedingungen, da  $4 \cdot 4 = 16$  Koeffizienten vorhanden sind. Als weitere Bedingungen wurden daher die gemischten Ableitungen an den Eckpunkten vorgegeben (sogenannte Twistvektoren). Allerdings ist ihre geometrische Bedeutung anschaulich schwer zu beschreiben. Sie werden deshalb oft einfach gleich 0 gesetzt.

Wie bei den Kurven ist der Nachteil der Flächenbeschreibung nach Ferguson die Unanschaulichkeit der Koeffizienten. Man verwendet daher auch bei den Flächen spezielle Basispolynome um Anschaulichkeit zu erreichen und eine leichtere Berechenbarkeit der Koeffizienten.

### 2.5.2 Bézier-Patches

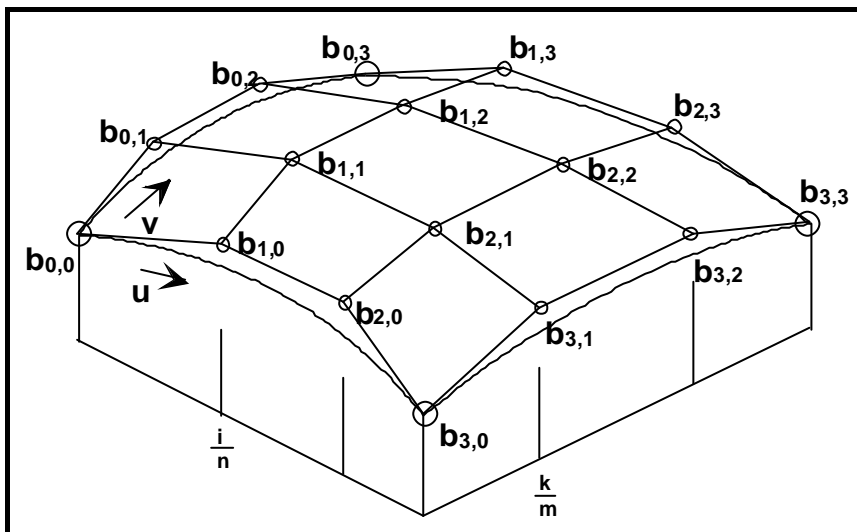
Ersetzt man in den Flächengleichungen nach Ferguson die polynomiale Basis durch die Bernsteinpolynome, so erhält man die Flächendarstellung nach Bézier.

$$x(u,v) = \sum_{i=0}^n \sum_{k=0}^m x_{i,k} B_{k,m}(u) B_{i,n}(v)$$

$$y(u,v) = \sum_{i=0}^n \sum_{k=0}^m y_{i,k} B_{k,m}(u) B_{i,n}(v)$$

$$z(u,v) = \sum_{i=0}^n \sum_{k=0}^m z_{i,k} B_{k,m}(u) B_{i,n}(v)$$

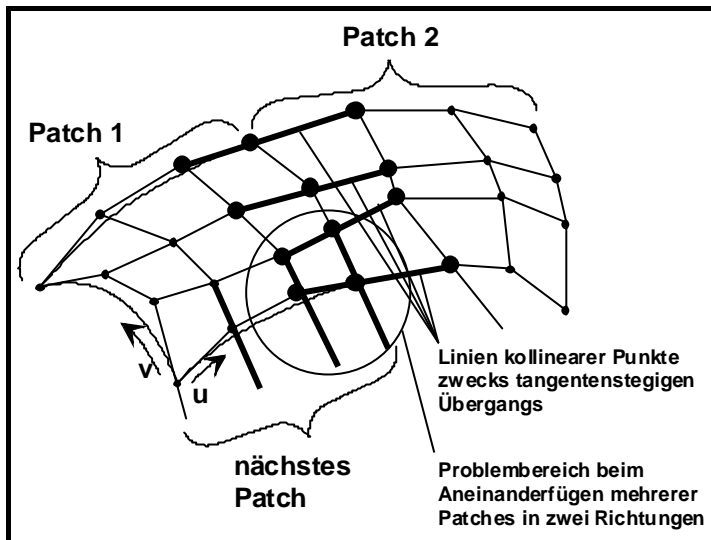
Die  $x_{i,k}$ ,  $y_{i,k}$  und  $z_{i,k}$  sind die Koordinaten der Bézierpunkte. Sie bilden analog zur Kurve das charakteristische Polygonnetz, das Béziernetz. Gemeinsame Punkte zwischen Béziernetz und dem definierten Flächenstück (dem sogenannten Bezier-Patch) sind die 4 Eckpunkte mit den Parameterwerten (0,0); (0,1); (1,0) und (1,1). Die  $b_{i,j}$  in Abbildung 27 stehen also stellvertretend für alle 3 Koordinaten.



**Abb. 27: Bikubisches Bézierpatch**

Durch Änderung eines inneren Bézierpunktes wird zwar das ganze Patch beeinflusst, die Randkurven, das sind die Isoparametrischen, bei denen einer der beiden Parameter entweder den konstanten Wert 1 oder 0 hat, bleiben jedoch erhalten.

Auch für die Flächen gilt, daß die Tangenten an den Rändern lediglich von den Randpunkten des Netzes und den jeweils nächsten Punkten abhängen. Will man also mehrere Patches zu einer größeren Fläche tangentialstetig zusammenfügen, muß man für Kollinearität der entsprechenden Punkte benachbarter Netze sorgen (vergl. Abb. 28).



**Abb. 28: Kollinearität der Punkte von Béziernetzen benachbarter Patches bei tangentialem Übergang**

Oft müssen Patches nicht nur in einer Richtung aneinandergesetzt werden. Hat man zwei Patches tangential stetig verbunden und will in der anderen Richtung weitere Patches tangential stetig anfügen, so widerspricht die Forderung nach Kollinearität der entsprechenden Randpunkte der bereits hergestellten Kollinearität in den Netzen der anderen Patches. Der betreffende Bereich ist in Abbildung 28 durch den Kreis gekennzeichnet. Das tangential stetige Aneinanderfügen vieler Patches ist daher oft ein mühsames Unternehmen. In vielen Fällen bleibt nichts anderes übrig, als den Polynomgrad der Patches zu erhöhen, um die Flexibilität zu vergrößern. Dabei können die Polynomgrade innerhalb eines Patches in beiden Richtungen unterschiedlich sein. Damit sind natürlich auch die Netze in beiden Richtungen ganz unterschiedlich mit Punkten besetzt. Einen Extremfall bilden die Regelflächen, die einer Richtung den Grad 1 haben.

### 2.5.3 B-Spline Flächen

B-Spline-Flächen machen beim Aneinanderfügen mehrerer Patches meist etwas weniger Schwierigkeiten. Zumindest ist es oft von Vorteil, daß der Polynomgrad nicht an die Zahl der Kontrollpunkte gekoppelt ist. Für das Herstellen tangential stetiger Übergänge zwischen Patches gilt allerdings bezüglich der Kollinearität der randnahen Punkte das gleiche wie bei den Bézierflächen. Die Gleichungen der B-Spline-Flächen lauten:

$$x(u, v) = \sum_{i=0}^n \sum_{k=0}^m x_{i,k} B_{k,p}(u) B_{i,q}(v)$$

$$y(u, v) = \sum_{i=0}^n \sum_{k=0}^m y_{i,k} B_{k,p}(u) B_{i,q}(v)$$

$$z(u, v) = \sum_{i=0}^n \sum_{k=0}^m z_{i,k} B_{k,p}(u) B_{i,q}(v)$$

Man erhält sie aus der Gleichung der Bézierflächen durch Austauschen der Ausdrücke für die Basispolynome. Hier bezeichnen die Indizes p und q den unabhängigen Grad der Basispolynome in u bzw. in v-Richtung. Selbstverständlich kann auch bei den B-Spline-

Flächen der Grad in beiden Richtungen unterschiedlich sein, daher auch unterschiedlich Indices. Zusätzlich ist hier noch die Zahl der Kontrollpunkte beliebig manipulierbar.

In modernen CAD-Systemen setzt sich aber anscheinend immer mehr das NURBS-Konzept durch. Man kann es sich leisten, da die Rechner immer leistungsfähiger werden, so daß man den erhöhten numerischen Aufwand nicht scheut. Die Gleichungen für die NURBS-Flächen lauten:

$$x(u, v) = \frac{\sum_{i=0}^n \sum_{k=0}^m x_{i,k} w_{i,k}^{(x)} B_{k,p}(u) B_{i,q}(v)}{\sum_{i=0}^n \sum_{k=0}^m w_{i,k}^{(x)} B_{k,p}(u) B_{i,q}(v)}$$

$$y(u, v) = \frac{\sum_{i=0}^n \sum_{k=0}^m y_{i,k} w_{i,k}^{(y)} B_{k,p}(u) B_{i,q}(v)}{\sum_{i=0}^n \sum_{k=0}^m w_{i,k}^{(y)} B_{k,p}(u) B_{i,q}(v)}$$

$$z(u, v) = \frac{\sum_{i=0}^n \sum_{k=0}^m z_{i,k} w_{i,k}^{(z)} B_{k,p}(u) B_{i,q}(v)}{\sum_{i=0}^n \sum_{k=0}^m w_{i,k}^{(z)} B_{k,p}(u) B_{i,q}(v)}$$

## 2.5.4 Coons-Flächen

Zuletzt seien an dieser Stelle noch die nach Coons (1967) benannten Coons-Patches erwähnt.

Basis der Coons-Patches sind die Hermiteschen Polynome (vergl. Kapitel 2.4.1). Daher sind Coons-Patches immer bikubisch. Die Koeffizienten sind (vergl. Kapitel 2.5.1):

- die Ortsvektoren der 4 Ecken
- die ersten Ableitungen nach u in den Ecken
- die ersten Ableitungen nach v in den Ecken
- die gemischten Ableitungen nach u und v in den Ecken (Twistvektoren)

Es müssen nicht so viele Punkte wie bei B-Splines oder Bézier-Flächen vorgegeben werden, da nur die Randpunkte (Interpolationspunkte), aber keine inneren Kontrollpunkte benötigt werden. Berechnungsprogramme ermitteln die Tangenten an den Punkten meist automatisch durch Ableitung der Interpolationspolynome über drei aufeinander folgende Punkte. Manipulationen der Tangentenrichtungen werden dann interaktiv angeboten. Bei der Manipulation der Twistvektoren treten natürlich die bereits bei den Ferguson-Patches erwähnten Schwierigkeiten auf. Daher ermittelt man diese heute oft nach Kriterien aus Glättungsberechnungen.

Die Handhabung von Coons-Patches kann in den NET-Funktionen des an der FH vorhandenen CAD-Systems CATIA ausprobiert werden.