Bézier Splines

Prof. Dr. Dörte Haftendorn, MuPAD 4, http://haftendorn.uni-lueneburg.de Aug.06

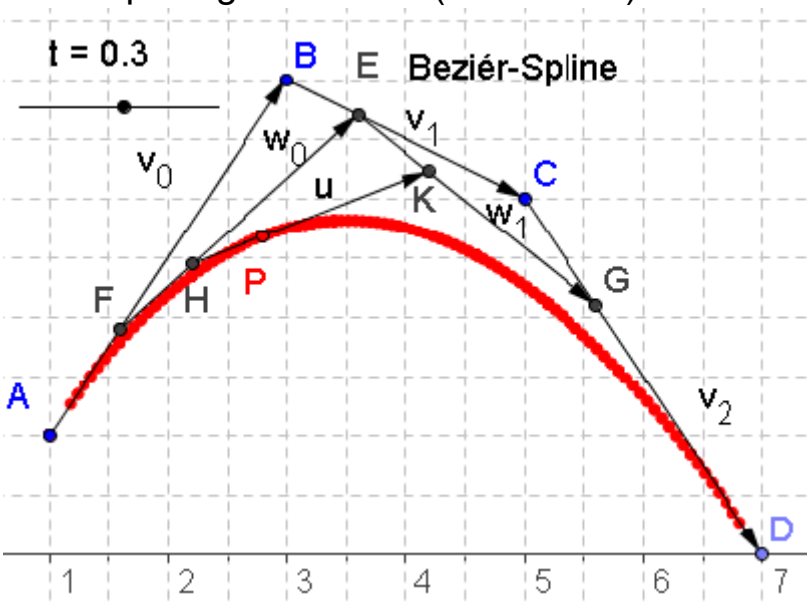
Automatische Übersetzung aus MuPAD 3.11, Nov. 05 Update 21.04.06

Es fehlen nocht textliche Änderungen, die MuPAD 4 direkt berücksichtigen, das ist in Arbeit.

Web: http://haftendorn.uni-lueneburg.de www.mathematik-verstehen.de

++

Bézier-Kurven sind Grundlage für "Kurven-Werkzeuge" in denNov. 05 Update 21.04.06
Malprogrammen am Computer. Mit ihnen werden auch Schriften entworfen.
Die Weiterführung in den Raum dient zum Modellieren z.B. von Autoformen, Architektur-Elementen, zum computerunterstützten Design überhaupt.
Dort lag auch der ursprüngliche Ansatz von Bézier.
Bézierspline geometrisch (GeoGebra)



Von 4 Punkten sind 2 Stützpunkte und 2 Steuerpunkte.
Es werden t-Teilungspunkte gebildet. der letzte ist P, Punkt der Bezér-Kurve.
Unten wird bewiesen, dass bei diesem Konzept 4 Polynome eine wichtige Rolle
spielen, nämlich die Bernstein-Polynome.
Eine Paramterdarstellung der Bézier-Kurve erhält man aus den Daten der vier Punkte sofort als Linearkomination der Bernsteinpolynome.

##
#

Bernsteinpolynome

```
B0:=t->(1-t)^3:        B0(t);
B1:=t->3*t*(1-t)^2:    B1(t);
B2:=t->3*t^2*(1-t):    B2(t);
B3:=t->t^3:            B3(t);
```

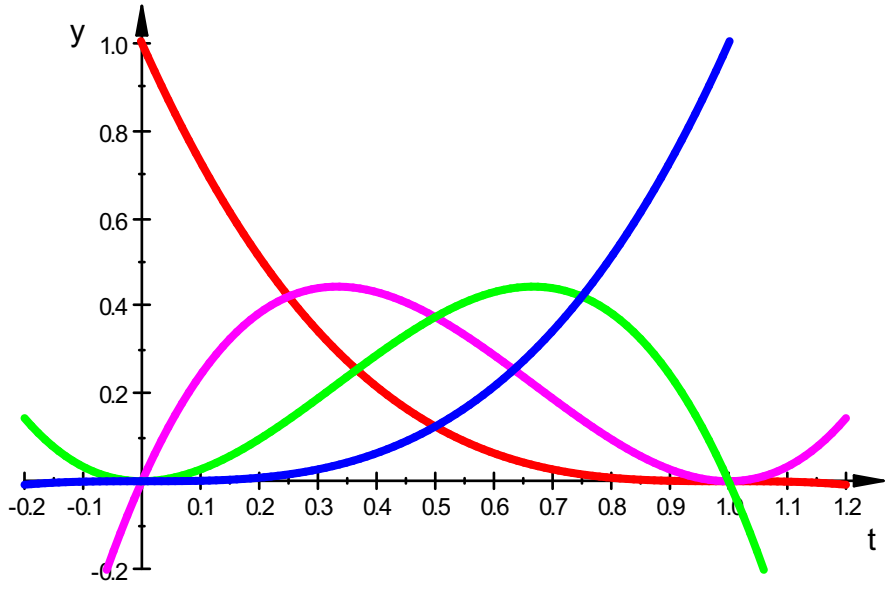
$$-(t-1)^3$$

$$3 \cdot t \cdot (t-1)^2$$

$$-3 \cdot t^2 \cdot (t-1)$$

$$t^3$$

```
grB0:=plot::Function2d(B0(t),t=-1..2, LineWidth=1,LineColor=[1,0,0])
grB1:=plot::Function2d(B1(t),t=-1..2, LineWidth=1,LineColor=[1,0,1])
grB2:=plot::Function2d(B2(t),t=-1..2, LineWidth=1,LineColor=[0,1,0])
grB3:=plot::Function2d(B3(t),t=-1..2, LineWidth=1,LineColor=[0,0,1])
plot(grB0,grB1,grB2,grB3,ViewingBox=[-0.2..1.2,-0.2..1])
```



Die Bernsteinpolynome bilden eine Basis im Vektoraum der Polynome bis zum 3. Grad. Sie sind linear unabhängig:

```
gl0:=subs(a*B0(t)+b*B1(t)+c*B2(t)+d*B3(t)=0,t=0);
gl1:=subs(a*B0(t)+b*B1(t)+c*B2(t)+d*B3(t)=0,t=1);
```

$$a = 0$$

$$d = 0$$

```
gl2:=subs(a*B0(t)+b*B1(t)+c*B2(t)+d*B3(t)=0,[a=0,b=0,t=1/2])
gl3:=subs(a*B0(t)+b*B1(t)+c*B2(t)+d*B3(t)=0,[a=0,b=0,t=1/4])
```

$$\frac{3 \cdot c}{8} + \frac{d}{8} = 0$$

$$\frac{9 \cdot c}{64} + \frac{d}{64} = 0$$

```
solve([gl0,gl1,gl2,gl3],[a,b,c,d])
```

$$\{[a = 0,\ b = z,\ c = 0,\ d = 0]\}$$

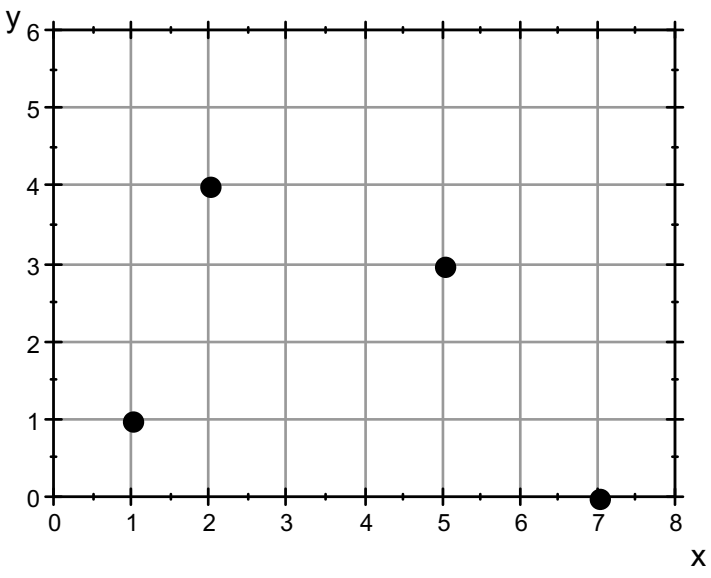
Die Linearkombination soll für alle t 0 ergeben, also kann man vier t-Werte einsetzen. Das führt zu 0 für alle Koeffizienten.
Damit ist die lineare Unabhängigkeit bewiesen.
##

Erzeugung eines Bézier-Splines mit 4 Datenpunkten

```
datenPunkte:=[1,1], [2,4],[5,3],[7,0]:
dp:=datenPunkte;
```

$$[1, 1], [2, 4], [5, 3], [7, 0]$$

```
x0:=dp[1][1]:x1:=dp[2][1]:x2:=dp[3][1]:x3:=dp[4][1]:
y0:=dp[1][2]:y1:=dp[2][2]:y2:=dp[3][2]:y3:=dp[4][2]:
graphDatenPunkte:=plot::Listplot([datenPunkte],
   LinesVisible=FALSE, PointSize=3,Scaling=Constrained,
    GridVisible=TRUE,ViewingBox=[0..8,0..6]):
plot(graphDatenPunkte)
```

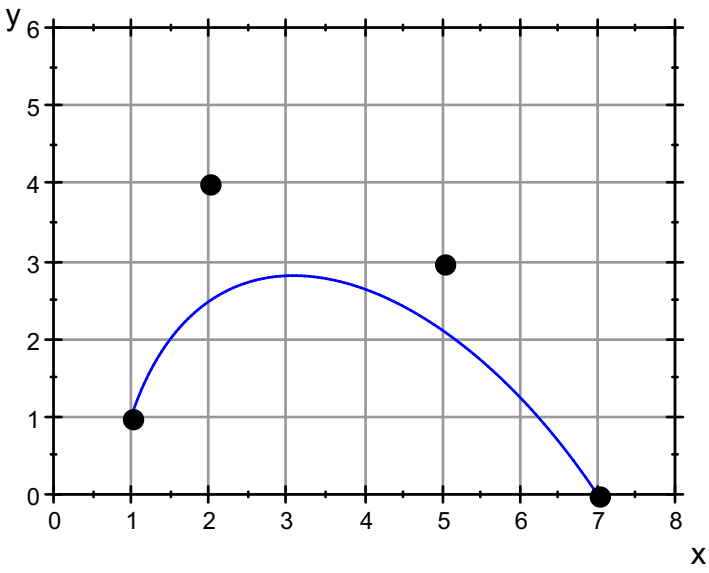


Linearkombination der Bernsteinpolynome mit den Daten

```
xb:=t->x0*B0(t)+x1*B1(t)+x2*B2(t)+x3*B3(t):
yb:=t->y0*B0(t)+y1*B1(t)+y2*B2(t)+y3*B3(t):
[xb(t),yb(t)]
```

$$[7 \cdot t^3 + 6 \cdot t \cdot (t-1)^2 - 15 \cdot t^2 \cdot (t-1) - (t-1)^3,\ 12 \cdot t \cdot (t-1)^2 - 9 \cdot t^2 \cdot (t-1) -$$

```
grbez:=plot::Curve2d([xb(t),yb(t)],t=0..1):
plot(grbez,graphDatenPunkte)
```



##

Herleitung der Bernsteinpolynome und Beweis der Richtigkeit des Ansatzes: A,B,C,D sind die Stützpunkte, alles ist vektoriell zu lesen.

```
v0:=B-A;v1:=C-B; v2:=D-C;
```

$$B - A$$

$$C - B$$

$$D - C$$

```
F:=A+t*v0;Ee:=B+t*v1;G:=C+t*v2;
```

$$A - t \cdot (A - B)$$

$$B - t \cdot (B - C)$$

$$C - t \cdot (C - D)$$

```
w0:=Ee-F;w1:=G-Ee;
```

$$B - A + t \cdot (A - B) - t \cdot (B - C)$$

$$C - B - t \cdot (C - D) + t \cdot (B - C)$$

```
H:=F+t*w0; K:=Ee+t*w1;
```

$$A - t \cdot (A - B) - t \cdot (A - B - t \cdot (A - B) + t \cdot (B - C))$$

$$B - t \cdot (B - C + t \cdot (C - D) - t \cdot (B - C)) - t \cdot (B - C)$$

```
u:=K-H; P:=H+t*u;
```

$$B - A - t\cdot(B - C + t\cdot(C - D) - t\cdot(B - C)) + t\cdot(A - B) - t\cdot(B - C) + t\cdot(A -$$

$$A - t\cdot(A - B) - t\cdot(A - B - t\cdot(A - B) + t\cdot(B - C)) - t\cdot(A - B + t\cdot(B - C +$$

```
P:=expand(P)
```

$$A - 3\cdot A\cdot t + 3\cdot B\cdot t + t^3\cdot D + 3\cdot A\cdot t^2 - A\cdot t^3 - 6\cdot B\cdot t^2 + 3\cdot B\cdot t^3 + 3\cdot C\cdot t^2$$

```
Af:=factor(1+3*t^2-t^3-3*t);
Bf:=factor(-6*t^2+3*t^3+3*t);
Cf:=factor(3*t^2-3*t^3);
Df:=t^3;
probe:=simplify(Af*A+Bf*B+Cf*C+Df*D-P);
```

$$-(t-1)^3$$

$$3\cdot t\cdot(t-1)^2$$

$$-3\cdot t^2\cdot(t-1)$$

$$t^3$$

$$0$$

Hier sind die Terme der Bernsteinpolynome. !!!! (1-t)=-(t-1) !!!!
Damit ist gezeigt, dass die Ortskurve der Stützkonstruktion tatsächlich als Linearkombination der Bernsteinpolynome erhalten werden kann.
Die Koeffizienten sind die Stützstellen, b.z.w. die Stützwerte.
##

Beweis, dass B und C auf den Randtangenten liegen.

Erst im Beispiel:

```
expand(xb'(t));expand(yb'(t))
```

$$-9\cdot t^2 + 12\cdot t + 3$$

$$6\cdot t^2 - 24\cdot t + 9$$

```
abl:=t->expand(yb'(t))/expand(xb'(t)):abl(t)
```

$$\frac{6\cdot t^2 - 24\cdot t + 9}{-9\cdot t^2 + 12\cdot t + 3}$$

```
abl(0),abl(1)
```

$$3,\ -\frac{3}{2}$$

Diese Steigungen kann man unmittelbar ablesen.

##

Allgemein

```
ableit:=diff(P,t)
```

$$3\cdot B-3\cdot A+6\cdot A\cdot t-12\cdot B\cdot t+6\cdot C\cdot t+3\cdot t^2\cdot D-3\cdot A\cdot t^2+9\cdot B\cdot t^2-9\cdot$$

```
subs(ableit,t=0);
subs(ableit,t=1);
```

$$3\cdot B-3\cdot A$$

$$3\cdot D-3\cdot C$$

Die Ableitungen haben also die Richtungen der Punktdifferenzen an den Ecken.

In Normaldarstellung delta y / delta x

```
3*(y1-y0)/(3*(x1-x0))
```

$$3$$

```
3*(y3-y2)/(3*(x3-x2))
```

$$-\frac{3}{2}$$

##

Beweis, dass die äußeren Ecken werden erreicht werden:

```
[xb(0),yb(0)], [xb(1), yb(1)]
```

$$[1,\ 1],\ [7,\ 0]$$

```
subs(P,t=0), subs(P,t=1)
```

$$A,\ D$$