

Kubische Splines

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

Automatische Übersetzung aus MuPAD 3.11, Nov. 05 Update 11.11.05

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

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

+++++

Konstruiert für n Datenpunkte, die hier beliebig eingegeben werden können.

Eine Anpassung der Zeichenbereiche ist ggf. "von Hand" nötig".

```
datenPunkte := [0,0], [2,1], [3,2], [6,2] :
```

```
dp := datenPunkte ;
```

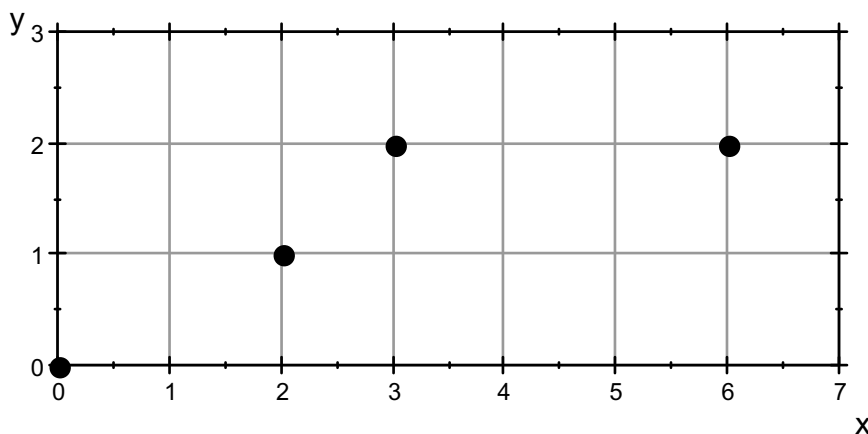
```
[0, 0], [2, 1], [3, 2], [6, 2]
```

```
graphDatenPunkte := plot::Listplot([datenPunkte],
```

```
    LinesVisible=FALSE, PointSize=3, Scaling=Constrained,
```

```
    GridVisible=TRUE, ViewingBox=[0..7,0..3]) :
```

```
plot(graphDatenPunkte)
```



Zuerst müssen die Datenpunkte in einer x-Datenliste und einer y-Datenliste aufgenommen werden.

```
anz := nops(dp) :
```

```
xd := [dp[i][1] $i=1..anz] ;
```

```
yd := [dp[i][2] $i=1..anz] ;
```

```
[0, 2, 3, 6]
```

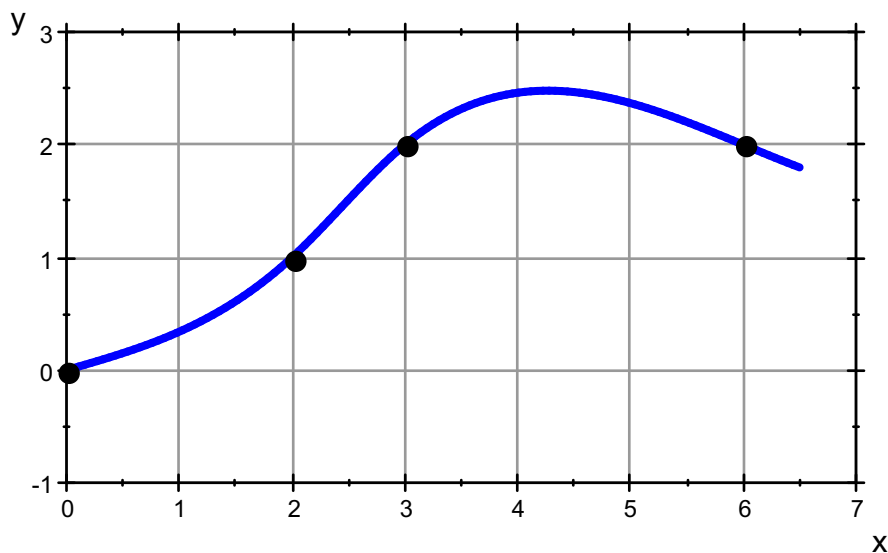
```
[0, 1, 2, 2]
```

```
spline := numeric::cubicSpline(dp, Natural) :
```

```
spline(4)
```

2.460992908

```
grspline:=plot::Function2d(spline(x),x=0..6.5, LineWidth=1):  
plot(grspline,graphDatenPunkte,  
      ViewingBox=[0..7,-1..3])
```

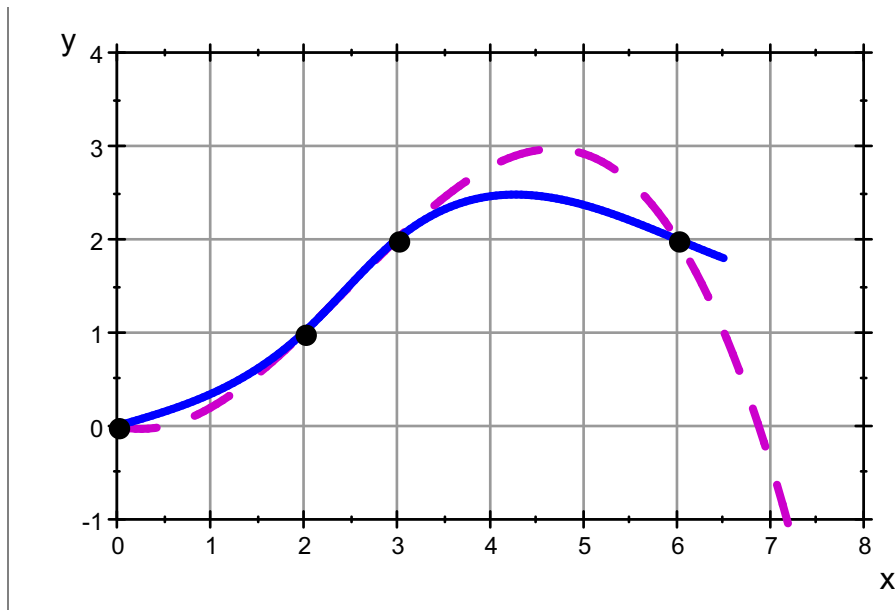


Zum Vergleich sei das Interpolationspolynom gezeichnet:

```
ip:=interpolate(xd,yd,x);  
gripol:=plot::Function2d(ip(x),x=0..8,  
      LineColor=[0.8,0,0.8], LineWidth=1, LineStyle=Dash)
```

$$\text{poly}\left(-\frac{5 \cdot x^3}{72} + \frac{37 \cdot x^2}{72} - \frac{x}{4}, [x]\right)$$

```
plot(gripol,grspline,graphDatenPunkte,  
      ViewingBox=[0..8,-1..4])
```



#####

Eigene Splineberechnung

```
p0 := x -> a0 + b0 * (x - dp[1][1]) + c0 * (x - dp[1][1])^2 + d0 * (x - dp[1][1])^3
p1 := x -> a1 + b1 * (x - dp[2][1]) + c1 * (x - dp[2][1])^2 + d1 * (x - dp[2][1])^3
p2 := x -> a2 + b2 * (x - dp[3][1]) + c2 * (x - dp[3][1])^2 + d2 * (x - dp[3][1])^3
```

$$d0 \cdot x^3 + c0 \cdot x^2 + b0 \cdot x + a0$$

$$a1 + b1 \cdot (x - 2) + c1 \cdot (x - 2)^2 + d1 \cdot (x - 2)^3$$

$$a2 + b2 \cdot (x - 3) + c2 \cdot (x - 3)^2 + d2 \cdot (x - 3)^3$$

```
pktli := [ p0(dp[1][1]) = dp[1][2],
           p1(dp[2][1]) = dp[2][2],
           p2(dp[3][1]) = dp[3][2] ]; // Punkte links
pktre := [ p0(dp[2][1]) = dp[2][2],
           p1(dp[3][1]) = dp[3][2],
           p2(dp[4][1]) = dp[4][2] ]; // Punkte rechts
abl := [ p0'(dp[2][1]) = p1'(dp[2][1]),
         p1'(dp[3][1]) = p2'(dp[3][1]) ]; // Ableitungen
kr := [ p0''(dp[2][1]) = p1''(dp[2][1]),
        p1''(dp[3][1]) = p2''(dp[3][1]) ]; // Krümmungen
nat := [ p0''(dp[1][1]) = 0, p2''(dp[4][1]) = 0 ]; // Natürliche Beding
```

$$[a0 = 0, a1 = 1, a2 = 2]$$

$$[a0 + 2 \cdot b0 + 4 \cdot c0 + 8 \cdot d0 = 1, a1 + b1 + c1 + d1 = 2, a2 + 3 \cdot b2 + 9 \cdot c2 + 27 \cdot d2 = 2]$$

$$[b0 + 4 \cdot c0 + 12 \cdot d0 = b1, b1 + 2 \cdot c1 + 3 \cdot d1 = b2]$$

$$[2 \cdot c_0 + 12 \cdot d_0 = 2 \cdot c_1, 2 \cdot c_1 + 6 \cdot d_1 = 2 \cdot c_2]$$

$$[2 \cdot c_0 = 0, 2 \cdot c_2 + 18 \cdot d_2 = 0]$$

Gleichungssystem mit 12 Variablen und 12 Gleichungen

```
lo:=op(solve(pktli.pktre.abl.kr.nat))
```

$$\left[a_0 = 0, a_1 = 1, a_2 = 2, b_0 = \frac{27}{94}, b_1 = \frac{87}{94}, b_2 = \frac{39}{47}, c_0 = 0, c_1 = \frac{15}{47}, c_2 = - \right.$$

```
tp0:=subs(p0(x), lo);
```

```
tp1:=subs(p1(x), lo);
```

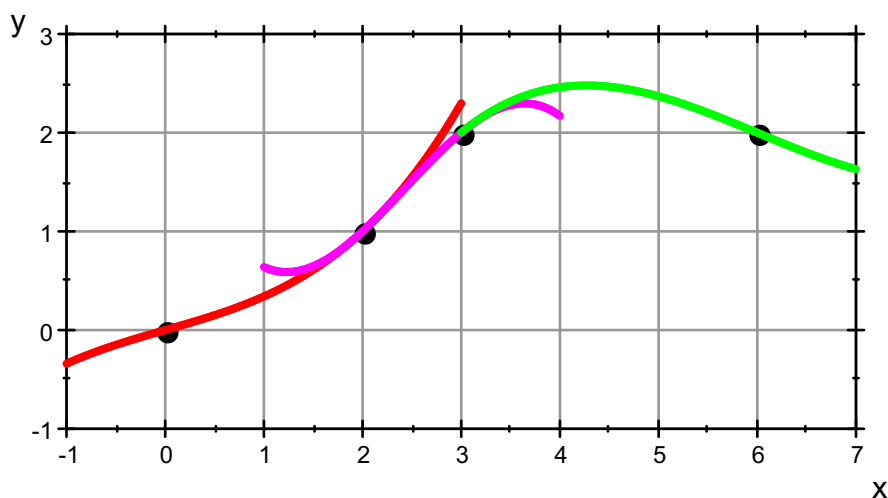
```
tp2:=subs(p2(x), lo);
```

$$\frac{5 \cdot x^3}{94} + \frac{27 \cdot x}{94}$$

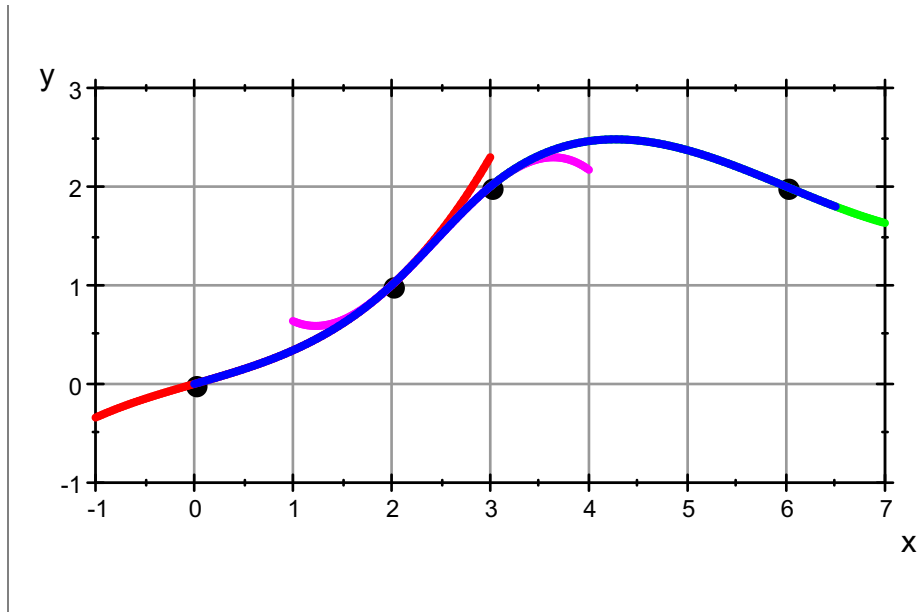
$$\frac{87 \cdot x}{94} + \frac{15 \cdot (x-2)^2}{47} - \frac{23 \cdot (x-2)^3}{94} - \frac{40}{47}$$

$$\frac{39 \cdot x}{47} - \frac{39 \cdot (x-3)^2}{94} + \frac{13 \cdot (x-3)^3}{282} - \frac{23}{47}$$

```
grp0:=plot::Function2d(tp0,x=-1..3,LineWidth=1,LineColor=|
grp1:=plot::Function2d(tp1,x=1..4,LineWidth=1,LineColor=[1,
grp2:=plot::Function2d(tp2,x=3..7,LineWidth=1,LineColor=[0,
plot(graphDatenPunkte,grp0,grp1,grp2,
      ViewingBox=[-1..7,-1..3],Scaling=Constrained,GridVisib
```



```
plot(graphDatenPunkte,grp0,grp1,grp2, grspline,
      ViewingBox=[-1..7,-1..3],Scaling=Constrained,GridVisib
```



Numerisch kann man den Spline wie eine Funktion verwenden.

```
numeric::int(spline(x), x=0..6)
```

```
9.228723404
```