Bézierovy křivky

Bézierovy křivky

Tomáš Kalvoda, 2014

Teoretické shrnutí

Pro zadané $n \in \mathbb{N}_0$ a $i = 0, 1, 2, \ldots, n$ definujme Bernsteinovy polynomy předpisem

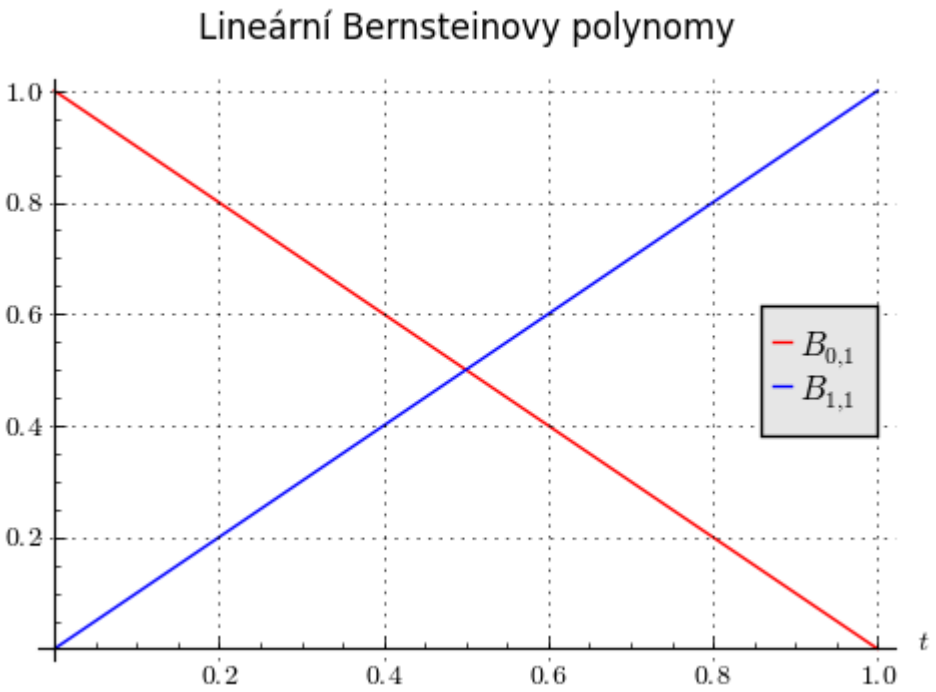
$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i},\ t \in \langle 0, 1 \rangle.$$

Nechť je zadáno $n+1$ (tzv. kontrolních) bodů P_0, $P_1, \ldots,$ P_n. Bézierovou křivkou zadanou těmito body nazýváme křivku danou parametrizací

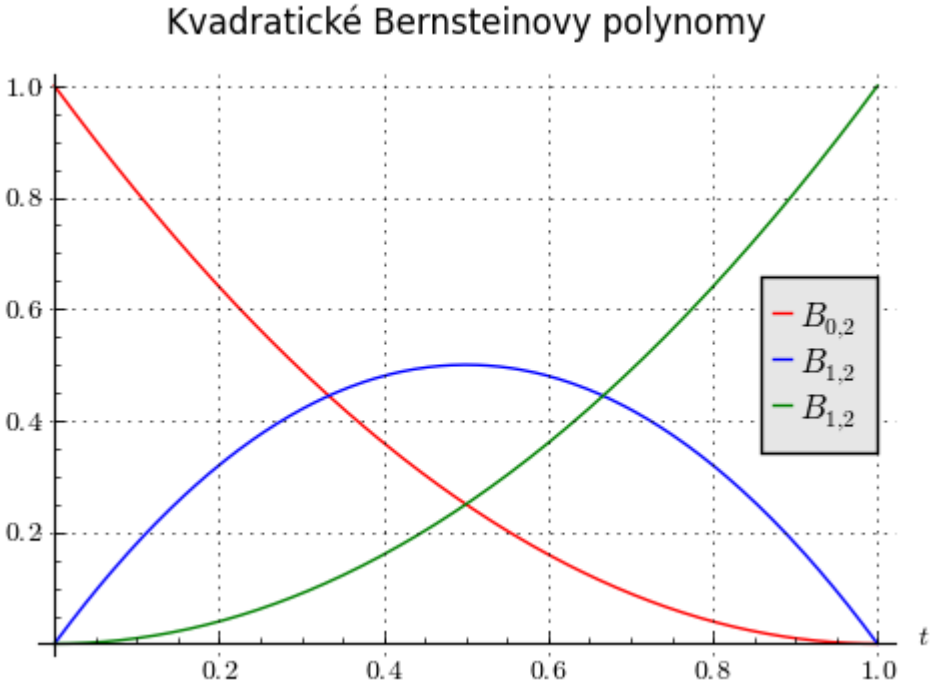
$$C(t) := \sum_{i=0}^{n} P_i B_{i,n}(t), \quad t \in \langle 0, 1 \rangle.$$

```
# Bernsteinův polynom
def bernstein(i,n,t):
    return binomial(n,i) * t^i * (1-t)^(n-i)
```

```
var('t')
fig_b01 = plot(bernstein(0,1,t),(t,0,1),color='red',
legend_label='$B_{0,1}$', figsize=5, tick_formatter='latex',
axes_labels=['$t$',''], gridlines=True, title=u'Lineární
Bernsteinovy polynomy')
fig_b11 = plot(bernstein(1,1,t),(t,0,1), color='blue',
legend_label='$B_{1,1}$')
fig_b01 + fig_b11
```

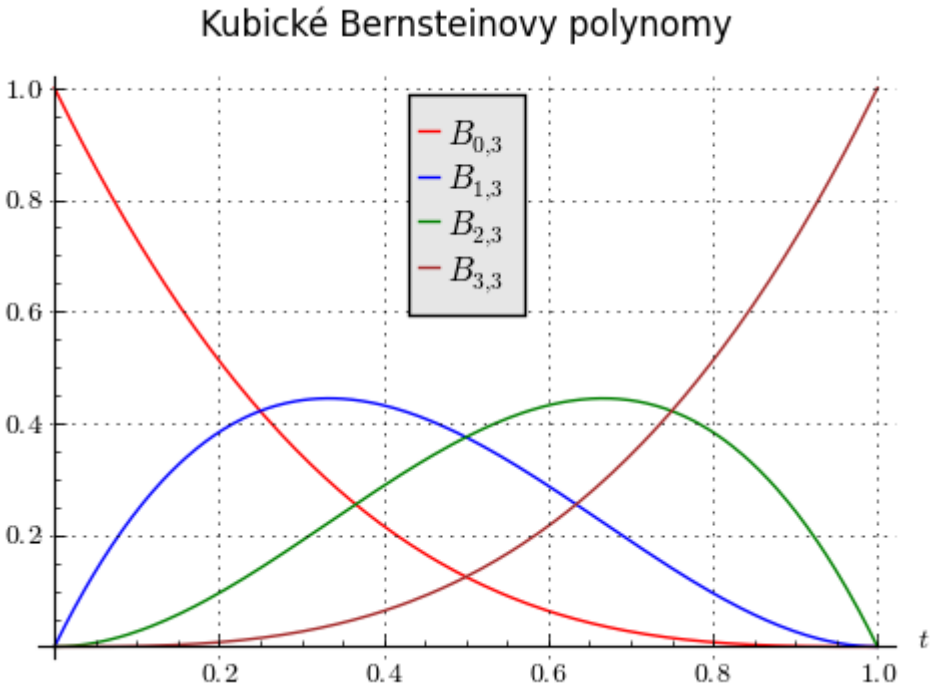



```
fig_b02 = plot(bernstein(0,2,t), (t,0,1), color='red',
legend_label='$B_{0,2}$', figsize=5, tick_formatter='latex',
axes_labels=['$t$',''], gridlines=True, title=u'Kvadratické
Bernsteinovy polynomy')
fig_b12 = plot(bernstein(1,2,t),
(t,0,1),color='blue',legend_label='$B_{1,2}$')
fig_b22 = plot(bernstein(2,2,t),
(t,0,1),color='green',legend_label='$B_{1,2}$')
fig_b02 + fig_b12 + fig_b22
```




```
fig_b03 = plot(bernstein(0,3,t), (t,0,1), color='red',
```

```
legend_label='$B_{0,3}$', figsize=5, tick_formatter='latex',
axes_labels=['$t$',''], gridlines=True, title=u'Kubické
Bernsteinovy polynomy')
fig_b13 = plot(bernstein(1,3,t),
(t,0,1),color='blue',legend_label='$B_{1,3}$')
fig_b23 = plot(bernstein(2,3,t),
(t,0,1),color='green',legend_label='$B_{2,3}$')
fig_b33 = plot(bernstein(3,3,t),
(t,0,1),color='brown',legend_label='$B_{3,3}$')
fig_b03 + fig_b13 + fig_b23 + fig_b33
```




```
# Bezierova křivka
def bezier(points):
    n = len(points) - 1
    return lambda t: sum([ bernstein(i,n,t)*points[i] for i
in range(n+1) ])
```

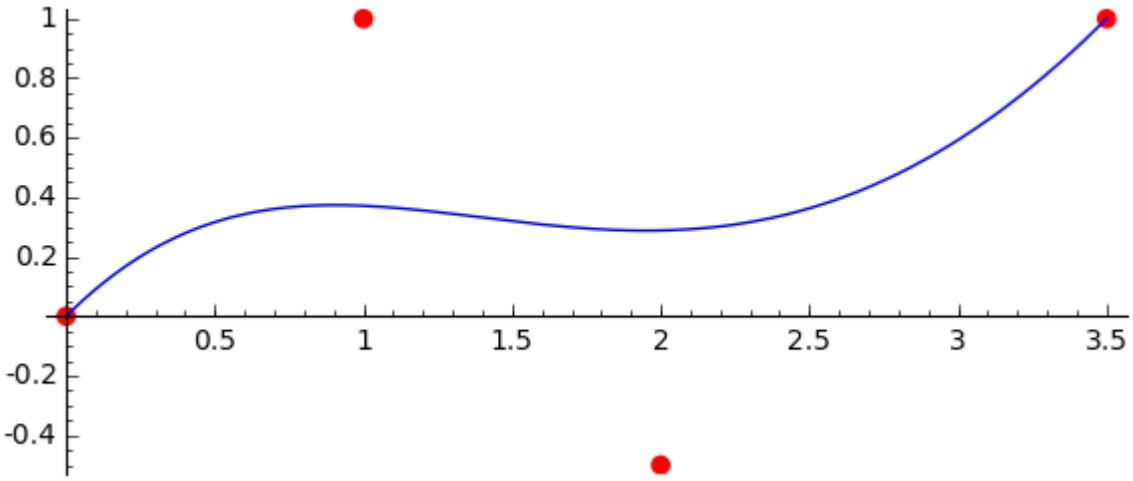
```
# Zvolme několik bodů v rovině...
R2 = VectorSpace(RR,2)
pts = [ R2([0,0]), R2([1,1]), R2([2,-0.5]), R2([3.5,1]) ]
```

```
#...a sestrojme parametrizaci
var('t')
c = bezier(pts)
```

Graf křivky.

```
points(pts,color='red',pointsize=50) + parametric_plot(c(t),
```

```
(t,0,1),figsize=6)
```



Písmeno a

Fonty ve formátu TTF jsou vektorové. Křivky formující písmena jsou právě kvadratické Bézierovy křivky.

```
<TTGlyph name="a" xMin="123" yMin="-29" xMax="1069"
yMax="1147">
      <contour>
        <pt x="702" y="563" on="1"/>
        <pt x="479" y="563" on="0"/>
        <pt x="307" y="461" on="0"/>
        <pt x="307" y="338" on="1"/>
        <pt x="307" y="240" on="0"/>
        <pt x="436" y="125" on="0"/>
        <pt x="547" y="125" on="1"/>
        <pt x="700" y="125" on="0"/>
        <pt x="885" y="342" on="0"/>
        <pt x="885" y="522" on="1"/>
        <pt x="885" y="563" on="1"/>
      </contour>
      <contour>
        <pt x="1069" y="639" on="1"/>
        <pt x="1069" y="0" on="1"/>
        <pt x="885" y="0" on="1"/>
        <pt x="885" y="170" on="1"/>
        <pt x="822" y="68" on="0"/>
        <pt x="634" y="-29" on="0"/>
        <pt x="498" y="-29" on="1"/>
        <pt x="326" y="-29" on="0"/>
        <pt x="123" y="164" on="0"/>
        <pt x="123" y="326" on="1"/>
        <pt x="123" y="515" on="0"/>
```

```
        <pt x="376" y="707" on="0"/>
        <pt x="627" y="707" on="1"/>
        <pt x="885" y="707" on="1"/>
        <pt x="885" y="725" on="1"/>
        <pt x="885" y="852" on="0"/>
        <pt x="718" y="991" on="0"/>
        <pt x="567" y="991" on="1"/>
        <pt x="471" y="991" on="0"/>
        <pt x="289" y="945" on="0"/>
        <pt x="205" y="899" on="1"/>
        <pt x="205" y="1069" on="1"/>
        <pt x="306" y="1108" on="0"/>
        <pt x="496" y="1147" on="0"/>
        <pt x="586" y="1147" on="1"/>
        <pt x="829" y="1147" on="0"/>
        <pt x="1069" y="895" on="0"/>
      </contour>
```

Extrakce kontrolních bodů.

```
c1 = [
        [702,563,1],
        [479,563,0],
        [307,461,0],
        [307,338,1],
        [307,240,0],
        [436,125,0],
        [547,125,1],
        [700,125,0],
        [885,342,0],
        [885,522,1],
        [885,563,1]
]
c1.append(c1[0])
c2 = [
        [1069,639,1],
        [1069,0,1],
        [885,0,1],
        [885,170,1],
        [822,68,0],
        [634,-29,0],
        [498,-29,1],
        [326,-29,0],
        [123,164,0],
        [123,326,1],
        [123,515,0],
        [376,707,0],
```

```
        [627,707,1],
        [885,707,1],
        [885,725,1],
        [885,852,0],
        [718,991,0],
        [567,991,1],
        [471,991,0],
        [289,945,0],
        [205,899,1],
        [205,1069,1],
        [306,1108,0],
        [496,1147,0],
        [586,1147,1],
        [829,1147,0],
        [1069,895,0]
]
c2.append(c2[0])
```

Zpracování křivky.

```
def unpack(cont):
    out = []
    for p in cont:
        out.append(p)
        if len(out) > 1:
            prev = out[-2]
            if p[-1] == 0 and prev[-1] == 0:
                out.insert(-1,[(p[0]+prev[0])/2,(p[1]+prev[1])
/2,1])
    return out

def create_splines(cont):
    k = 0
    out = []
    while k+1 < len(cont):
        if cont[k+1][-1] == 1:
            pts = [ R2(cont[k][:2]), R2(cont[k+1][:2]) ]
            k += 1
        else:
            pts = [R2(cont[k][:2]),R2(cont[k+1]
[:2]),R2(cont[k+2][:2])]
            k += 2
        out.append(bezier(pts))
    return out

auxclr = 'red'
def colorswitch():
```

```
    global auxclr
    if auxclr == 'red':
        auxclr = 'blue'
        return 'blue'
    else:
        auxclr = 'red'
        return 'red'
```

Zpracování dat.

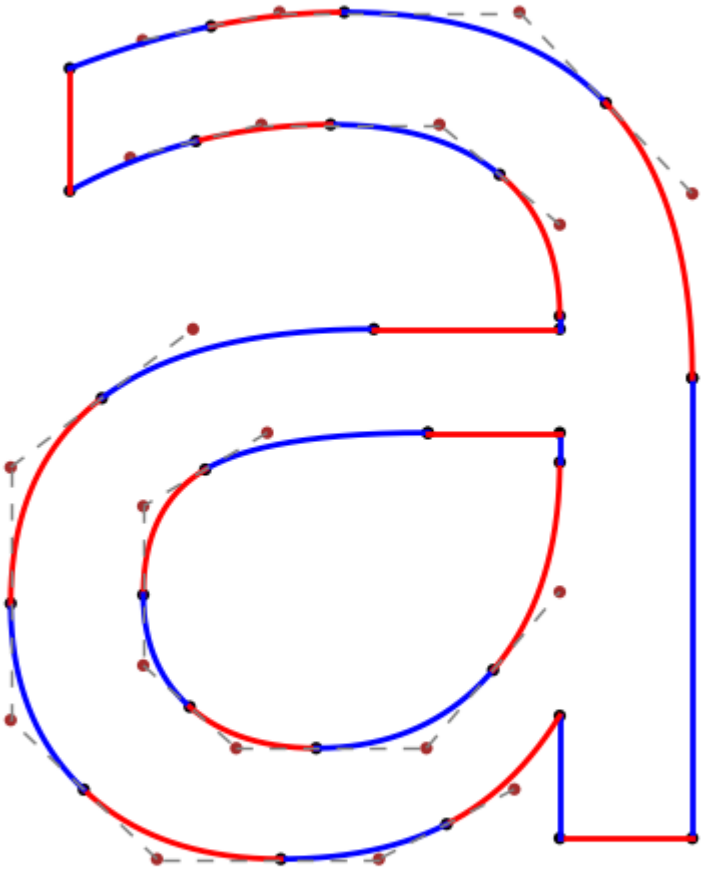
```
c1 = unpack(c1)
c2 = unpack(c2)
s1 = create_splines(c1)
s2 = create_splines(c2)
```

```
graph1 = [ parametric_plot(f(t),(t,0,1),xmin=115, ymin=-442,
xmax=965, ymax=1466, axes=False, color=colorswitch(),
thickness=2) for f in s1 ]
graph2 = [ parametric_plot(f(t),(t,0,1),xmin=115, ymin=-442,
xmax=965, ymax=1466, axes=False, color=colorswitch(),
thickness=2) for f in s2 ]
```

```
on_curve = [ p[:2] for p in c1 + c2 if p[-1] == 1 ]
off_curve = [ p[:2] for p in c1 + c2 if p[-1] == 0 ]
```

```
controls = [ line([ c1[k-1][:2], c1[k][:2], c1[k+1][:2] ],
linestyle='--', color='gray') for k in [1,..,len(c1)-2] if
(c1[k-1][-1] == 0 and c1[k][-1] == 1 and c1[k+1][-1] == 0) ]
+ [ line([ c2[k-1][:2], c2[k][:2], c2[k+1][:2] ],
linestyle='--', color='gray') for k in [1,..,len(c2)-2] if
(c2[k-1][-1] == 0 and c2[k][-1] == 1 and c2[k+1][-1] == 0) ]
```

```
show(sum(graph1) + sum(graph2) + points(on_curve,
color='black', size=20) + points(off_curve,
color='brown',size=20) + sum(controls), figsize=10)
```



Symbol integrálu

Příklad s matematickým symbolem.

```
<TTGlyph name="integral" xMin="115" yMin="-442" xMax="965"
yMax="1466">
      <contour>
        <pt x="203" y="-350" on="1"/>
```

```
<pt x="230" y="-389" on="0"/>
<pt x="272" y="-389" on="1"/>
<pt x="319" y="-389" on="0"/>
<pt x="372" y="-288" on="0"/>
<pt x="390" y="-183" on="0"/>
<pt x="399" y="-104" on="1"/>
<pt x="412" y="9" on="0"/>
<pt x="433" y="232" on="0"/>
<pt x="459" y="459" on="0"/>
<pt x="467" y="514" on="1"/>
<pt x="481" y="607" on="0"/>
<pt x="503" y="750" on="0"/>
<pt x="528" y="900" on="0"/>
<pt x="555" y="1056" on="0"/>
<pt x="569" y="1133" on="1"/>
<pt x="595" y="1273" on="0"/>
<pt x="697" y="1466" on="0"/>
<pt x="805" y="1466" on="1"/>
<pt x="875" y="1466" on="0"/>
<pt x="965" y="1358" on="0"/>
<pt x="965" y="1288" on="1"/>
<pt x="965" y="1251" on="0"/>
<pt x="916" y="1200" on="0"/>
<pt x="879" y="1200" on="1"/>
<pt x="840" y="1200" on="0"/>
<pt x="791" y="1251" on="0"/>
<pt x="791" y="1288" on="1"/>
<pt x="791" y="1323" on="0"/>
<pt x="842" y="1374" on="0"/>
<pt x="877" y="1374" on="1"/>
<pt x="848" y="1413" on="0"/>
<pt x="807" y="1413" on="1"/>
<pt x="760" y="1413" on="0"/>
<pt x="707" y="1312" on="0"/>
<pt x="689" y="1207" on="0"/>
<pt x="680" y="1128" on="1"/>
<pt x="667" y="1015" on="0"/>
<pt x="646" y="792" on="0"/>
<pt x="620" y="565" on="0"/>
<pt x="612" y="510" on="1"/>
<pt x="580" y="278" on="0"/>
<pt x="510" y="-109" on="1"/>
<pt x="486" y="-247" on="0"/>
<pt x="381" y="-442" on="0"/>
<pt x="274" y="-442" on="1"/>
<pt x="206" y="-442" on="0"/>
<pt x="115" y="-332" on="0"/>
<pt x="115" y="-264" on="1"/>
<pt x="115" y="-240" on="0"/>
```

```
        <pt x="137" y="-200" on="0"/>
        <pt x="177" y="-176" on="0"/>
        <pt x="203" y="-176" on="1"/>
        <pt x="240" y="-176" on="0"/>
        <pt x="289" y="-225" on="0"/>
        <pt x="289" y="-264" on="1"/>
        <pt x="289" y="-300" on="0"/>
        <pt x="239" y="-350" on="0"/>
      </contour>
      <instructions><assembly>
          NPUSHB[ ]  /* 38 values pushed */
          30 0 2 32 2 9 34 18 32 0 6 34 2 45 0 6 2 4 52 24 30
1 66 21 27
          1 6 0 1 66 55 48 1 6 2 59 16 3
          CALL[ ]
          IUP[1]
          SVTCA[0]
          MDAP[0]
          MDAP[0]
          CALL[ ]
          CALL[ ]
          IUP[0]
        </assembly></instructions>
    </TTGlyph>
```

```
c = [
        [203,-350,1],
        [230,-389,0],
        [272,-389,1],
        [319,-389,0],
        [372,-288,0],
        [390,-183,0],
        [399,-104,1],
        [412,9,0],
        [433,232,0],
        [459,459,0],
        [467,514,1],
        [481,607,0],
        [503,750,0],
        [528,900,0],
        [555,1056,0],
        [569,1133,1],
        [595,1273,0],
        [697,1466,0],
        [805,1466,1],
        [875,1466,0],
        [965,1358,0],
        [965,1288,1],
```

```
        [965,1251,0],
        [916,1200,0],
        [879,1200,1],
        [840,1200,0],
        [791,1251,0],
        [791,1288,1],
        [791,1323,0],
        [842,1374,0],
        [877,1374,1],
        [848,1413,0],
        [807,1413,1],
        [760,1413,0],
        [707,1312,0],
        [689,1207,0],
        [680,1128,1],
        [667,1015,0],
        [646,792,0],
        [620,565,0],
        [612,510,1],
        [580,278,0],
        [510,-109,1],
        [486,-247,0],
        [381,-442,0],
        [274,-442,1],
        [206,-442,0],
        [115,-332,0],
        [115,-264,1],
        [115,-240,0],
        [137,-200,0],
        [177,-176,0],
        [203,-176,1],
        [240,-176,0],
        [289,-225,0],
        [289,-264,1],
        [289,-300,0],
        [239,-350,0]
]
c.append(c[0])
```

```
c = unpack(c)
```

```
splines = create_splines(c)
```

```
on_curve = [ p[:2] for p in c if p[-1] == 1 ]
off_curve = [ p[:2] for p in c if p[-1] == 0 ]
```

```
graph = [ parametric_plot(f(t),(t,0,1),xmin=115, ymin=-442,
xmax=965, ymax=1466, axes=False, color=colorswitch(),
thickness=2) for f in splines ]
```

```
show(sum(graph) + points(on_curve, color='black', size=20),
figsize=8)
```