Splines

Splines

Jana Ernekerová & Tomáš Kalvoda, 2014

Lineární interpolace

Nejjednodušší a nejčastěji používaná interpolace je lineární a spočívá v proložení dvou sousedních bodů přímkou. Přímka propojující dva body (a, b) a (c, d), kde $a \neq c$ je dána rovnicí:

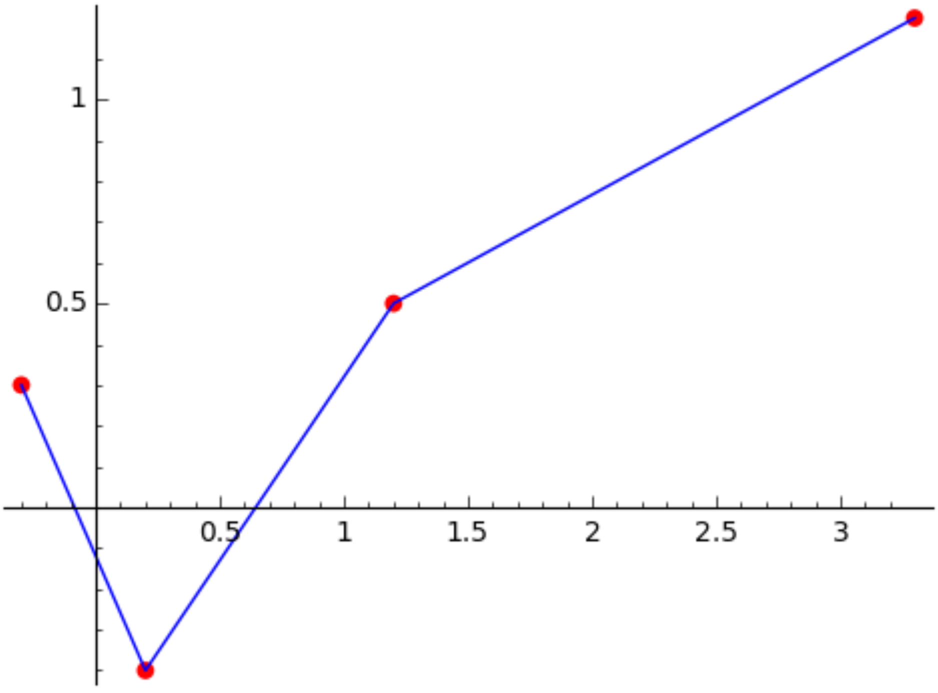
$$y = \frac{d-b}{c-a}(x-a) + b.$$

Příklad

Vytvořím několik bodů a zobrazíme jejich lineární interpolaci.

```
V = RR^2;
A = V([-0.3,0.3]); B = V([0.2,-0.4]); C = V([1.2,0.5]); D =
V([3.3,1.2]);
```

```
picpts = points([A,B,C,D],color='red',pointsize=40)
picline = line([A,B,C,D],color='blue')
(picpts+picline).show(figsize=5)
```



Nelineární interpolace (splines)

Místo spojování bodů přímkami (grafy polynomů prvního stupně) mužeme body spojovat polynomiálními křivkami. Často se používá polynomů třetího stupně (kubická interpolace, splines). Důvodem je, že mají poměrně jednoduchý tvar, avšak už umožňují inflexi (obrat rovinné křivky). Kubický polynom je funkce tvaru

$$f(x) = ax^3 + bx^2 + cx + d.$$

Je tedy jednoznačně zadána čtyřmi konstantami a, b, c, d.

Uvažujme tedy pro jednoduchost množinu čtyř bodů

$$\{\ (x_1, y_1),\ (x_2, y_2), \big(x_3, y_3\big), (x_4, y_4)\big\}$$

Kdyby náhodou dvě x-ové souřadnice byly stejné, pak tyto body v ,,tomto směru'' nelze spojit a je třeba změnit souřadnice.

Hledáme tři kubické polynomy

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i, x \in \langle x_i, x_{i+1} \rangle, i = 1, 2, 3.$$

Požadujeme, aby zadané body spojovaly, tedy máme 6 podmínek na funkční hodnoty

$$f_1(x_1) = y_1, f_1(x_2) = f_2(x_2) = y_2, f_2(x_3) = f_3(x_3) = y_3, f_3(x_4) = y_4$$

Dále požadujeme, aby ve spojích měly stejnou první i druhou derivaci (stejný sklon, tečna a stejná konkávnost/konvexnost), to nám dává další čtyři podmínky

$$f_1'(x_2) = f_2'(x_2), f_2'(x_3) = f_3'(x_3), f_1''(x_2) = f_2''(x_2), f_2''(x_3) = f_3''(x_3)$$

Konečně, můžeme zadat jaký požadujeme sklon na koncích, tedy dvě podmínky

$$f_1(x_1) = A, f_3(x_4) = B.$$

Celkem máme 12 rovnic pro 12 neznámých. Rovnice jsou lineární. Řešením získáme hodnoty 12-ti konstant a pak stačí vykreslit takto vypočtene polynomy na příslušných intervalech.

Příklad

Několik bodů.

```
V = RR^2
A = V([-0.2,0.3]);
B = V([0.5,-0.2]);
C = V([1.2,0.7]);
D = V([2.4,0.1]);
```

Kubická interpolace.

```
var('a1,a2,a3,a4')
var('b1,b2,b3,b4')
var('c1,c2,c3,c4')
var('x')
```

x

```
f1(x) = a1*x^3 + a2*x^2 + a3*x + a4
f2(x) = b1*x^3 + b2*x^2 + b3*x + b4
f3(x) = c1*x^3 + c2*x^2 + c3*x + c4
df1(x) = f1.diff(x)(x)
df2(x) = f2.diff(x)(x)
df3(x) = f3.diff(x)(x)
ddf1(x) = f1.diff(x,2)(x)
ddf2(x) = f2.diff(x,2)(x)
ddf3(x) = f3.diff(x,2)(x)
```

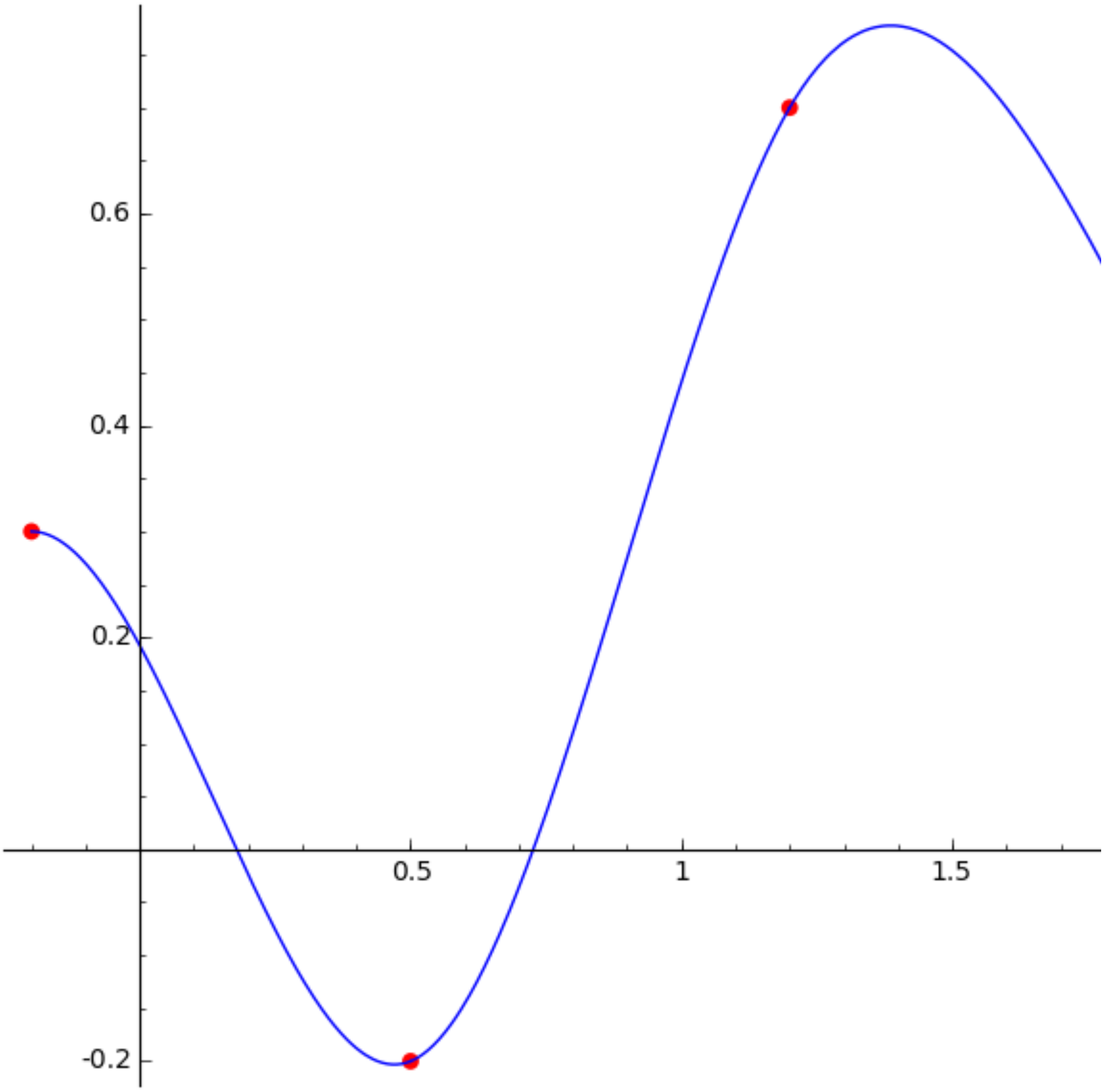
```
coeffs = solve([
f1(A[0]) == A[1],
f1(B[0]) == B[1],
f2(B[0]) == B[1],
f2(C[0]) == C[1],
f3(C[0]) == C[1],
f3(D[0]) == D[1],
df1(B[0]) == df2(B[0]),
df2(C[0]) == df3(C[0]),
ddf1(B[0]) == ddf2(B[0]),
ddf2(C[0]) == ddf3(C[0]),
df1(A[0]) == 0,
df3(D[0]) == 0
],[a1,a2,a3,a4,b1,b2,b3,b4,c1,c2,c3,c4],
solution_dict=True)[0]
```

```
coeffs
```

```
{b3: -137469/24010,
 a3: -11322/12005,
 c2: -725/98,
 b2: 78849/9604,
 a2: -13011/9604,
 c1: 4595/3528,
 b1: -14565/4802,
 c4: -15931/2450,
 a1: 16055/4802,
 b4: 4751/4802,
 a4: 1847/9604,
 c3: 3186/245}
```

```
ff1(x) = coeffs[a1]*x^3 + coeffs[a2]*x^2 + coeffs[a3]*x +
coeffs[a4]
ff2(x) = coeffs[b1]*x^3 + coeffs[b2]*x^2 + coeffs[b3]*x +
coeffs[b4]
ff3(x) = coeffs[c1]*x^3 + coeffs[c2]*x^2 + coeffs[c3]*x +
coeffs[c4]
```

```
picf1 = plot(ff1,A[0],B[0])
picf2 = plot(ff2,B[0],C[0])
picf3 = plot(ff3,C[0],D[0])
picf1+picf2+picf3+points([A,B,C,D],color='red',pointsize=40)
```



Interaktivní ukázka

```
@interact
def _(slopeleft=(-10,10), sloperight=(-10,10)):
    coeffs = solve([
    f1(A[0]) == A[1],
    f1(B[0]) == B[1],
    f2(B[0]) == B[1],
    f2(C[0]) == C[1],
    f3(C[0]) == C[1],
    f3(D[0]) == D[1],
    df1(B[0]) == df2(B[0]),
    df2(C[0]) == df3(C[0]),
    ddf1(B[0]) == ddf2(B[0]),
```

```
    ddf2(C[0]) == ddf3(C[0]),
    df1(A[0]) == slopeleft,
    df3(D[0]) == sloperight
    ],[a1,a2,a3,a4,b1,b2,b3,b4,c1,c2,c3,c4],
solution_dict=True)[0]
    ff1(x) = coeffs[a1]*x^3 + coeffs[a2]*x^2 + coeffs[a3]*x +
coeffs[a4]
    ff2(x) = coeffs[b1]*x^3 + coeffs[b2]*x^2 + coeffs[b3]*x +
coeffs[b4]
    ff3(x) = coeffs[c1]*x^3 + coeffs[c2]*x^2 + coeffs[c3]*x +
coeffs[c4]
    picf1 = plot(ff1,A[0],B[0])
    picf2 = plot(ff2,B[0],C[0])
    picf3 = plot(ff3,C[0],D[0])

show(picf1+picf2+picf3+points([A,B,C,D],color='red',pointsize=40)
```

slopeleft -10.0

sloperight -10.0