

UNIVERZA V LJUBLJANI  
FAKULTETA ZA MATEMATIKO IN FIZIKO  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Računalništvo in matematika – 2. stopnja

Kevin Štampar

**ORODJE ZA GRAFIČNI PRIKAZ  
KONCEPTOV BEZIERJEVIH IN PH KRIVULJ**

Magistrsko delo

Mentor: prof. dr. Emil Žagar

Ljubljana, 2024



# **Zahvala**

Zahvaljujem se mentorju za zelo sproščen odnos!



# Kazalo

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Bezierjeve krivulje</b>	<b>2</b>
2.1	Bernsteinovi bazni polinomi . . . . .	2
2.2	Večdimenzionalne oznake . . . . .	5
2.3	Bezierjeve krivulje . . . . .	5
2.4	Decasteljaujev algoritem . . . . .	6
2.5	Subdivizija . . . . .	10
2.6	Ekstrapolacija . . . . .	11
2.7	Višanje stopnje . . . . .	11
2.8	Odvod Bezierjeve krivulje . . . . .	13
2.9	Racionalne Bezierjeve krivulje . . . . .	14
2.9.1	Metode racionalnih Bezierjevih krivulj . . . . .	16
2.9.2	Decasteljaujev algoritem za racionalne Bezierjeve krivulje . . .	17
2.9.3	Farinove točke . . . . .	17
2.9.4	Izsek krožnice . . . . .	18
<b>3</b>	<b>Zlepki Bezierjevih krivulj</b>	<b>20</b>
3.1	Alfa parametrizacije . . . . .	22
<b>4</b>	<b>PH Krivulje</b>	<b>23</b>
4.1	Dolžina krivulje kot racionalna funkcija . . . . .	23
4.2	Polinomska parametrična hitrost . . . . .	23
4.3	Kontrolne točke Bezierjevih PH krivulj . . . . .	25
4.4	Parametrična hitrost in dolžina krivulje . . . . .	27
4.5	Enakomerna parametrizacija . . . . .	28
4.6	Tangenta, normala in ukrivljenost . . . . .	29
4.7	Racionalne odmične krivulje . . . . .	30
<b>5</b>	<b>Orodje za grafični prikaz konceptov</b>	<b>32</b>
5.1	Implementacija konceptov magistrskega dela . . . . .	32
5.2	Orodja . . . . .	32
5.2.1	Izvoz slik . . . . .	32
5.2.2	Ponastavitev grafa . . . . .	32
5.2.3	Prikaz mreže . . . . .	32
5.2.4	Povečava . . . . .	32
5.2.5	Prednastavitve . . . . .	32
5.2.6	Preklopnik za oznake točk . . . . .	32
5.3	Ukazi izbrane krivulje . . . . .	33
5.3.1	Vse krivulje . . . . .	33
5.3.2	Bezierjeva krivulja . . . . .	33
5.3.3	Racionalna Bezierjeva Krivulja . . . . .	33
5.3.4	PH Bezierjeva krivulja . . . . .	34
5.3.5	Zlepki . . . . .	34

<b>6</b>	<b>Integrali po <math>\omega</math>-kompleksih</b>	<b>35</b>
6.1	Definicija . . . . .	35

## **Program dela**

Mentor naj napiše program dela skupaj z osnovno literaturo.

Podpis mentorja:





## **Orodje za grafični prikaz konceptov Bezierjevih in PH krivulj**

### **POVZETEK**

Tukaj napišemo povzetek vsebine. Sem sodi razlaga vsebine in ne opis tega, kako je delo organizirano.

## **Tool for graphically displaying concepts of Bezier and PH curves**

### **ABSTRACT**

An abstract of the work is written here. This includes a short description of the content and not the structure of your work.

**Math. Subj. Class. (2020):** 74B05, 65N99

**Ključne besede:** integracija, kompleks

**Keywords:** integration, complex



# 1 Uvod

Napišite kratek zgodovinski in matematični uvod. Pojasnite motivacijo za problem, kje nastopa, kje vse je bil obravnavan. Na koncu opišite tudi organizacijo dela – kaj je v katerem razdelku.

**Ker je v okviru magistrskega dela nastalo orodje za grafični prikaz konceptov, bo delo polno s slikovnimi primeri.**

## 2 Bezierjeve krivulje

### 2.1 Bernsteinovi bazni polinomi

V tem podrazdelku bomo predstavili Bernsteinove bazne polinome in nekaj njihovih lastnosti, ki bodo ključne pri upeljavi Bezierjevih krivulj. Začnimo z njihovo definicijo.

**Definicija 2.1.** Za nenegativna cela števila  $n$  je  $i$ -ti *Bernsteinov bazni polinom* podan s predpisom

$$B_i^n(t) := \binom{n}{i} t^i (1-t)^{n-i}.$$

Iz definicije je očitno, da so za indekse  $i > n$ , oziroma  $i < 0$ , Bernsteinovi bazni polinomi ničelni. Pri določeni stopnji  $n$  je neničelnih Bernsteinovih baznih polinomov torej  $n+1$ . Brez dokaza povejmo, da so linearno neodvisni in zato tvorijo bazo prostora  $\mathbb{P}_n$ . Takšni bazi pravimo *Bernsteinova baza*, polinomu izraženemu v njej pa pravimo *Bernsteinov polinom*.

**Primer 2.2.** Za primer si bomo ogledali Bernsteinove bazne polinome stopenj  $n = 0, 1, 2, 3$ . Pri stopnji 0 imamo konstantni polinom s funkcijskim predpisom  $B_0^0(t) = 1$ . Za stopnjo 1 dobimo polinoma s funkcijskimi predpisi  $B_0^1(t) = 1 - t$  in  $B_1^1(t) = t$ , ki sta znana iz linearne interpolacije dveh točk. Funkcijski predpisi kvadratnih in kubičnih Bernsteinovih baznih polinomov pa so naslednji

$$\begin{aligned} B_0^2(t) &= (1-t)^2, & B_1^2(t) &= 2t(1-t), & B_2^2(t) &= t^2, \\ B_0^3(t) &= (1-t)^3, & B_1^3(t) &= 3t(1-t)^2, & B_2^3(t) &= 3t^2(1-t), & B_3^3(t) &= t^3. \end{aligned}$$

Grafe polinomov iz primera si lahko ogledamo na sliki 1. ◇

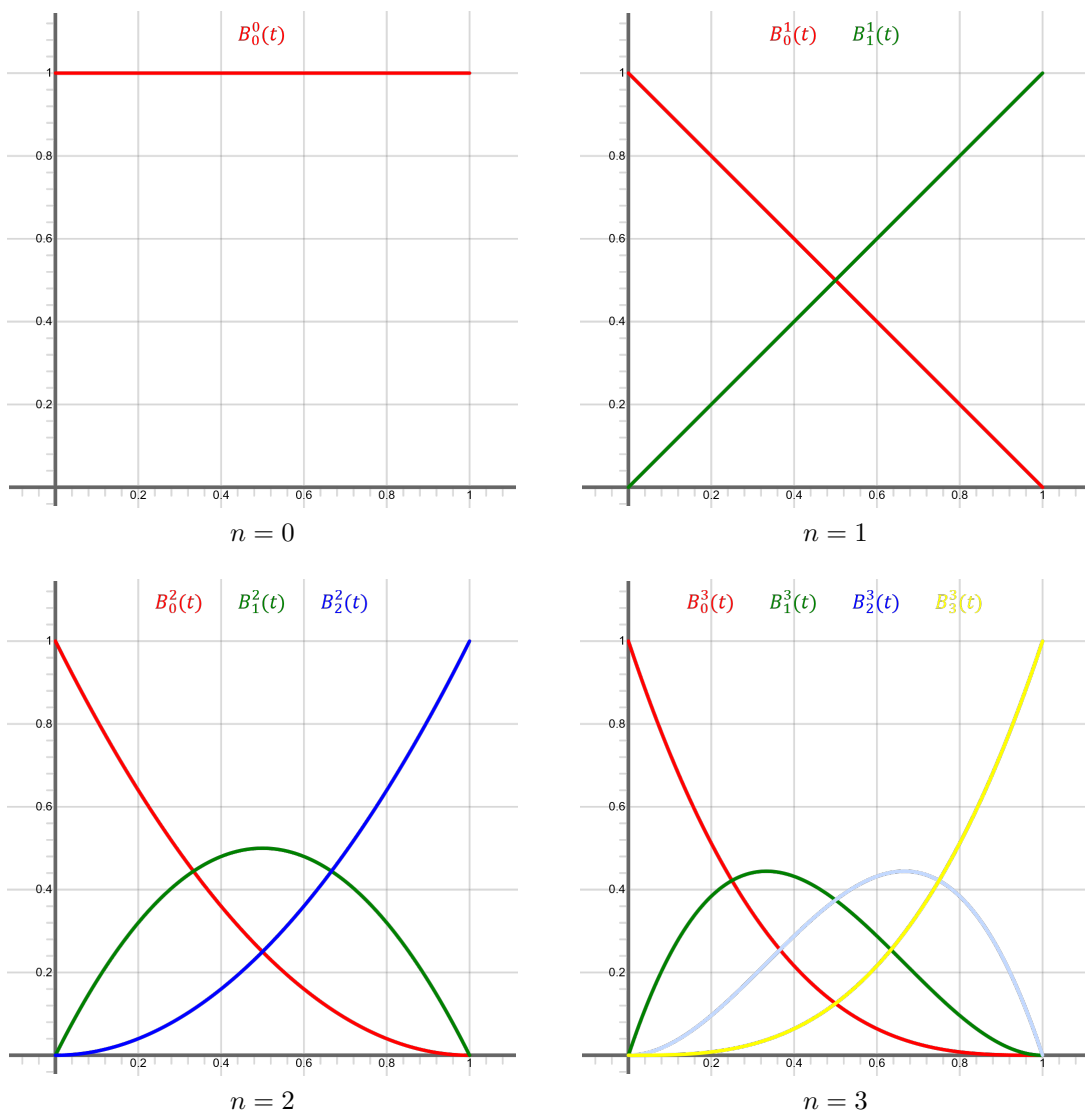
Brez dokaza v naslednjem izreku naštejmo nekaj osnovnih lastnosti Bernsteinovih baznih polinomov.

**Izrek 2.3.** Za Bernsteinove bazne polinome  $B_i^n$  velja naslednje.

1. So nenegativni na intervalu  $[0, 1]$ .
2.  $B_i^n(0) = \delta_{i,0}$  in  $B_i^n(1) = \delta_{i,n}$ , kjer je  $\delta_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$ .
3. So simetrični, tj.  $B_i^n(1-t) = B_{n-i}^n(t)$ ,  $\forall t \in \mathbb{R}$ .
4. So razčlenitev enote, tj.  $\sum_{i=0}^n B_i^n(t) = 1$ ,  $\forall t \in \mathbb{R}$ .

Prve tri lastnosti iz izreka lahko opazimo na že prej ogledani sliki 1. Četrto lastnost pa je moč opaziti na sliki 2, kjer so prikazani naloženi ploščinski grafiki Bernsteinovih baznih polinomov. Količina barve pri določenem parametru  $t \in [0, 1]$  pove, koliko pripadajoč polinom  $B_i^n$  prispeva k razčlenitvi enote.

S sledečim izrekom podamo rekurzivno zvezo za računanje vrednosti Bernsteinovih baznih polinomov.



Slika 1: Bernsteinovi bazni polinomi stopenj  $n = 0, 1, 2, 3$ .

**Izrek 2.4.** Za Bernsteinove bazne polinome stopnje  $n > 0$  velja rekurzivna zveza

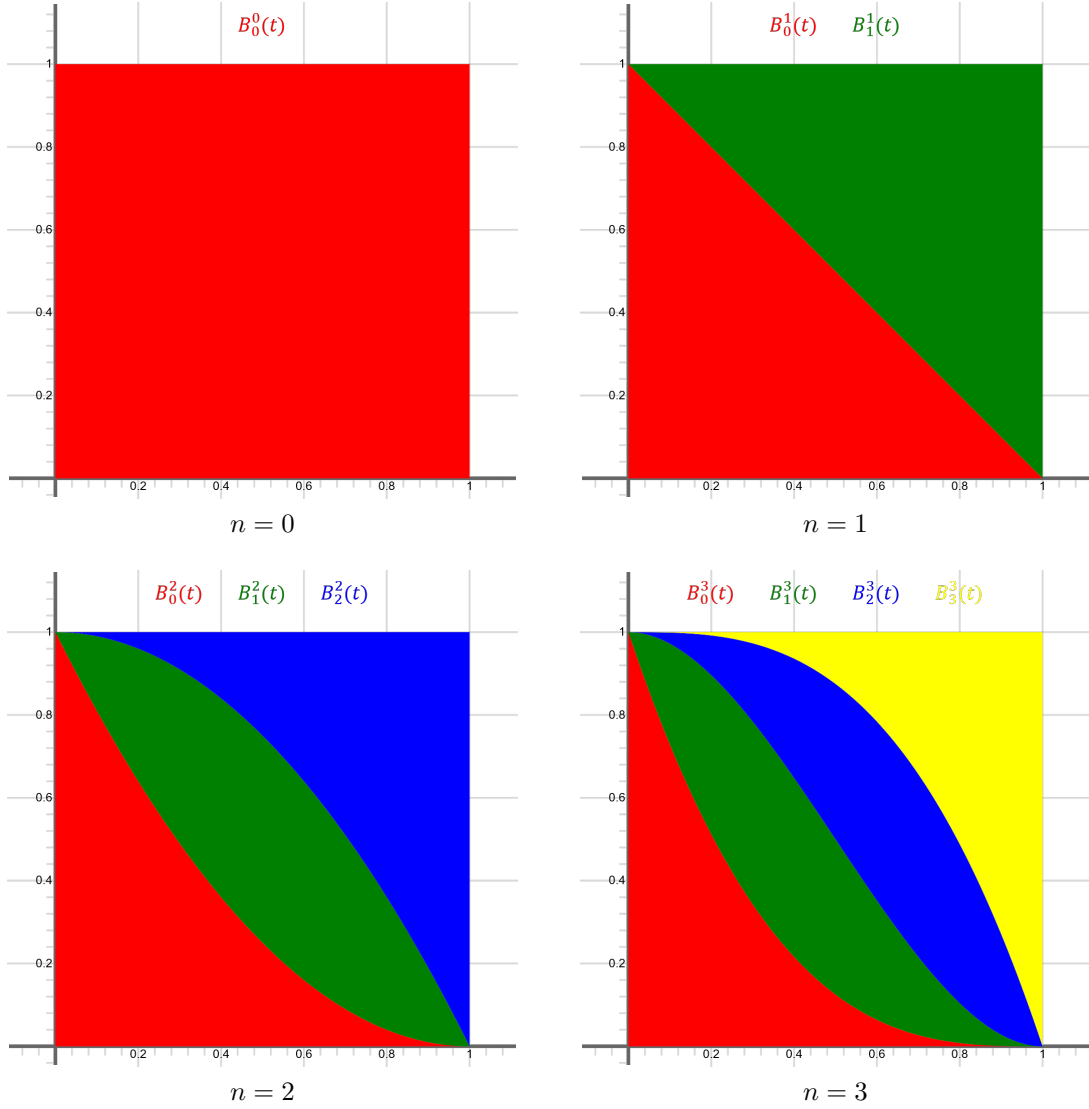
$$B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t), \quad \forall t \in \mathbb{R}.$$

Izrek je enostavno dokazati s pomočjo indukcije, zato bomo dokaz izpustili. V kasnejših razdelkih bomo potrebovali tudi odvode Bernsteinovih baznih polinomov. Ker Bernsteinovi bazni polinomi stopnje  $n$  tvorijo bazo prostora  $\mathbb{P}_n$ , lahko njihove odvode izrazimo v Bernsteinovi bazi stopnje  $n-1$ . Z nekaj računanja bi prišli do sledečega rezultata.

**Izrek 2.5.** Za odvode Bernsteinovih baznih polinomov velja zveza

$$B_i^{n'} = n(B_{i-1}^{n-1} - B_i^{n-1}).$$

Za konec podrazdelka izpeljimo še formulo za zmnožek dveh Bernsteinovih polinomov.



Slika 2: Naloženi ploščinski grafikoni Bernsteinovih baznih polinomov.

**Izrek 2.6.** Naj bosta  $f$  in  $g$  Bernsteinova polinoma s funkcijskimi predpisi  $f(t) = \sum_{i=0}^m \alpha_i B_i^m(t)$  in  $g(t) = \sum_{i=0}^n \beta_i B_i^n(t)$ . Potem za njun zmnožek velja

$$f(t)g(t) = \sum_{i=0}^{m+n} \left( \sum_{j=\max(0, i-n)}^{\min(m, i)} \frac{\binom{m}{j} \binom{n}{i-j}}{\binom{m+n}{i}} \alpha_i \beta_{i-j} \right) B_i^{m+n}(t).$$

*Dokaz.* Naj bosta  $f$  in  $g$  Bernsteinova polinoma iz izreka. Polinoma zmnožimo, da dobimo sledeče

$$f(t)g(t) = \sum_{i=0}^m \alpha_i B_i^m(t) \sum_{j=0}^n \beta_j B_j^n(t) = \sum_{i=0}^{m+n} \sum_{l=0}^i \alpha_l B_l^m(t) \beta_{i-l} B_{i-l}^n(t).$$

V izraz vstavimo predpise Bernsteinovih baznih polinomov in ga poenostavimo. Od tod sledi

$$\sum_{i=0}^{m+n} \sum_{l=0}^i \alpha_l \beta_{i-l} \binom{m}{l} \binom{n}{i-l} t^i (1-t)^{m+n-i}$$

Kar lahko z izpostavitvijo binoma  $\binom{m+n}{i}$  predstavimo v Bernsteinovi bazi kot

$$\sum_{i=0}^{m+n} \left( \sum_{l=0}^i \alpha_l \beta_{i-l} \frac{\binom{m}{l} \binom{n}{i-l}}{\binom{m+n}{i}} \right) \binom{m+n}{i} t^i (1-t)^{m+n-i}.$$

V primerih, ko velja  $l > m$  ali  $i - l > n$  imamo v števcu ulomka 0, kar privede do zapisa iz izreka.  $\square$

## 2.2 Večdimenzionalne oznake

Da bodo zapisi v sledečih razdelkih bolj pregledni, bomo uvedli večdimenzionalne oznake. Večdimenzionalnost bomo ponazarjali z odebelitvijo črke. Tako bomo večdimenzionalnost točke  $x \in \mathbb{R}^{n+1}$  označili z  $\mathbf{x} = (x_0, x_1, \dots, x_n)$ , večdimenzionalnost funkcije  $f : \mathbb{R} \rightarrow \mathbb{R}^{n+1}$  pa z  $\mathbf{f} = (f_0, f_1, \dots, f_n)$ .

## 2.3 Bezierjeve krivulje

Če v Bernsteinov polinom namesto skalarjev vstavimo točke, dobimo predpis parametrizacije Bezierjeve krivulje. Povedano formaliziramo z naslednjo definicijo.

**Definicija 2.7.** Bezierjeva krivulja  $\mathbf{B}_n : [0, 1] \rightarrow \mathbb{R}^d$  stopnje  $n \in \mathbb{N}$  in dimenzije  $d \in \mathbb{N}$ , za  $d > 1$ , je polinomska krivulja podana s točkami  $\mathbf{p}_i \in \mathbb{R}^d$ , za  $i = 0, 1, \dots, n$ , in parametrizacijo

$$\mathbf{B}_n(t) = \sum_{i=0}^n \mathbf{p}_i B_i^n(t).$$

Točkam  $\mathbf{p}_i$  pravimo *kontrolne točke*. Če zaporedne kontrolne točke povežemo, pa dobimo *kontrolni poligon*.

**Opomba 2.8.** Kjer je potrebno, lahko definicijo razširimo tudi na stopnjo  $n = 0$ . Iz zgornje parametrizacije potem sledi  $\mathbf{B}_0(t) = \mathbf{p}_0$ .

**Opomba 2.9.** Pri slikovnem gradivu iz dela se bomo omejili na prostor dimenzije  $d = 2$ , torej na Bezierjeve krivulje v ravnini.

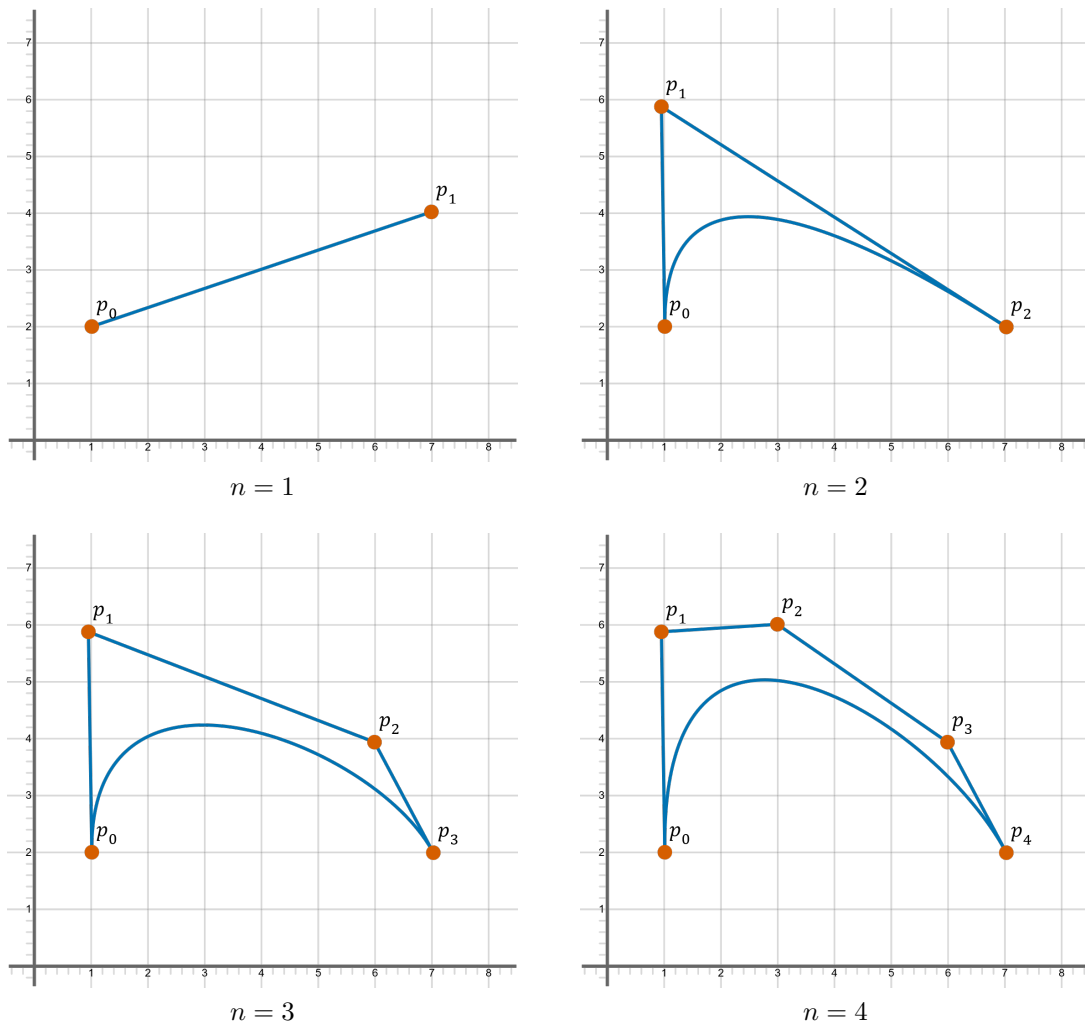
Na sliki 3 si lahko ogledamo primere Bezierjevih krivulj stopenj  $n = 1, 2, 3, 4$  s pripadajočimi kontrolnimi poligoni. Zapišimo sedaj nekaj osnovnih lastnosti Bezierjevih krivulj.

**Izrek 2.10.** Bezierjeva krivulja  $\mathbf{B}_n$  s kontrolnimi točkami  $\mathbf{p}_i$ , za  $i = 0, 1, \dots, n$ , ima sledeče lastnosti.

1. Interpolira končne točke, t.j. velja  $\mathbf{B}_n(0) = \mathbf{p}_0$  in  $\mathbf{B}_n(1) = \mathbf{p}_n$ .
2. Je afino invariantna, t.j. za poljubno afino transformacijo  $\varphi$  velja

$$\varphi \left( \sum_{i=0}^n \mathbf{p}_i B_i^n(t) \right) = \sum_{i=0}^n \varphi(\mathbf{p}_i) B_i^n(t).$$

3. Leži znotraj konveksne ovojnice svojih kontrolnih točk.



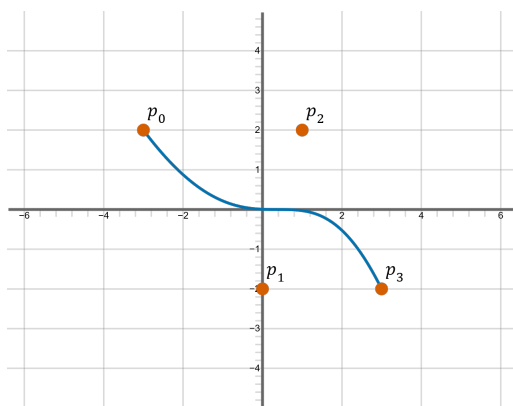
Slika 3: Bezierjeve krivulje s pripadajočimi kontrolnimi poligoni za stopnje  $n = 1, 2, 3, 4$ .

Dokazi lastnosti so enostavni, zato jih izpustimo. Preden si lastnosti ogledamo na slikah, povejmo zakaj so pomembne za CAGD sisteme. Interpolacija končnih točk uporabniku omogoča kontrolo nad tem, kje bo krivuljo začel in kje zaključil. Zaradi Afine invariance lahko uporabnikove transformacije krivulje v ozadju CAGD sistema prevedemo v transformacije kontrolnih točk. Tretja lastnost pa uporabniku s kontrolnimi točkami omogoča upravljanje krivulje, kjer je krivulja zmerom v bližini svojih kontrolnih točk. Lastnosti si sedaj oglejmo na slikah. Interpolacijo končnih točk je bilo moč videti že na sliki 3. Posledice afine invariance si lahko ogledamo na sliki 4. Na sliki 5 pa si lahko ogledamo konveksni ovojnici kontrolnih točk dveh Bezierjevih krivulj. Vidimo lahko, da krivulji ležita znotraj njih.

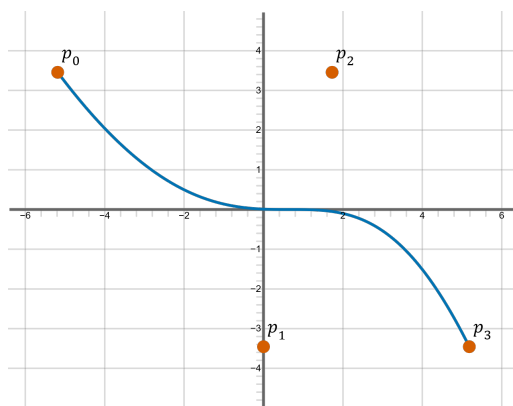
## 2.4 Decasteljauev algoritem

Stabilnost metod je v CAD in CAGD sistemih bistvene narave. Direktno računanje vrednosti Bernsteinovih polinomov preko enačb iz definicije 2.1 pa ni stabilno [placeholder]. Da lahko točke Bezierjevih krivulj računamo stabilno, potre-

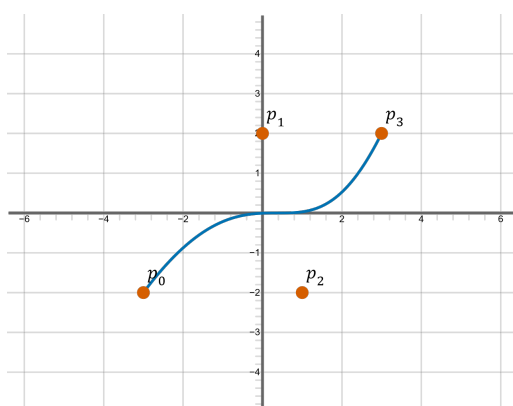




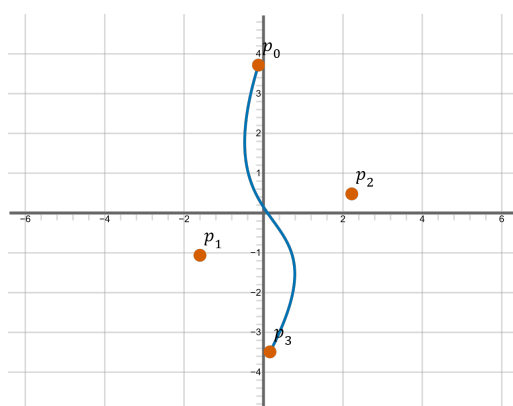
Začetna krivulja



Povečanje

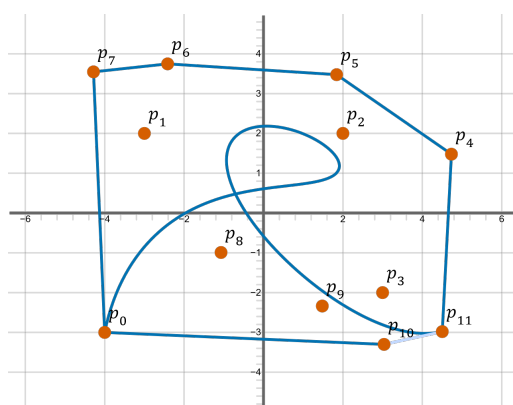
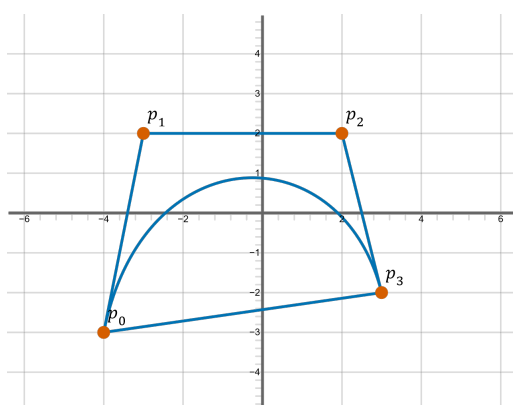


Zrcaljenje čez  $x$  os



Rotiranje za  $\frac{\pi}{2}$

Slika 4: Afine transformacije Bezierjeve krivulje.



Slika 5: Konveksni ovojnici kontrolnih točk Bezierjevih krivulj.

bujemo sledeč izrek.

**Izrek 2.11.** Označimo z  $B(t)_{[p_0, p_1, \dots, p_n]}$  parametrizacijo Bezierjeve krivulje s kontrolnimi točkami  $p_0, p_1, \dots, p_n$ . Potem za poljubno realno število  $t$  in naravno število  $n$  velja rekurzivna zveza

$$B(t)_{[p_0, p_1, \dots, p_n]} = (1 - t)B(t)_{[p_0, p_1, \dots, p_{n-1}]} + tB(t)_{[p_1, p_2, \dots, p_n]}.$$

Izrek s pomočjo indukcije tudi dokažimo.

*Dokaz.* Za  $n = 1$  zveza drži, saj iz nje dobimo

$$\mathbf{B}(t)_{[\mathbf{p}_0, \mathbf{p}_1]} = (1 - t)\mathbf{B}(t)_{[\mathbf{p}_0]} + t\mathbf{B}(t)_{[\mathbf{p}_1]} = (1 - t)\mathbf{p}_2 + t\mathbf{p}_1.$$

Indukcijski korak pa dokažemo tako, da v desni del rekurzivne zveze iz izreka vstavimo parametrizaciji Bezierjevih krivulj in dobimo

$$(1 - t)\mathbf{B}(t)_{[\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{n-1}]} + t\mathbf{B}(t)_{[\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n]} = (1 - t) \sum_{i=0}^{n-1} \mathbf{p}_i B_i^{n-1}(t) + t \sum_{i=0}^{n-1} \mathbf{p}_{i+1} B_i^{n-1}(t).$$

Nato zamaknemo indeks desne vsote in skupne točke postavimo pod eno vsoto. Od tod sledi

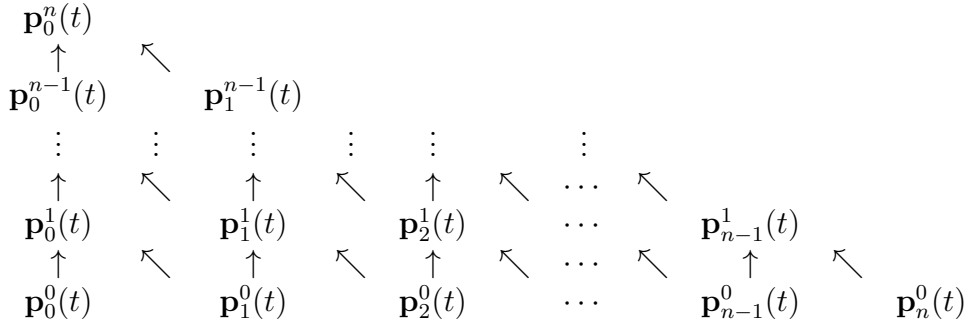
$$\begin{aligned} & (1 - t) \sum_{i=0}^{n-1} \mathbf{p}_i B_i^{n-1}(t) + t \sum_{i=1}^n \mathbf{p}_i B_{i-1}^{n-1}(t) \\ &= \mathbf{p}_0(1 - t)B_0^{n-1}(t) + \sum_{i=1}^{n-1} \left( (1 - t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t) \right) \mathbf{p}_i + \mathbf{p}_n B_{n-1}^{n-1}(t). \end{aligned}$$

Uporabimo še rekurzivno zvezo Bernsteinovih baznih polinomov iz izreka 2.4, da dobimo

$$\mathbf{p}_0 B_0^n(t) + \sum_{i=1}^{n-1} \mathbf{p}_i B_i^n(t) + \mathbf{p}_n B_n^n(t) = \sum_{i=0}^n \mathbf{p}_i B_i^n(t).$$

□

Označimo sedaj  $\mathbf{p}_i^r(t) = \mathbf{B}(t)_{[\mathbf{p}_i, \mathbf{p}_{i+1}, \dots, \mathbf{p}_{i+r}]}$ . Velja torej  $\mathbf{p}_i^0(t) = \mathbf{p}_i$  in  $\mathbf{p}_0^n(t) = \mathbf{B}(t)_{[\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n]}$ . Iz izreka 2.11 sledi, da lahko točke Bezierjeve krivulje  $\mathbf{B}(t)_{[\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n]}$  računamo s pomočjo *Decasteljaujeve sheme*, ki jo lahko vidimo na sliki 2.4. V shemi diagonalne puščice ponazarjajo množenje točke z vrednostjo  $t$ , vertikalne pa z vrednostjo  $1 - t$ . V vrhu puščic dobljene vrednosti seštejemo.



Slika 6: Decasteljaujeva shema

**Najbrž bom menjal bom sliko, ko zadevo implementiram na Bezgu**

Izračun točke Bezierjeve krivulje pri poljubnem parametru  $t$  lahko sedaj podamo v obliki *Decasteljaujevega algoritma* 1.

---

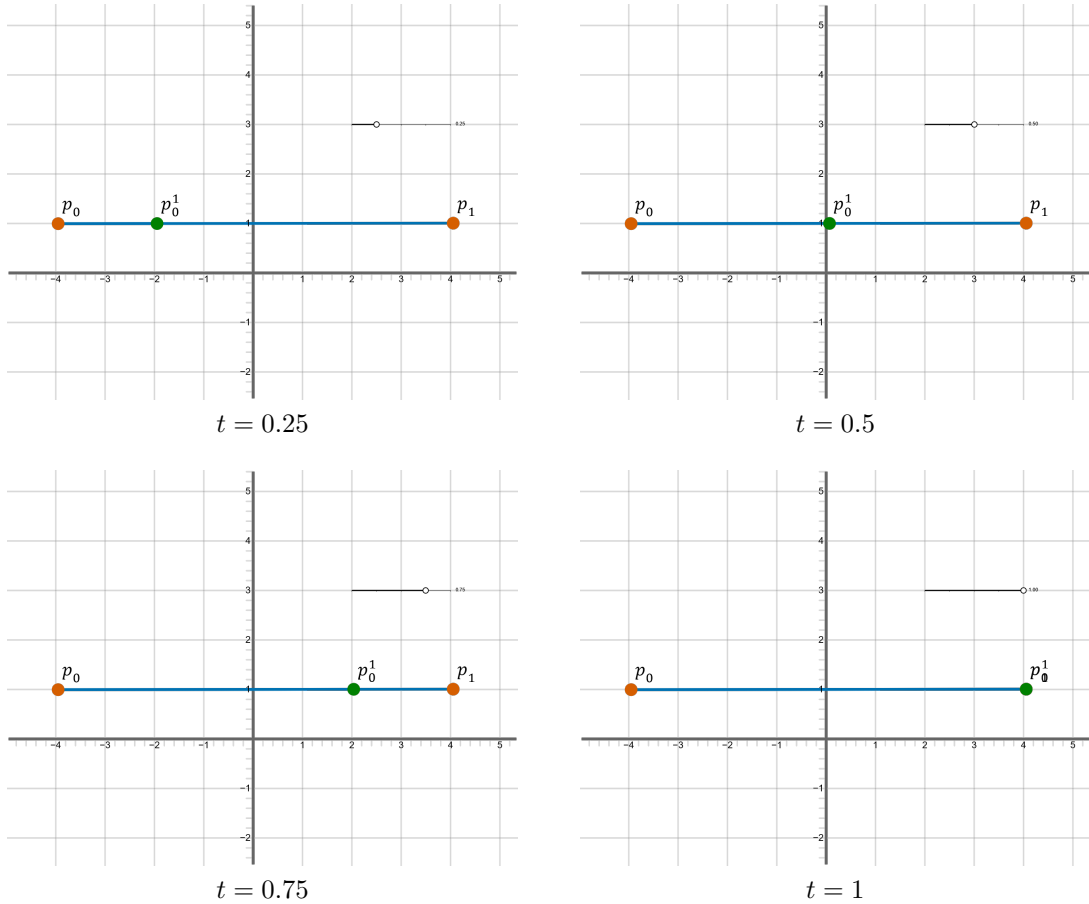
**Algoritem 1** Decasteljaujev algoritem

---

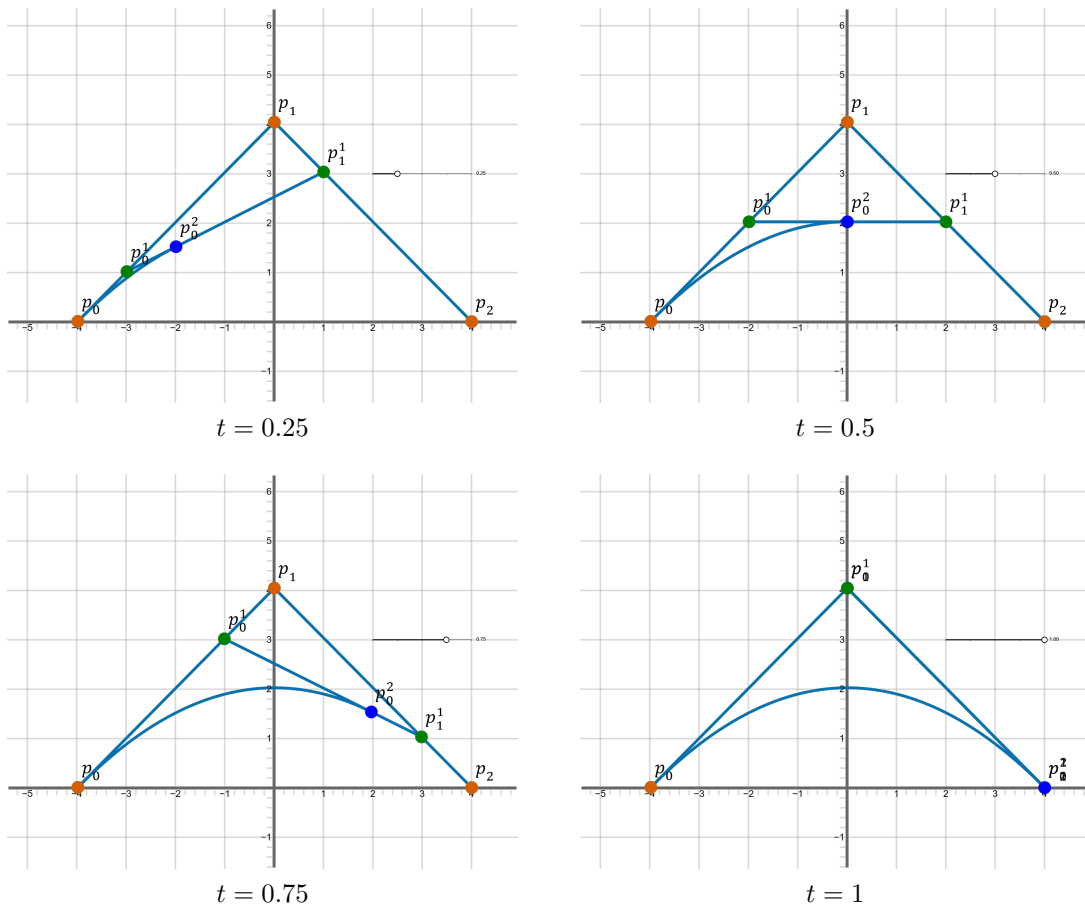
```
 $\mathbf{p} \leftarrow \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$   
 $t \leftarrow t$   
for  $i = 0, 1, \dots, n$  do  
   $\mathbf{p}_i^0(t) = \mathbf{p}_i$   
end for  
for  $r = 1, 2, \dots, n$  do  
  for  $i = 0, 1, \dots, n - r$  do  
     $\mathbf{p}_i^r(t) = (1 - t)\mathbf{p}_i^{r-1}(t) + t\mathbf{p}_{i+1}^{r-1}(t)$   
  end for  
end for  
return  $\mathbf{p}_0^n(t)$ 
```

---

Algoritem 1 ima tudi geometrijski pomen. Pri stopnji  $n = 1$  se prevede na interpolacijo dveh točk, kar lahko vidimo na sliki 7. Pri višjih stopnjah  $n$  pa algoritem predstavlja zaporedno interpolacijo točk, saj v njem za vsak nivo  $r = 1, 2, \dots, n$  interpoliramo sosednje točke prejšnjega nivoja. Slednje lahko vidimo na slikama 8 in 9.



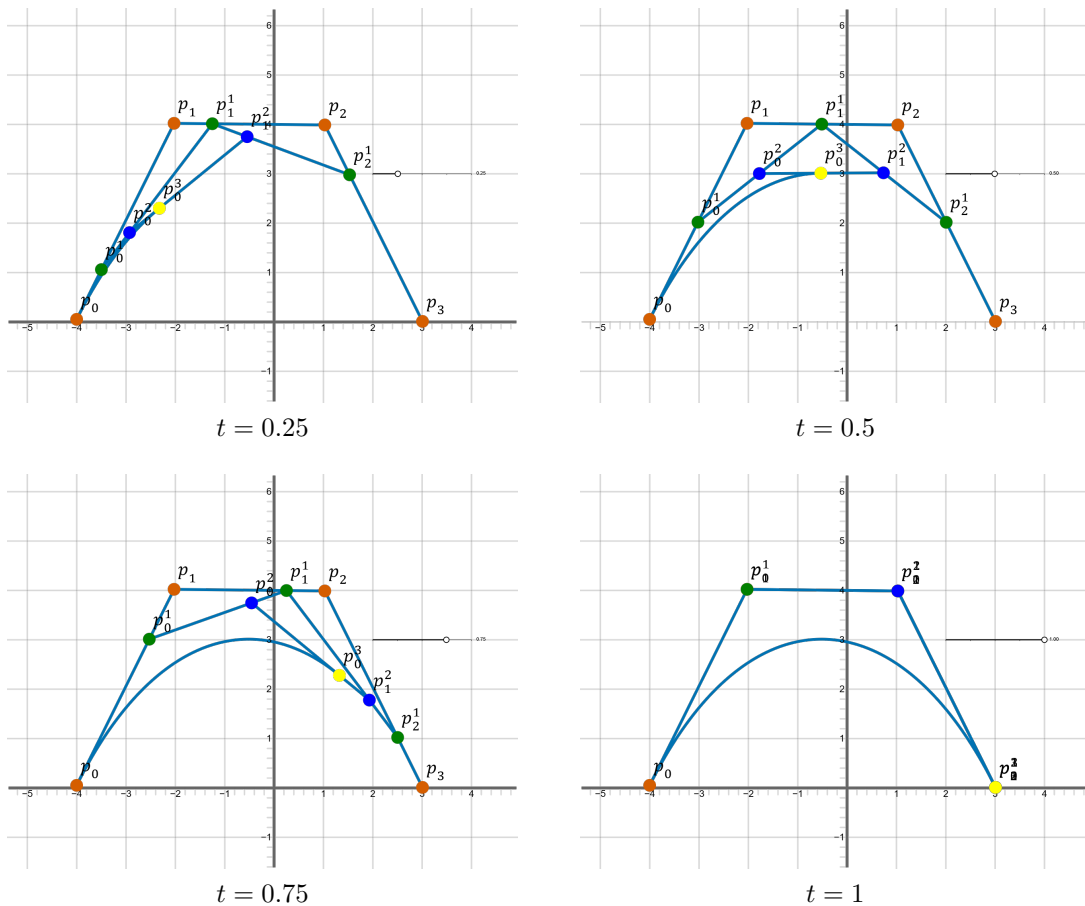
Slika 7: Decasteljaujev algoritem za  $n = 1$



Slika 8: Decasteljaujev algoritem za  $n = 2$

## 2.5 Subdivizija

V CAGD in CAD sistemih se mnogokrat zgodi, da uporabnik želeli ohraniti le en kos Bezierjeve krivulje. Naj bo to kos krivulje, ki ga dobimo tako, da za prvotno krivuljo parameter  $t$  omejimo na interval  $[0, t_0]$ , za neko pozitivno realno število  $t_0 < 1$ . Ta del krivulje označimo z  $B_{t_0-}$ . Izkaže se, da lahko krivuljo  $B_{t_0-}$  podamo kot Bezierjevo krivuljo s kontrolnimi točkami  $\mathbf{p}_0^i(t_0)$  za  $i = 0, 1, \dots, n$ , kjer točke  $\mathbf{p}_0^i(t_0)$  dobimo iz Decasteljaujeve sheme pri  $t = t_0$ . Podobno se izkaže tudi to, da lahko preostali kos krivulje,  $B_{t_0+}$ , podamo kot Bezierjevo krivuljo s kontrolnimi točkami  $\mathbf{p}_1^i(t_0)$  za  $i = 0, 1, \dots, n$ . Radovedni bralci lahko dokaz trditev najdejo v delu [placeholder]. Procesu deljenja krivulje na dva dela pravimo *subdivizija*. Na sliki sheme Decasteljaujevega algoritma 10 lahko vidimo, da pri določenem parametru  $t_0$  točke  $\mathbf{p}_0^0, \mathbf{p}_0^1, \mathbf{p}_0^2, \mathbf{p}_0^3$  tvorijo kontrolni poligon bezierjeve krivulje  $B_{t_0-}$ , točke  $\mathbf{p}_0^0, \mathbf{p}_1^1, \mathbf{p}_2^2, \mathbf{p}_3^3$  pa kontrolni poligon krivulje  $B_{t_0+}$ . Če izberimo sedaj  $t_0 = \frac{1}{2}$  in subdiviziramo krivuljo, dobimo dve krivulji. Če subdivizijo nato na dobljenih krivuljah ponavljamo, dobimo po  $k$  korakih  $2^k$  krivulj. Na sliki 11 si lahko ogledamo postopek za prve tri korake. Opazimo lahko, da so kontrolni poligoni dobljenih krivulj zmerom bližje krivulji.



Slika 9: Decasteljaujev algoritem za  $n = 3$

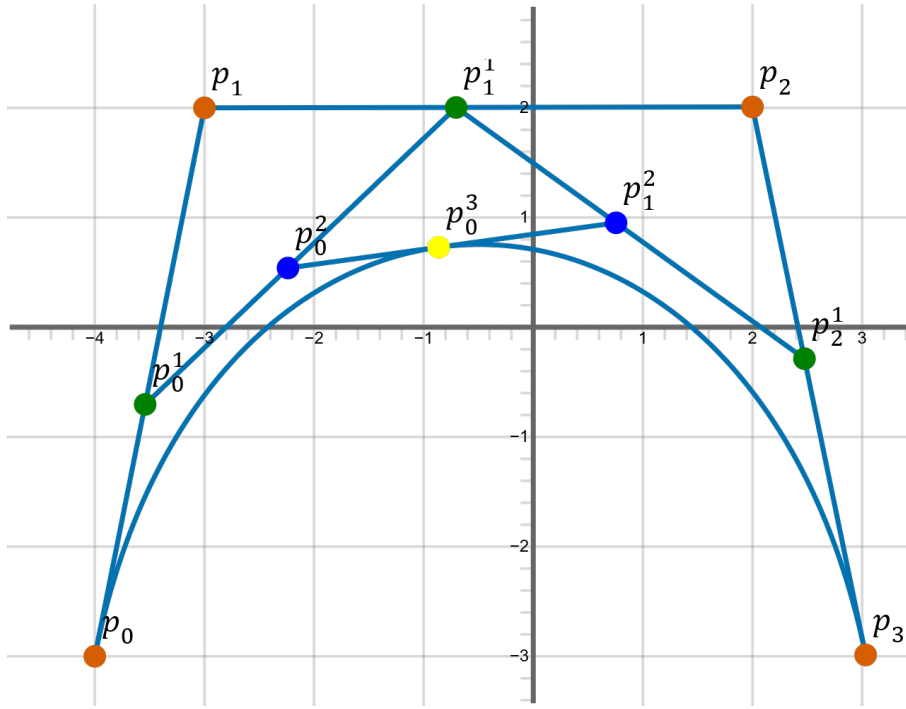
## 2.6 Ekstrapolacija

Ker so Bernsteinovi bazni polinomi definirani za vsa realna števila  $t$ , lahko krivuljo rišemo tudi izven intervala  $[0, 1]$ . Recimo, da jo želimo risati na intervalu  $[0, t_0]$ , kjer je  $t_0 > 1$ . Če želimo ekstrapolirano krivuljo želeli nato zapisati kot Bezierjevo krivuljo na intervalu  $[0, 1]$ , lahko kontrolne točke ekstrapolirane krivulje pridobimo na enak način kot smo dobili kontrolne točke krivulje  $B_{t_0-}$  pri subdiviziji.

## 2.7 Višanje stopnje

Nekateri algoritmi v CAD/CAGD sistemih za vhod potrebujejo dve Bezierjevi krivulji iste stopnje. Recimo, da imamo Bezierjevo krivuljo  $\mathbf{B}$  stopnje  $n$ , ki jo želimo predstaviti kot Bezierjevo krivuljo stopnje  $n + 1$ . Upoštevajoč  $1 - t + t = 1$ , lahko parametrizacijo krivulje  $\mathbf{B}$  zapišemo tudi kot

$$\mathbf{B}(t) = (1 - t)\mathbf{B}(t) + t\mathbf{B}(t) = \sum_{i=0}^n \mathbf{p}_i(1 - t)B_i^n(t) + \sum_{i=0}^n \mathbf{p}_i t B_i^n(t). \quad (2.1)$$



Slika 10: Decasteljaujev algoritem

Če sedaj zvezi za Bernsteinove bazne polinome

$$\begin{aligned} (1-t)B_i^n(t) &= \frac{n+1-i}{n+1} \binom{n+1}{i} t^i (1-t)^{n+1-i} = \frac{n+1-i}{n+1} B_i^{n+1}(t), \\ tB_i^n(t) &= \frac{i+1}{n+1} \frac{(n+1)!}{(n-i)!(i+1)!} t^{i+1} (1-t)^{n+1-i-1} = \frac{i+1}{n+1} B_{i+1}^{n+1}(t) \end{aligned}$$

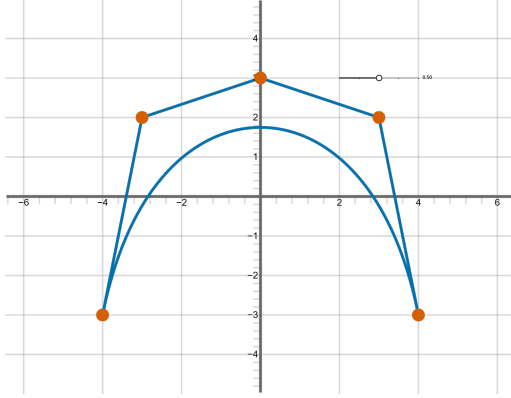
vstavimo v parametrizacijo 2.1, dobimo

$$\begin{aligned} \mathbf{B}(t) &= \sum_{i=0}^n \mathbf{p}_i \frac{n+1-i}{n+1} B_i^{n+1}(t) + \sum_{i=0}^n \mathbf{p}_i \frac{i+1}{n+1} B_{i+1}^{n+1}(t) \\ &= \sum_{i=0}^n \mathbf{p}_i \frac{n+1-i}{n+1} B_i^{n+1}(t) + \sum_{i=1}^{n+1} \mathbf{p}_{i-1} \frac{i}{n+1} B_i^{n+1}(t). \end{aligned}$$

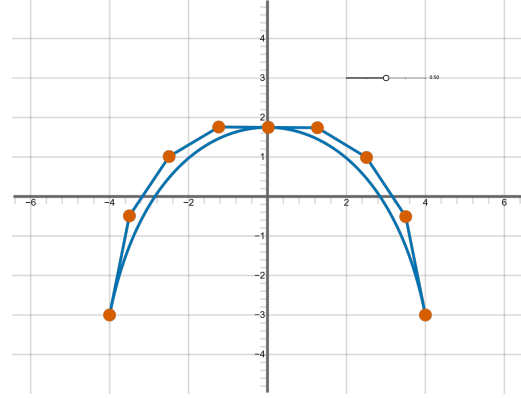
Od tod sledi, da lahko krivuljo  $\mathbf{B}$  predstavimo kot Bezierjevo krivuljo stopnje  $n+1$  s kontrolnimi točkami

$$c_0 = \mathbf{p}_0, \quad c_i = \mathbf{p}_i \frac{n+1-i}{n+1} + \mathbf{p}_{i-1} \frac{i}{n+1}, \quad c_{n+1} = \mathbf{p}_n.$$

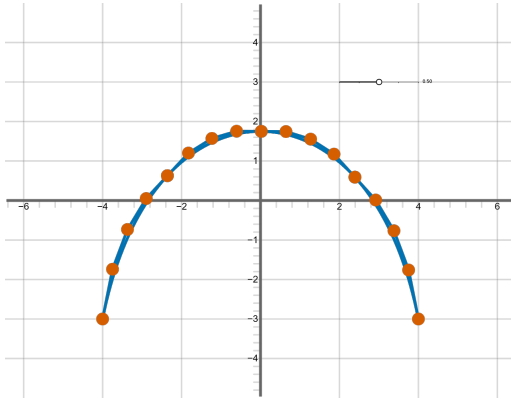
Oglejmo si kako višanje stopnje izgleda na neki krivulji. Na sliki 12 je na prvem grafu narisana Bezierjeva krivulja stopnje 3. Na drugem grafu, smo stopnjo krivulje zvišali za 1, na tretjem in četrtem grafu pa smo naredili 10 oziroma 20 višanj stopnje začetne krivulje. Krivulja je na vseh grafih enaka, imamo le več kontrolnih točk. Opaziti je tudi možno, da so kontrolne točke z vsakim višanjem stopnje bližje začetni krivulji, kontrolni poligon pa se zmerom bolj prilega začetni krivulji.



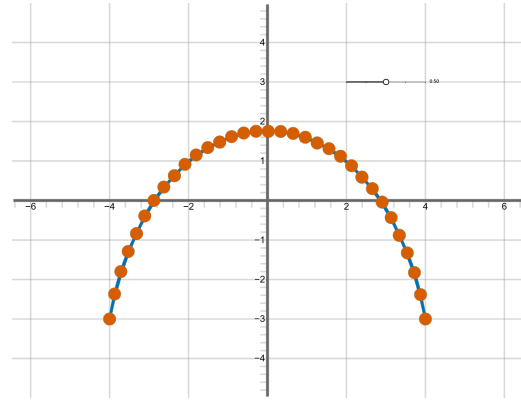
(a) Začetna krivulja s kontrolnim poligonom



(b) 2 krivulji, pridobljeni s subdivizijo



(c) 4 krivulje, pridobljene s subdivizijo



(d) 8 krivulj, pridobljenih s subdivizijo

Slika 11: Ponavljanje subdivizije na krivulji

## 2.8 Odvod Bezierjeve krivulje

V kasnejših razdelkih bomo potrebovali odvode Bezierjevih krivulj, zato jih na tem mestu izpeljemo. Z upoštevanjem izreka o odvodu Bernsteinovih baznih polinomov 2.5, dobimo naslednje

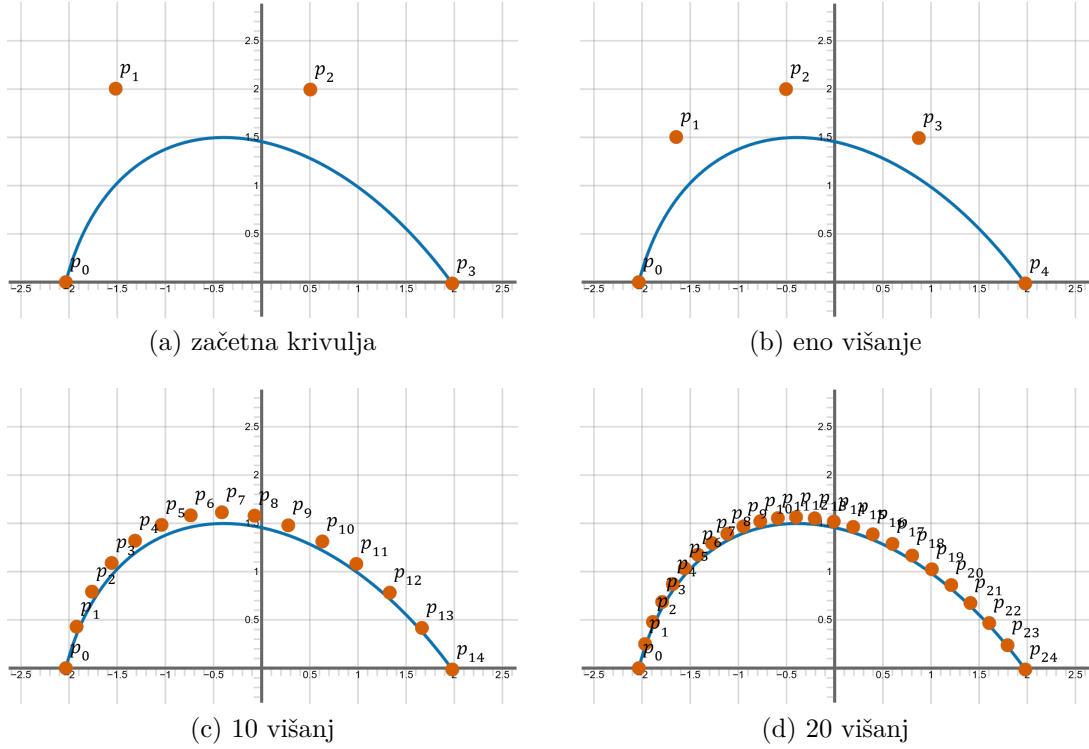
$$\begin{aligned} \mathbf{B}'(t) &= \left( \sum_{i=0}^n \mathbf{p}_i B_i^n(t) \right)' &= n \sum_{i=0}^n \mathbf{p}_i (B_{i-1}^{n-1}(t) - B_i^{n-1}(t)) \\ &= n \left( \sum_{i=1}^n \mathbf{p}_i B_{i-1}^{n-1}(t) - \sum_{i=0}^{n-1} \mathbf{p}_i B_i^{n-1}(t) \right) &= n \sum_{i=0}^{n-1} (\mathbf{p}_{i+1} - \mathbf{p}_i) B_i^{n-1}(t). \end{aligned}$$

Za krajše zapise višjih odvodov Bezierjeve krivulje, bomo uvedli operator  $\Delta$ , ki mu pravimo *prema diferenca*. Operator deluje na zaporedni točki  $\mathbf{p}_i$ , definiramo pa ga rekurzivno kot

$$\Delta^0 \mathbf{p}_i = \mathbf{p}_i, \quad \Delta \mathbf{p}_i = \mathbf{p}_{i+1} - \mathbf{p}_i, \quad \Delta^k \mathbf{p}_i = \Delta^{k-1} \mathbf{p}_{i+1} - \Delta^{k-1} \mathbf{p}_i.$$

Za pozitivno naravno število  $k$  lahko v zaključeni obliki podamo ekvivalenten zapis

$$\Delta^k \mathbf{p}_i = \sum_{j=0}^k \binom{k}{j} (-1)^{k-j} \mathbf{p}_{i+j}.$$



Slika 12: Višanje stopnje Bezierjeve krivulje

Odvod Bezierjeve krivulje lahko z uporabo preme difference predstavimo kot

$$\mathbf{B}'(t) = n \sum_{i=0}^{n-1} \Delta \mathbf{p}_i B_i^{n-1}(t).$$

Višje odvode Bezierjeve krivulje pa lahko s pomočjo indukcije predstavimo kot

$$\mathbf{B}^{(k)}(t) = \frac{n!}{n-k!} \sum_{i=0}^{n-k} \Delta^k \mathbf{p}_i B_i^{n-k}(t).$$

## 2.9 Racionalne Bezierjeve krivulje

Vseh krivulj se ne moremo opisati z navadnimi Bezierjevimi krivuljami. Med njimi so tudi takšne, ki so za CAGD sisteme zelo pomembne, na primer izsek krožnice. Za opis takšnih krivulj lahko posežemo po racionalnih Bezierjevih krivuljah. Racionalno Bezierjevo krivuljo stopnje  $n$  v  $\mathbb{R}^d$  dobimo tako, da Bezierjevo krivuljo stopnje  $n$  v  $\mathbb{R}^{d+1}$  projiciramo na hiperravnino  $w = 1$ . Točke iz  $\mathbb{R}^{d+1}$  pri tem definiramo kot  $(w, x_1, \dots, x_n)$ , projekcijo pa s predpisom  $(w, \mathbf{x}) \rightarrow (1, \frac{\mathbf{x}}{w})$ . Takšna izpeljava privede do naslednje parametrizacije.

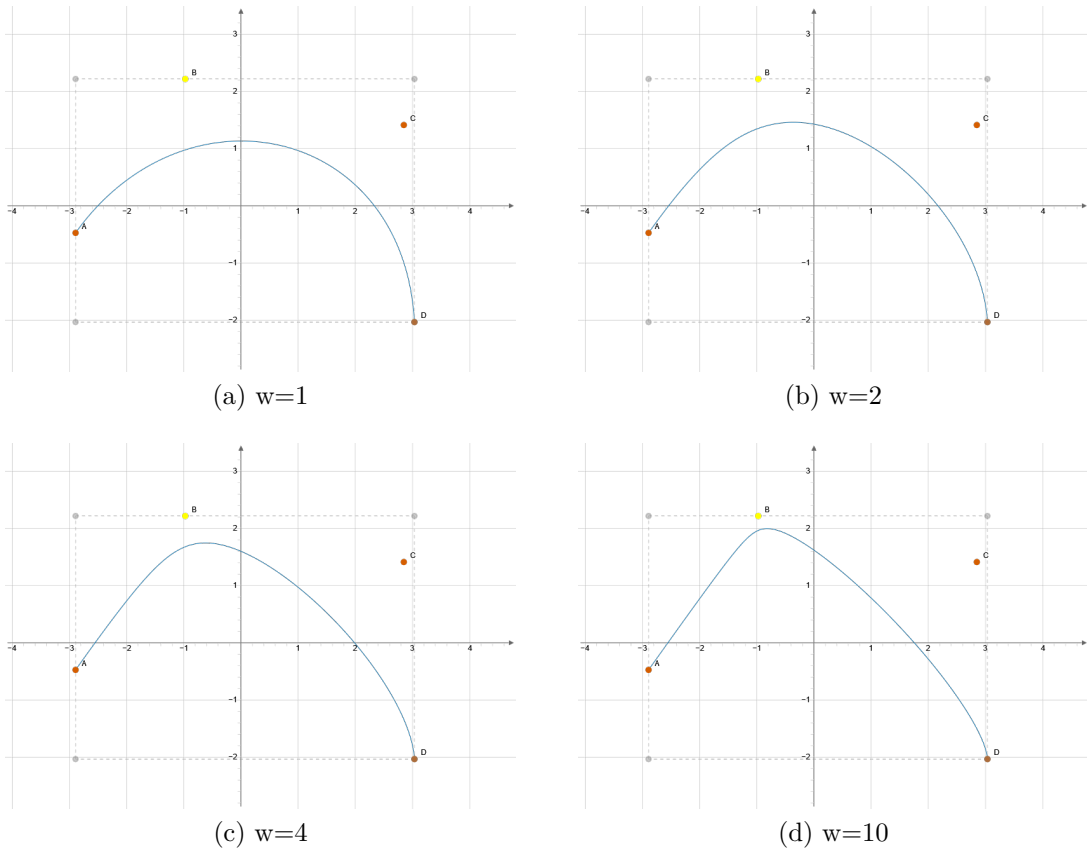
**Definicija 2.12.** Racionalna Bezierjeva krivulja stopnje  $n \in \mathbb{N}$  je podana s kontrolnimi točkami  $\mathbf{p}_i$  za  $i = 0, 1, \dots, n$  in parametrizacijo  $\mathbf{R} : [0, 1] \rightarrow \mathbb{R}^d$  določeno s predpisom

$$\mathbf{R}(t) = \frac{\sum_{i=0}^n w_i \mathbf{p}_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)},$$

kjer so kontrolne točke  $\mathbf{p}_i \in \mathbb{R}^d$  in uteži  $w_i \in \mathbb{R}$ .



Uteži so prosti parametri, ki jih lahko uporabimo pri oblikovanju. Kadar so vse uteži enake, je racionalna Bezierjeva krivulja enaka Bezierjevi krivulji z istimi kontrolnimi točkami. Da bi se izognili težavam pri deljenju z 0 ponavadi privzamemo, da so vse uteži pozitivne. Vpliv uteži si pogledjmo na sliki 13. Utež spreminjamo le pri rumeni točki, vse ostale uteži puščamo enake 1. Na grafu (a) je utež nastavljena na število 1, krivulja na sliki je zato navadna Bezierjeva krivulja. Na grafu (b) je utež nastavljena na število 2, krivulja je zato bližje rumeni točki kakor na grafu (a). Na grafih (c) in (d) pa lahko vidimo, da se z višanjem uteži, krivulja še bolj bliža rumeni točki. Iz zapisa parametrizacije v definiciji 2.12 lahko hitro vidimo,



Slika 13: Vpliv uteži

da množenje vseh uteži s poljubnim neničelnim številom parametrizacije krivulje ne spremeni. Tako lahko brez izgube splošnosti poljubno utež fiskiramo na 1. Ni pa očitno to, da lahko to storimo za dve uteži. Pogledjmo si naslednji izrek.

**Izrek 2.13.** *Racionalno Bezierjevo krivuljo s pozitivnimi utežmi  $w_i$  in parametrizacijo  $\mathbf{R}$ , lahko reparametriziramo v parametrizacijo  $\tilde{\mathbf{R}}$  s pozitivnimi utežmi  $\tilde{w}_i$  tako, da velja  $\tilde{w}_0 = \tilde{w}_n = 1$ .*

Izrek nam pove, da smo z uvedbo racionalnih Bezierjevih krivulj dobili dobili le  $n - 1$  dodatnih prostih parametrov, glede na navadne Bezierjeve krivulje. Dokaz izreka bo konstrukcijske narave. Našli bomo torej uteži  $\tilde{w}_i$ , katere lahko zamenjamo z utežmi  $w_i$ , tako da ohranimo isto krivuljo.

*Dokaz.* Uporabimo reparametrizacijsko funkcijo  $\varphi(t) : [0, 1] \rightarrow [0, 1]$  s predpisom  $\varphi(t) = \frac{t}{\rho(1-t)+t}$ , kjer je  $\rho$  pozitivno realno število. Če reparametrizacijsko funkcijo vstavimo v  $i$ -ti Bernsteinov bazni polinom dobimo naslednje

$$\begin{aligned} B_i^n(\varphi(t)) &= \binom{n}{i} \left( \frac{t}{\rho(1-t)+t} \right)^i \left( 1 - \frac{t}{\rho(1-t)+t} \right)^{n-i} \\ B_i^n(\varphi(t)) &= \binom{n}{i} \left( \frac{t}{\rho(1-t)+t} \right)^i \left( \frac{\rho(1-t)}{\rho(1-t)+t} \right)^{n-i} \\ &= \binom{n}{i} \frac{\rho^{n-1} t^i (1-t)^{n-i}}{(\rho(1-t)+t)^n} = \frac{\rho^{n-1}}{(\rho(1-t)+t)^n} B_i^n(t). \end{aligned}$$

Reparametrizirane Bernsteinove bazne polinome sedaj vstavimo v parametrizacijo racionalne Bezierjeve krivulje da dobimo

$$\mathbf{R}(\varphi(t)) = \frac{\sum_{i=0}^n \rho^{n-i} w_i \mathbf{p}_i B_i^n(t)}{\sum_{i=0}^n \rho^{n-i} w_i B_i^n(t)}.$$

S starimi utežmi izrazimo nove uteži  $\hat{w}_i = \rho^{n-i} w_i$ . Želimo, da bi veljalo  $\hat{w}_0 = \hat{w}_n$ , zato nastavimo  $\rho = \sqrt[n]{\frac{w_n}{w_0}}$ . Ker velja  $\hat{w}_n = w_n$  lahko uteži  $\hat{w}_i$  delimo z utežjo  $w_n$ , da dobimo željene uteži

$$\tilde{w}_i = \frac{1}{w_n} \hat{w}_i = \frac{w_i}{\sqrt[n]{w_n^i w_0^{n-i}}}.$$

□

**Opomba 2.14.** Če velja  $w_0 = w_n = 1$  pravimo, da je racionalna Bezierjeva krivulja predstavljena v *standardni formi*.

Lastnosti Bezierjevih krivulj, ki smo jih podali v izreku 2.10 veljajo tudi za racionalne Bezierjeve krivulje s pozitivnimi utežmi. Interpolacijo točk lahko dokažemo na podoben način, kakor smo to storili pri dokazu izreka 2.10. Da dokažemo, da je racionalna Bezierjeva krivulja afino invariantna, ter da leži znotraj konveksne ovojnice svojih kontrolnih točk, pa posežemo po naslednjem zapisu.

$$\mathbf{R}(t) = \sum_{i=0}^n \mathbf{p}_i N_i^n, \quad N_i^n(t) := \frac{w_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)}$$

Če pri dokazu izreka 2.10 namesto Bernsteinovih polinomov  $B_i^n(t)$  vstavimo funkcijo  $N_i^n$  iz zgornjega zapisa, dobimo dokaz lastnosti za racionalne Bezierjeve krivulje.

### 2.9.1 Metode racionalnih Bezierjevih krivulj

Racionalne Bezierjeve krivulje  $\in \mathbb{R}^d$  so projekcije Bezierjevih krivulj  $\in \mathbb{R}^{d+1}$ . Metode, ki smo jih predstavili na Bezierjevih krivuljah, lahko enostavno razširimo tudi na racionalne Bezierjeve krivulje. To storimo tako, da metode uporabimo na Bezierjevi krivulji  $\in \mathbb{R}^{d+1}$  ter jo nato projiciramo nazaj v ravnino  $\in \mathbb{R}^d$ . Vse metode razen ekstrapolacije ohranjajo pozitivnost uteži. **A je to res res?**

### 2.9.2 Decasteljaujev algoritem za racionalne Bezierjeve krivulje

Točke racionalnih Bezierjevih krivulj bi lahko računali tako, kakor smo pokazali za metode. **<slabtekstneberi Takšno računanje je nestabilno\*\*\*\*, zato tudi tukaj podamo stabilnejši način računanja, ki je razširitev Decasteljaujevega algoritma.** Pravilnost algoritma podamo z naslednjim izrekom.

---

#### Algoritem 2 Racionalni Decasteljaujev algoritem

---

```

 $\mathbf{p} \leftarrow \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$ 
 $w \leftarrow \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$ 
for  $i = 0, 1, \dots, n$  do
     $\mathbf{p}_i^0(t) = \mathbf{p}_i$ 
     $w_i^0(t) = w_i$ 
end for
for  $r = 1, 2, \dots, n$  do
    for  $i = 0, 1, \dots, n - r$  do
         $w_i^r(t) = (1 - t)w_i^{r-1}(t) + tw_{i+1}^{r-1}(t)$ 
         $\mathbf{p}_i^r(t) = (1 - t)\frac{w_i^{r-1}(t)}{w_i^r(t)}\mathbf{p}_i^{r-1}(t) + t\frac{w_{i+1}^{r-1}(t)}{w_i^r(t)}\mathbf{p}_{i+1}^{r-1}(t)$ 
    end for
end for
return  $\mathbf{p}_0^n(t)$ 

```

---

**Izrek 2.15.** Za vmesne točke iz algoritma  $\mathbf{p}_i^r(t)$  in poljubno realno število  $t$  velja izraz

$$\mathbf{p}_i^r(t) = \frac{\sum_{j=0}^r w_{i+j} \mathbf{p}_{i+j} B_j^r(t)}{\sum_{j=0}^r w_{i+j} B_j^r(t)}.$$

*Dokaz.*

□

### 2.9.3 Farinove točke

Ročno nastavljanje uteži z vpisovanjem, bi bilo za uporabnika CAGD sistema zamudno in nenaravno. Uporabniku želimo delo olajšati tako, da pridelamo bolj naraven način kontroliranja uteži. Racionalne Bezierjeve krivulje imajo geometrijski pomeni, ki ga lahko za ta namen uporabimo. Uteži lahko kontroliramo s t.i. *Farinovimi točkami*. Farinove točke ležijo na daljicah kontrolnega poligona, kjer  $i$ -ta Farinova točka deli  $i$ -to daljico kontrolnega poligona v razmerju  $w_{i+1} : w_i$ . Slednje označimo z  $\frac{|\mathbf{f}_i - \mathbf{p}_i|}{|\mathbf{f}_i - \mathbf{p}_{i+1}|} = \frac{w_{i+1}}{w_i}$ . S kontrolnimi točkami in utežmi lahko Farinove točke izrazimo takole

$$\mathbf{f}_i := \frac{w_i}{w_i + w_{i+1}} \mathbf{p}_i + \frac{w_{i+1}}{w_i + w_{i+1}} \mathbf{p}_{i+1}.$$

Želimo pa tudi obratno – s Farinovimi točkami želimo izraziti uteži. Brez izgube splošnosti lahko za prvo utež izberemo  $w_0 = 1$ . Ostale točke lahko nato rekurzivno izračunamo s pomočjo naslednjega izraza

$$w_{i+1} = w_i \frac{|\mathbf{f}_i - \mathbf{p}_i|}{|\mathbf{f}_i - \mathbf{p}_{i+1}|}.$$

Po želji lahko točke nato še standardiziramo s pomočjo formule,

$$\tilde{w}_i = \frac{w_i}{\sqrt[n]{w_n^i}},$$

ki sledi iz dokaza 2.9.

Na sliki 14 si lahko ogledamo delovanje Farinovih točk. Uteži na sliki niso standardizirane zato, da lahko bolj vidimo razmerja med njimi. Na grafu (a) lahko vidimo, da  $i$ -ta Farinova točka leži točno na sredini med kontrolima točkama  $\mathbf{p}_i$  in  $\mathbf{p}_{i+1}$ . Ker so vsa razmerja  $|\mathbf{p}_i - \mathbf{f}_i| : |\mathbf{p}_{i+1} - \mathbf{f}_i|$  enaka, so vse uteži enake 1. Na grafu (b) lahko vidimo, da se s premikom prve Farinove točke bližje k točki  $\mathbf{p}_0$ , k njej približa tudi krivulja. Utež  $w_1$ , ki predstavlja razmerje  $|\mathbf{p}_0 - \mathbf{f}_0| : |\mathbf{p}_1 - \mathbf{f}_0|$ , je zato manjša kot 1. Na grafu (c) lahko vidimo premik Farinove točke bližje k točki  $\mathbf{p}_1$ . Krivulja se točki približa, utež  $w_1$  pa se poveča na več kot 1. Ker na grafoma (b) in (c) druga in tretja Farinova točka ležita na sredini svojih daljic, so uteži  $w_1, w_2$  in  $w_3$  enake. Na grafoma (d) in (e) lahko vidimo, kako se krivulja in uteži obnašajo ob premiku druge Farinove točke. Na grafoma (f) ter (g) pa lahko vidimo obnašanje krivulje, ko premaknemo prvo ter tretjo Farinovo točko hkrati gor oziroma dol. Poglejmo si še, kaj se zgodi, če vse tri točke premaknemo pomaknemo tako, da so razmerja na vseh daljicah enaka. Na levem grafu slike 15 lahko vidimo, da krivulja izgleda kakor začetna. Če uteži standardiziramo, kar smo storili na desnem grafu, vidimo, da je temu res tako. Da bi preprečili takšno izbiro Farinovih točk, lahko po vsakem premiku Farinove točke uteži standardiziramo, ter ponovno izračunamo Farinove točke.

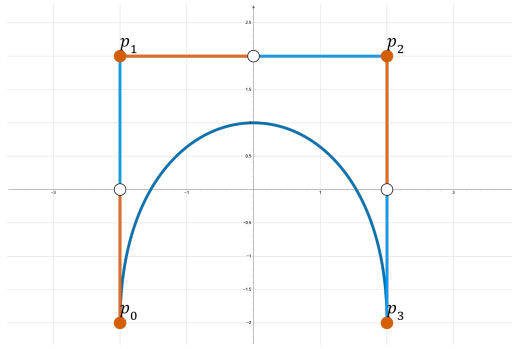
#### 2.9.4 Izsek krožnice

V uvodu podrazdelka smo povedali, da se nekaterih krivulj ne da opisati z navadnimi Bezierjevimi krivuljami. Kot enega izmed primerov smo podali izsek krožnice. V tem delu podrazdelka bomo izpeljali kontrolne točke in uteži racionalne Bezierjeve krivulje, da bo krivulja predstavljala izsek krožnice s polmerom  $R$  in kotom  $\alpha$ . Izsek krožnice bomo podali s kvadratičnimi racionalnimi Bezierjevimi krivuljami. Za konstrukcijo imamo na voljo kontrolne točke  $\mathbf{p}_0, \mathbf{p}_1$  in  $\mathbf{p}_2$ , ter uteži  $w_0, w_1$  in  $w_2$ . Iz izreka 2.13 sledi, da lahko robni uteži fiksiramo na  $w_0 = w_2 = 1$ . Točki  $\mathbf{p}_0$  in  $\mathbf{p}_2$  morata ležati na začetku ter koncu izseka krožnice, saj racionalne Bezierjeve krivulje interpolirajo končne točke. Ker so racionalne Bezierjeve krivulje afino invariantne, lahko brez izgube splošnosti postavimo točki  $\mathbf{p}_0$  in  $\mathbf{p}_2$  tako, da velja  $\mathbf{p}_0 = -\mathbf{p}_2$ . Parametrizacija željene krivulje izgleda takole

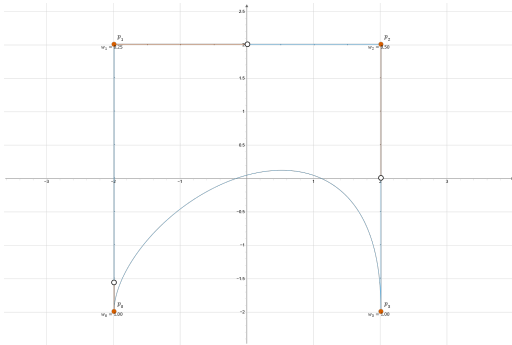
$$\mathbf{R}(t) = \frac{\sum_{i=0}^2 w_i \mathbf{p}_i B_i^2(t)}{\sum_{i=0}^2 w_i B_i^2(t)} = \frac{\mathbf{p}_0(1-t)^2 + w_1 \mathbf{p}_1 2t(1-t) + \mathbf{p}_2 t^2}{(1-t)^2 + w_1 2t(1-t) + t^2}.$$

Velja torej  $T = R(\frac{1}{2}) = \frac{w_1 \mathbf{p}_1}{w_1 + 1}$ . Poglejmo si sliko [referenca](#). Iz nje lahko razberemo zvezi

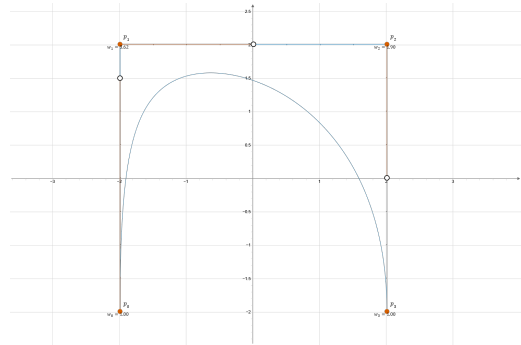
$$\begin{aligned} |OT| &= |ST| - |SO| = R - R \sin \frac{\alpha}{2} = R(1 - \sin \frac{\alpha}{2}) \\ |Op_1| &= |Sp_1| - |SO| = \frac{R}{\sin \frac{\alpha}{2}} - R \sin \frac{\alpha}{2} = R \frac{1 - \sin^2 \frac{\alpha}{2}}{\sin \frac{\alpha}{2}} \end{aligned}$$



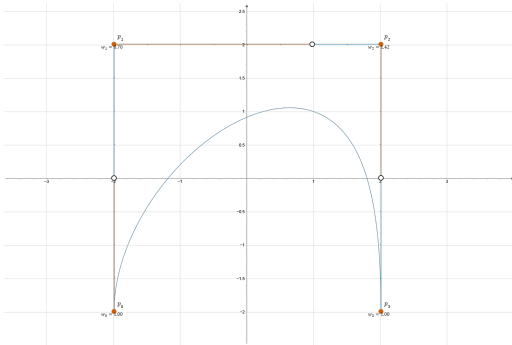
(a)



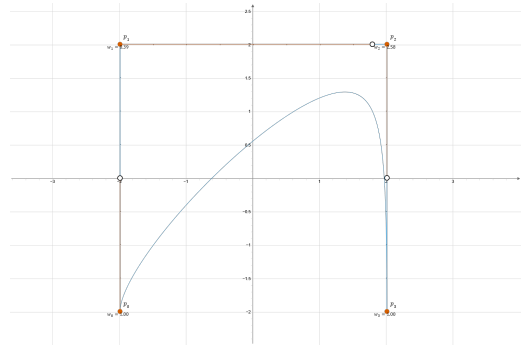
(b)



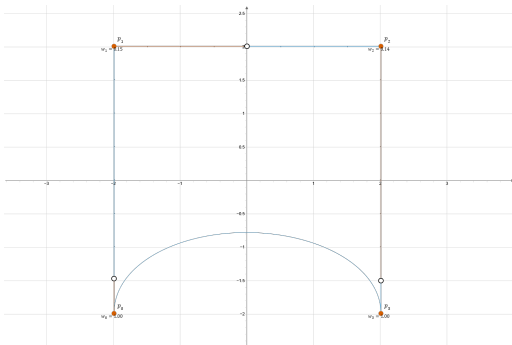
(c)



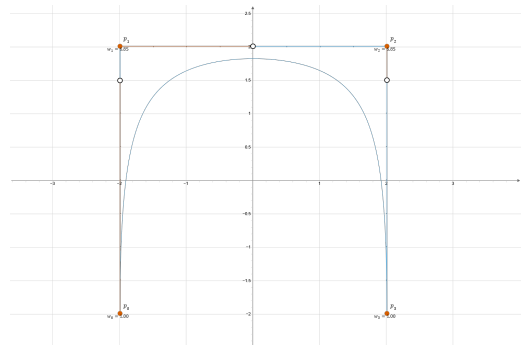
(d)



(e)



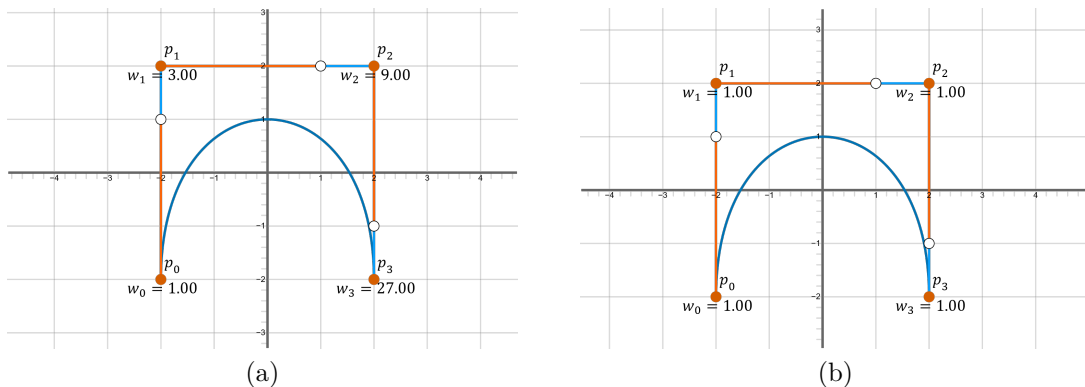
(f)



(g)

Slika 14: Kontroliranje krivulje s Farinovimi točkami

Iz zveze  $\frac{|OT|}{|Op_1|} = \frac{\sin \frac{\alpha}{2}}{\sin \frac{\alpha}{2} + 1} = \frac{w_1}{w_1 + 1}$  nato sledi  $w_1 = \sin \frac{\alpha}{2}$ . Pričakujemo, da bo točka  $p_1$  oblike  $(0, y)$ . Velja torej  $|Op_1| = y = R \frac{1 - \sin^2 \frac{\alpha}{2}}{\sin \frac{\alpha}{2}}$ .



Slika 15: Krivulja z enakimi premiki vseh Farinovih točk, pred in po standardizaciji

### 3 Zlepki Bezierjevih krivulj

Če si ponovno ogledamo Decasteljaujev algoritem 1, lahko hitro opazimo, da je časovna kompleksnost algoritma  $O(n^2)$ . Računanje točk Bezierjevih krivulj pri visokih stopnjah  $n$  je zato lahko zamudno. Takšne krivulje se zato v CAD sistemih izkažejo za neuporabne. Da bi ohranili čim večjo kontrolo krivulj s čim manj računanja, posežemo po zlepkih Bezierjevih krivulj.

**Definicija 3.1.** Zlepek krivulj  $\mathbf{S} : [a, b] \rightarrow \mathbb{R}^d$  stopnje  $n$  nad zaporedjem stičnih točk  $a = u_0 < u_1 < \dots < u_{m-1} < u_m = b$  je odsekoma polinomska krivulja, za katero velja  $\mathbf{S}|_{[u_{i-1}, u_i]} \in \mathbb{P}_n^d$ .

Želimo si tudi, da bi naši zlepki tvorili neprekinjeno krivuljo brez kolen, saj so takšne krivulje v CAGD sistemih najbolj uporabne. Krivulja je po definiciji na posameznih odsekih polinomska in zato tudi gladka, problem je le v stičnih točkah. Naj bo

$$\mathbf{s}(u) = \begin{cases} \mathbf{s}_1(u) = \sum_{i=0}^n \mathbf{p}_i^{(1)} B_i^n\left(\frac{u - u_0}{\Delta u_0}\right), & u \in [u_0, u_1), \\ \mathbf{s}_2(u) = \sum_{i=0}^n \mathbf{p}_i^{(2)} B_i^n\left(\frac{u - u_1}{\Delta u_1}\right), & u \in [u_1, u_2] \end{cases}$$

zlepek dveh Bezierjevih krivulj. Da bo zlepek zvezen, mora v stični točki  $u_1$  veljati  $\mathbf{s}_1(u_1) = \mathbf{s}_2(u_1)$  oziroma  $\mathbf{p}_n^{(1)} = \mathbf{p}_0^{(2)}$ . Takšen zlepek, si lahko ogledamo na sliki 16. Vidimo lahko, da takšen zlepek v CAGD sistemih ne bo najbolj uporaben, saj ima v stični točki koleno. Če želimo, da bo zlepek vsaj zvezno odvedljiv, mora veljati prejšnje, morata pa v stični točki sovpadati tudi odvoda  $\mathbf{s}_1'(u_1) = \mathbf{s}_2'(u_1)$ . Iz RE-FBEZIERODVOD sledi, da mora zato veljati  $\mathbf{p}_n^{(1)} - \mathbf{p}_{n-1}^{(1)} = \mathbf{p}_1^{(2)} - \mathbf{p}_0^{(2)}$ . Upoštevajoč  $\mathbf{p}_n^{(1)} = \mathbf{p}_0^{(2)}$ , lahko enačbo zapišemo tudi kot  $\mathbf{p}_n^{(1)} = \mathbf{p}_0^{(2)} = \frac{\mathbf{p}_1^{(2)} + \mathbf{p}_{n-1}^{(1)}}{2}$ . Takšen zlepek si lahko ogledamo na sliki 17. Opaziti je možno tudi geometrijski pomen pogoja, saj stična kontrolna točka leži točno na sredini med kontrolnima točkama. Kaj pa, če točka ne bi bila točno na sredini? Izkaže se, da za to, da kolena v stični točki ne bi bilo, ne potrebujemo zvezne odvedljivosti v analitičnem pomenu, dovolj je zvezna odvedljivost v geometrijskem pomenu.

### Definicija 3.2. Zlepek krivulj

$$\mathbf{s}(u) = \begin{cases} \mathbf{s}_1(u), & u \in [u_0, u_1), \\ \mathbf{s}_2(u), & u \in [u_1, u_2] \end{cases}$$

je  $n$ -krat geometrijsko zvezno odvedljiv, če lahko krivulji reparametriziramo v takšni parametrizaciji  $\tilde{\mathbf{s}}_1$  in  $\tilde{\mathbf{s}}_2$ , da je zlepek krivulj

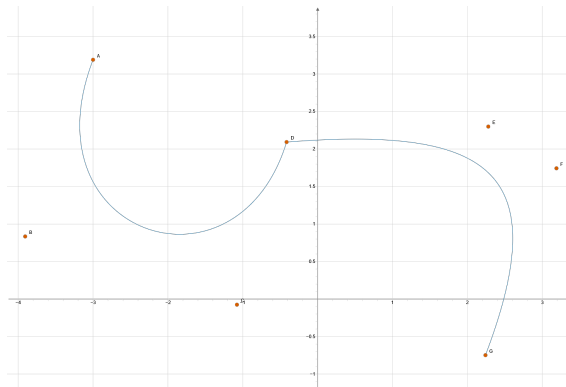
$$\tilde{\mathbf{s}}(u) = \begin{cases} \tilde{\mathbf{s}}_1(u), & u \in [u_0, u_1), \\ \tilde{\mathbf{s}}_2(u), & u \in [u_1, u_2] \end{cases}$$

$n$ -krat zvezno odvedljiv.

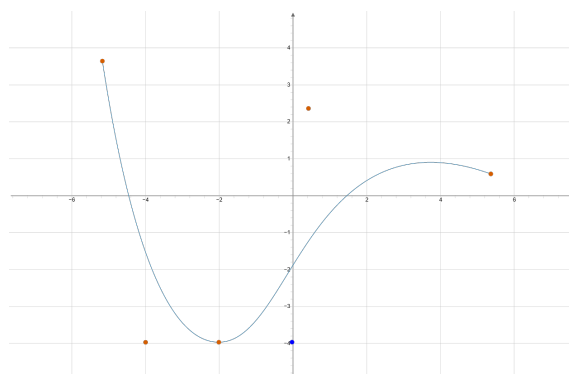
Takšna definicija, nam ne pomaga. **<- ne bomo tako pustili .. :D.** Pogoje za geometrijsko zveznost lahko prevedemo na enačbe imenovane *beta pogoji*. Zapišimo sedaj beta pogoje za geometrijsko odvedljivost stopnje 2. Zlepek iz definicije 3.2 je

1.  $G^0$ , natanko tedaj, ko je zvezen v analitičnem smislu.
2.  $G^1$ , natanko tedaj, ko velja (1) in obstaja realno število  $\beta_1 > 0$ , da velja  $\mathbf{s}'_1(u_1) = \beta_1 \mathbf{s}'_2(u_1)$ .
3.  $G^2$ , natanko tedaj, ko velja (2) in obstaja realno število  $\beta_2$ , da velja  $\mathbf{s}''_1(u_1) = \beta_2^2 \mathbf{s}''_2(u_1) + \beta_2 \mathbf{s}''_2(u_1)$ .

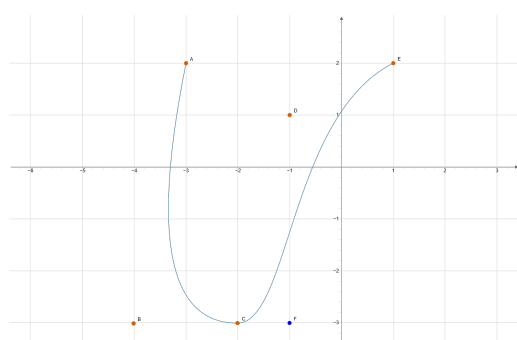
**Bom dokazal to? Nevem..... Verjetno samo ustvarim parametrizacijo na podlagi pogoja (t-> bt) vstavim not in bo?** Pogoj za  $G^1$  nam pogoj za zveznost **\*\*dodajref** spremeni v to, da mora stična točka ležati na daljici med drugima kontrolnima točkama **nekako referencirat prave**. S tem, smo pridobili prosti parameter, ki ga lahko v CAGD sistemih uporabimo za dodatno kontrolo nad krivuljo.  $G^1$  zlepek si lahko ogledamo na sliki 18.



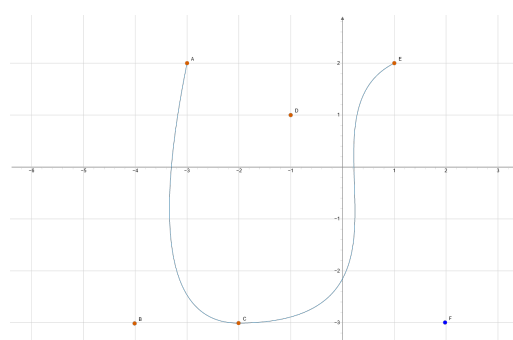
Slika 16:  $C^0$  zlepek dveh Bezierjevih krivulj



Slika 17:  $C^1$  zlepek dveh Bezierjevih krivulj



(a)  $\beta_1 = 0.5$



(b)  $\beta_1 = 2$

Slika 18: Vpliv izbora  $\beta_1$  pri  $G^1$  zlepku

### 3.1 Alfa parametrizacije

Pri zlepkih se nam lahko zgodi, da imamo nekatere odseke krajše kakor druge. Pri izbiri ekvidistantnih  $t$  dobimo potem razmike na zlepku  $s(t_i) - s(t_{i+1})$ , ki lahko zelo varirajo. Želeli bi, da so razmiki čimbolj podobni.

$$d_i = |T_i - T_{i-1}|^\alpha$$

**NE NE NE Tisto je bilo nekaj drugega. Te alfa parametrizacije so za zlepke!! Ma mentor je najbrz mislil, da bi ne o tem a? Ker to bi prislo prav, ce naredim tudi zlepke PH krivulj (kar naceleoma ne bi smelo bit tako problematicno...).**



## 4 PH Krivulje

Motivacija: fajn so nam takšne parametrizacije, da se isto hitro premikamo po njih. (ce karkoli rises recimo) V poglavju se bomo osredotočili (mogoče celo samo govorili?) o krivuljah v  $\mathbb{R}^2$ .

### 4.1 Dolžina krivulje kot racionalna funkcija

Dolžina krivulje podane s parametrizacijo  $\mathbf{r}(t) = (x(t), y(t))$  je definirana z enačbo  $\int_0^t \sqrt{x'^2(t) + y'^2(t)}$ . Integrandu v enačbi pravimo *parametrična hitrost* krivulje, označimo ga z  $\sigma(t)$ . Idealno bi bilo, da je hitrost  $\sigma(t) = 1$  za vsak  $t \in [0, t_0]$ , saj bi v tem primeru veljalo  $s(t) = t$ . Takšna parametrizacija je zlahka dosegljiva za črte, tudi za odseke krožnice (z rabo trigonometričnih funkcij), v splošnem pa ne. Dokaz da nobena krivulja ne more biti parametrizirana z racionalnimi funkcijami svoje dolžine.???

### 4.2 Polinomska parametrična hitrost

Kot smo zgoraj dokazali, ne moremo doseči racionalne parametrizacije krivulje, ki bi imela hitrost enako 1. Če krivulje nekoliko omejimo, lahko dosežemo, da bo funkcija hitrosti vsaj polinom. To dosežemo tako, da zahtevamo, da je polinom  $x'^2(t) + y'^2(t)$  popoln kvadrat. V takšnem primeru se namreč koren v integrandu izraza REFNAENACBO pokrajša, pod integralom pa ostane polinom. Krivuljam za katere to drži pravimo *krivulje s pitagorejskimi hodografi* oziroma *PH krivulje*. Integral polinoma je tudi polinom, iz česar sledi, da lahko dolžino PH krivulje izračunamo tako, da evalviramo polinom. V naslednjih razdelkih bomo videli, da nam tale lastnost omogoča več, kot samo

**Definicija 4.1.** Hodograf krivulje podane s parametrizacijo  $\mathbf{r}(t) \in \mathbb{R}^n$  je krivulja katere parametrizacija je odvod parametrizacije prvotne krivulje. **V primeru, da komponente parametrizacije hodografa krivulje tvorijo Krivulja podana s parametrizacijo  $\mathbf{r}(t) \in \mathbb{R}^n$  je krivulja s pitagorejskim hodografom, če komponente hodografa tvorijo pitagorejsko  $(n + 1)$ -terko. (???????????????? se tako rece?)**

Pri konstrukciji Bezierjevih PH krivulj bomo potrebovali polinome, ki tvorijo pitagorejske trojice. Pomagali si bomo z naslednjim izrekom.

**Izrek 4.2.** Polinomi  $a, b$  in  $c$  tvorijo pitagorejsko trojico, tj. zadoščajo enačbi

$$a^2(t) + b^2(t) = c^2(t), \quad (4.1)$$

natanko tedaj, ko obstajata tuja si polinoma  $u$  in  $v$  ter nek polinom  $w$  za katere velja

$$\begin{aligned} a(t) &= [u^2(t) - v^2(t)]w(t), \\ b(t) &= 2u(t)v(t)w(t), \\ c(t) &= [u^2(t) + v^2(t)]w(t). \end{aligned} \quad (4.2)$$

Izrek tudi dokažimo.

*Dokaz.*

( $\Leftarrow$ ) Dokaz je enostaven, saj lahko vrednosti polinomov  $a, b$  in  $c$  iz enačbe (4.2) vstavimo v enačbo (4.1) in preverimo.

( $\Rightarrow$ ) Najprej definiramo polinom  $w(t) = \gcd(a(t), b(t), c(t))$ . Pri tako definiranim polinomu  $w$ , so si polinomi  $\tilde{a}(t) = \frac{a(t)}{w(t)}$ ,  $\tilde{b}(t) = \frac{b(t)}{w(t)}$ ,  $\tilde{c}(t) = \frac{c(t)}{w(t)}$  tuji in zadoščajo enačbi  $\tilde{a}^2(t) + \tilde{b}^2(t) = \tilde{c}^2(t)$ . Enačbo zapišemo nekoliko drugače

$$\tilde{b}^2(t) = \tilde{c}^2(t) - \tilde{a}^2(t) = [\tilde{c}(t) + \tilde{a}(t)][\tilde{c}(t) - \tilde{a}(t)].$$

Polinoma  $\tilde{c}(t) + \tilde{a}(t)$  in  $\tilde{c}(t) - \tilde{a}(t)$  ne moreta imeti skupnih ničel, saj bi to impliciralo skupne ničle polinomov  $\tilde{a}, \tilde{b}, \tilde{c}$ , kar bi bilo v nasprotju z njihovo definicijo. Ker skupnih ničel nimata, mora biti vsaka ničla polinoma  $\tilde{b}$  tudi ničla sode stopnje enega izmed polinomov  $\tilde{c}(t) + \tilde{a}(t)$  ali  $\tilde{c}(t) - \tilde{a}(t)$ . Tako lahko definiramo tuja si polinoma  $u$  in  $v$ , da velja  $\tilde{c}(t) + \tilde{a}(t) = 2u^2(t)$  in  $\tilde{c}(t) - \tilde{a}(t) = 2v^2(t)$ . Iz enačb potem hitro sledi

$$\tilde{a} = u^2(t) - v^2(t), \quad \tilde{b}(t) = 2u(t)v(t), \quad \tilde{c}(t) = u^2(t) + v^2(t).$$

Če enačbe pomnožimo s polinomom  $w$  dobimo željeno obliko (4.2).  $\square$

Iz konstrukcije dokaza lahko hitro vidimo, da je za pitagorejske trojice pri katerih je največji skupni delitelj konstanta, konstanten tudi polinom  $w$  ter  $\gcd(u(t), v(t))$ . Takšnim trojicam pravimo *primitivne pitagorejske trojice*. Parametrizacijo ravninske PH krivulje  $\mathbf{r}(t) = (x(t), y(t))$  lahko dobimo tako, da vstavimo tuja si polinoma  $u, v$  in polinom  $w$  v izraza

$$x'(t) = [u^2(t) - v^2(t)]w(t), \quad y'(t) = 2u(t)v(t)w(t) \quad (4.3)$$

in integriramo. Brez izgube splošnosti lahko  $x'$  oziroma  $y'$  asociiramo z  $a$  ali  $b$  (pac kateri izraz kateremo pripopamo, ma je treba to dokazat). Nekateri izbori polinomov  $u, v, w$  porodijo izrojene(???) krivulje, takšne izbore bi radi izločili. Izbori, ki porodijo izrojene krivulje so sledeči:

1.  $w(t) = 0$  ali  $u(t) = v(t) = 0$ , hodograf je v takšnem primeru enak  $x'(t) = y'(t) = 0$ . Takšen hodograf ne definira krivulje ampak točko.
2. Če so polinomi  $w(t), u(t), v(t)$  konstantni in ne ustrezajo točki (a), potem je konstanten tudi hodograf. Takšen hodograf definira enakomerno parametrizirano premo črto.
3. Če sta polinoma  $u$  in  $v$  konstantna ter vsaj en neničelen, polinom  $w$  pa ni konstanten. Potem hodograf spet definira premo črto, a tokrat parametrizacija ni enakomerna. (hitrost ni konstanta)
4. Enako kot v točki (c) dobimo tudi v primeru, ko je polinom  $w$  neničelen in en izmed polinomov  $u$  in  $v$  je ničlen.

Neizrojene PH krivulje dobimo torej pri izborih neničelnih polinomov  $u, v$  in  $w$ , kjer vsaj en izmed polinomov  $u$  in  $v$  ni konstanten.

S številom  $\lambda$  označimo stopnjo polinoma  $w$ , s številom  $\mu$  pa  $\max(\deg(u(t)), \deg(v(t)))$ . PH krivulja, pridobljena z integracijo zgornjega hodografa, je stopnje  $n = \lambda + 2\mu + 1$ . Prostih parametrov pa je manj. Vsak izmed polinomov  $u$  in  $v$  je namreč definiran z največ  $\mu + 1$  parametri. Brez izgube splošnosti lahko vodilni koeficient polinoma  $w$

fiksiramo na 1. Polinom  $w$  je zato definiran z  $\lambda$  parametri. Integracijska konstanta pri integriranju hodografa nam poda še 2 prosta parametra. Skupno je prostih parametrov zato  $\lambda + 2(\mu + 1) + 2 = \lambda + 2\mu + 4 = n + 3$ . Kjer zadnja enakost sledi iz zgornje enačbe za stopnjo PH krivulje. Vseh parametrov ne moremo porabiti za obliko krivulje. Dva parametra nam določi izbor začetne točke, en parameter nam določi usmerjenost/rotacija/(???) krivulje, še dva nam določi izbor parametrizacije, saj substitucija  $t \rightarrow at + b$  ne spremeni oblike ali stopnje krivulje. Ostane nam  $n - 2$  prostih parametrov za obliko krivulje.

### 4.3 Kontrolne točke Bezierjevih PH krivulj

V tem podrazdelku bomo s kontrolnimi točkami Bezierjevih krivulj karakterizirali *Bezierjeve PH krivulje*, to so Bezierjeve krivulje katerih hodograf je pitagorejska trojica. Osredotočili se bomo na hodografe primitivnih pitagorejskih trojic. Takšni hodografi definirajo regularne PH krivulje, saj za njih velja  $\mathbf{r}(t) \neq 0$  pri vsakem realnem številu  $t$ . Tako dobljene PH krivulje so lihe stopnje  $n = 2\mu + 1$ .

Najbolj osnovne netrivialne PH krivulje dobimo tako, da za polinoma  $u$  in  $v$  izberemo Bernsteinova polinoma

$$u(t) = u_0 B_0^1(t) + u_1 B_1^1(t), \quad v(t) = v_0 B_0^1(t) + v_1 B_1^1(t)$$

pri katerih mora veljati  $u_0 v_1 - u_1 v_0 \neq 0$  in  $(u_1 - u_0)^2 + (v_1 - v_0)^2 \neq 0$ , da sta si polinoma  $u$  in  $v$  tuja, ter da vsaj en od njiju ni konstanten. Za polinom  $w$  izberemo konstanten polinom  $w(t) = 1$ . Ko polinome vstavimo v enačbo REFERENCANAENACBO, dobimo hodograf

$$\begin{aligned} x'(t) &= (u_0^2 - v_0^2) B_0^2(t) + (u_0 u_1 - v_0 v_1) B_1^2(t) + (u_1^2 - v_1^2) B_2^2(t) \\ y'(t) &= 2u_0 v_0 B_0^2(t) + (u_0 v_1 + u_1 v_0) B_1^2(t) + 2u_1 v_1 B_2^2(t). \end{aligned}$$

Hodograf sedaj integriramo tako, da uporabimo lastnost ?? iz izreka 2.3. Če upoštevamo še, da so Bernsteinovi polinomi razčlenitev enote dobimo

$$\begin{aligned} x(t) &= x_0 (B_0^3(t) + B_1^3(t) + B_2^3(t) + B_3^3(t)) \\ &\quad + \frac{1}{3} (u_0^2 - v_0^2) (B_1^3(t) + B_2^3(t) + B_3^3(t)) \\ &\quad + \frac{1}{3} (u_0 u_1 - v_0 v_1) (B_2^3(t) + B_3^3(t)) \\ &\quad + \frac{1}{3} (u_1^2 - v_1^2) B_3^3(t) \\ y(t) &= y_0 (B_0^3(t) + B_1^3(t) + B_2^3(t) + B_3^3(t)) \\ &\quad + \frac{1}{3} 2u_0 v_0 (B_1^3(t) + B_2^3(t) + B_3^3(t)) \\ &\quad + \frac{1}{3} (u_0 v_1 + u_1 v_0) (B_2^3(t) + B_3^3(t)) \\ &\quad + \frac{1}{3} 2u_1 v_1 B_3^3(t), \end{aligned}$$

kar ustreza Bezierjevi krivulji s kontrolnimi točkami

$$\begin{aligned}\mathbf{p}_1 &= \mathbf{p}_0 + \frac{1}{3}(u_0^2 - v_0^2, 2u_0v_0), \\ \mathbf{p}_2 &= \mathbf{p}_1 + \frac{1}{3}(u_0u_1 - v_0v_1, u_0v_1 + u_1v_0), \\ \mathbf{p}_3 &= \mathbf{p}_2 + \frac{1}{3}(u_1^2 - v_1^2, 2u_1v_1)\end{aligned}\tag{4.4}$$

kjer je točka  $\mathbf{p}_0 = (x_0, y_0)$ ,  $x_0$  in  $y_0$  pa sta integracijski konstanti.

Kontrolni poligoni, karakterizirani z enačbo (4.4) imajo tudi geometrijsko interpretacijo....

**Kam zdej zapeljat? Se bomo sli kompleksnih števil ali ne? Mater tezka odlocitev.... pogledat je treba racionalne odmike in shit, ce je slucjano treba dejansko nujno zapeljat v kompleksne vode.... sej so lepe samo nocem prevec balasta.**

Kot smo v prejšnjem razdelku povedali, ima PH krivulja  $n$ -te stopnje  $n - 2$  parametrov, ki jih definirajo obliko. Pri pravkar definiranih krivuljah to pomeni, da ima krivulja le en prosti parameter, ki jih definira obliko. Takšne krivulje so v CAD sistemih neuporabne, saj uporabniku ne nudijo dovolj kontrole. Če želimo podoben nivo kontrole, kakor pri kubičnih Bezierjevih krivuljah, moramo poseči po *kvintičnih* Bezierjevih PH krivuljah. Podobno kakor prej, za polinom  $w$  izberemo  $w(t) = 1$ , za polinoma  $u$  in  $v$  pa izberemo Bernsteinova polinoma, le da tokrat izberemo polinoma stopnje 2

$$u(t) = u_0B_0^2(t) + u_1B_1^2(t) + u_2B_2^2(t), \quad v(t) = v_0B_0^2(t) + v_1B_1^2(t) + v_2B_2^2(t).$$

Da sta si polinoma  $u$  in  $v$  tuja, mora tukaj veljati  $(u_2v_0 - u_0v_2)^2 \neq 4(u_0v_1 - u_1v_0)(u_1v_2 - u_2v_1)$ . Vstavimo ju v hodograf in integriramo, da dobimo Bezierjevo krivuljo z naslednjimi kontrolnimi točkami

$$\begin{aligned}\mathbf{p}_1 &= \mathbf{p}_0 + \frac{1}{5}(u_0^2 - v_0^2, 2u_0v_0), \\ \mathbf{p}_2 &= \mathbf{p}_1 + \frac{1}{5}(u_0u_1 - v_0v_1, u_0v_1 + u_1v_0), \\ \mathbf{p}_3 &= \mathbf{p}_2 + \frac{2}{5}(u_1^2 - v_1^2, 2u_1v_1) + \frac{1}{5}(u_0u_2 - v_0v_2, u_0v_2 + u_2v_0), \\ \mathbf{p}_4 &= \mathbf{p}_3 + \frac{1}{5}(u_1u_2 - v_1v_2, u_1v_2 + u_2v_1), \\ \mathbf{p}_5 &= \mathbf{p}_4 + \frac{1}{5}(u_2^2 - v_2^2, 2u_2v_2).\end{aligned}\tag{4.5}$$

**Lema 4.3.** *Kvintična Bezierjeva PH krivulja, definirana s kontrolnimi točkami podanimi z izrazi (4.5) ima ali dva prevoja, ali pa prevoja nima, odvisno od pozitivnosti števila*

$$\Delta = (u_2v_0 - u_0v_2)^2 - 4(u_0v_1 - u_1v_0)(u_1v_2 - u_2v_1).$$

**Dokaz. DRUGAČE, referenciraj se na hodograf kvintične krivulje.... Odvajaj tisto.... bedno bo, tak je lajf.**

Ker smo za polinom  $w$  izbrali konstanten polinom  $w(t) = 1$ , velja  $x'(t) = u^2(t) - v^2(t)$ ,  $y'(t) = 2u(t)v(t)$ . Če polinoma  $x'$  in  $y'$  odvajamo, dobimo  $x''(t) = 2(u(t) -$

$v(t)$  in  $y''(t) = 2(u'(t)v(t) + u(t)v'(t))$ . Polinome  $x', y', x''$  in  $y''$  vstavimo v enačbo za ukrivljenost  $k = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}}$ . Ker je imenoalec zmeraj pozitiven, nas zanima le števec.

$$x'(t)y''(t) - y'(t)x''(t) = 2(u^2(t) - v^2(t))(u'(t)v(t) + u(t)v'(t)) - 4u(t)v(t)(u(t) - v(t))$$

□

## 4.4 Parametrična hitrost in dolžina krivulje

Parametrična hitrost PH krivulje podane s parametrizacijo  $\mathbf{r}(t) = (x(t), y(t))$  je dana s polinomom

$$\sigma(t) = |\mathbf{r}'(t)| = \sqrt{x'^2(t) + y'^2(t)} = u^2(t) + v^2(t).$$

Če je krivulja stopnje  $n$ , potem morata biti polinoma  $u$  in  $v$  stopnje  $m = \frac{1}{2}(n-1)$ . Zapišimo polinome  $u, v$  in  $\sigma$  v Bernsteinovi obliki

$$u(t) = \sum_{j=0}^m u_j B_j^m, \quad v(t) = \sum_{j=0}^m v_j B_j^m, \quad \sigma(t) = \sum_{j=0}^{n-1} \sigma_j B_{j,m}$$

S pomočjo pravila za množenje Bernsteinovih polinomov iz izreka 2.3, lahko koeficiente  $\sigma_j$  izrazimo s koeficienti polinomov  $u$  in  $v$ :

$$\sigma_j = \sum_{k=\max(0, j-m)}^{\min(m, j)} = \frac{\binom{m}{k} \binom{m}{j-k}}{\binom{n-1}{j}} (u_k u_{j-k} + v_k v_{j-k}).$$

Pridobili smo funkcijo parametrične hitrosti PH krivulje, ki je zapisana v Bernsteinovi bazi. To je zelo super(enadrugabeseda dej), saj znamo vrednosti takšnih polinomov računati hitro\*\*\* in stabilno\*\*\*\*\*, poleg tega pa takšen zapis dopušča enostavno integracijo v zaprte sisteme. Izpeljimo še funkcijo dolžine PH krivulje. Da pridobimo dolžino PH krivulje moramo izračunati določeni itegral polinoma hitrosti  $\int_0^t \sigma(\tau) d\tau = s(t)$ . Integral izračunamo s pomočjo integracijskega pravila iz izreka ???. Tako dobimo dolžino PH krivulje izraženo kot polinom v Bernsteinovi bazi

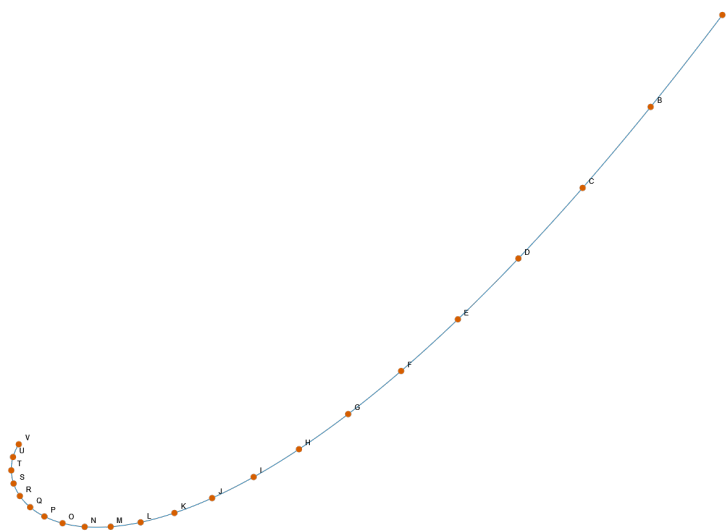
$$s(t) = \sum_{k=0}^n s_k \binom{n}{k} (1-t)^{n-k} t^k,$$

kjer je koeficient  $s_0 = 0$  in koeficienti  $s_k = \frac{1}{n} \sum_{j=0}^{k-1} \sigma_j$ . Iz zapisa je hitro mogoče videti, da je dolžina Bezierjeve PH krivulje enaka  $s(1) = \frac{\sigma_0 + \sigma_1 + \dots + \sigma_{n-1}}{n}$ . Ker velja  $\int_a^b \sigma(t) dt = \int_0^b \sigma(t) dt - \int_0^a \sigma(t) dt = s(b) - s(a)$ , lahko dolžino odseka izračunamo tako, da Bernsteinov polinom  $s$  evalviramo v točkah  $b$  in  $a$  ter vzamemo razliko. To je seveda spet super, ker je takšno računanje hitro in natančno (pri navadnih polinomih ponavadi uporabljamo aproksimacije). Lastnosti funkcije  $s$  bomo v naslednjem podrazdelku uporabili tudi pri enakomerni parametrizaciji.

## 4.5 Enakomerna parametrizacija

MOTIVACIJA: Želimo definirati enakomerno oddaljene točke na krivulji. Takšni problemi se naravno pojavijo pri .....

Krivuljo želimo s parametri  $\{t_i\}_{i=0}^N$  razdeliti na  $N$  kosov. Če to naredimo tako, da parametre  $t_i$  postavimo ekvidistantno, tj.  $\Delta t = t_i - t_{i-1}$  je enak za vse  $i$ , točke na krivulji ne bodo enako oddaljene druga od druge (glede na dolžino odseka med njima), ker parametrična hitrost krivulje  $\sigma(t)$  ni konstantna. Slednje lahko vidimo na sliki 19. Čeprav parametrična hitrost PH krivulje niti ni konstantna, lahko

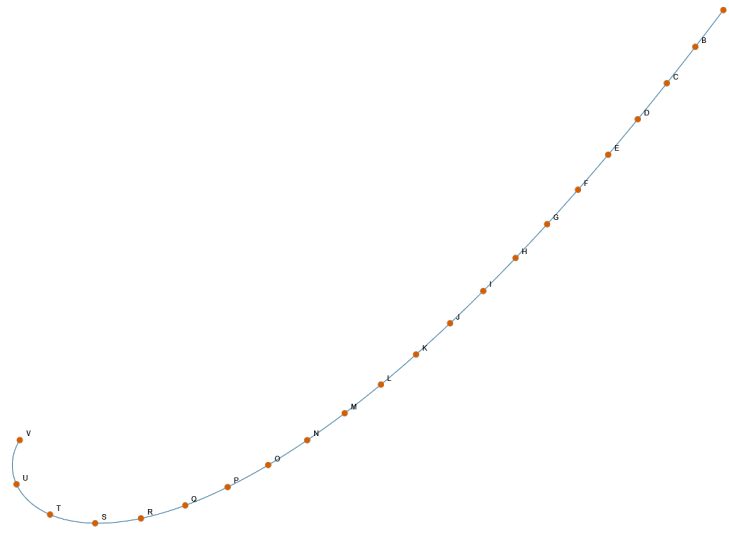


Slika 19: 20 točk na krivulji pri ekvidistantnih  $t_i$

njen enostaven izračun izkoristimo, da najdemo parametre  $\{t_i\}_{i=0}^N$  za katere velja  $s(t_k) = k\Delta s$ , kjer je  $\Delta s = \frac{s(1)}{N}$ . Ker je parametrična hitrost  $\sigma(t) = s'(t)$  pozitivna za vsa realna števila  $t \in [0, 1]$  (ko sta u, v tuja... ampak a ni da to smo mi ze privzeli prej(?) :/), je funkcija dolžine krivulje pri parametru  $t$ ,  $s(t)$ , strogo naraščajoča. Parametri  $t_i$  so zato enolično določeni in ležijo med  $t_{i-1}$  in 1. Uporabimo lahko Newton-Raphsonovo iteracijo. Za začetni približek vzamemo  $t_k^{(0)} = t_{k-1} + \frac{\Delta s}{\sigma(t_{k-1})}$  popravljamo pa ga z iteriranjem

$$t_k^{(r)} = t_k^{(r-1)} - \frac{s(t_k^{(r-1)})}{\sigma(t_k^{(r-1)})}, \quad r = 1, 2, \dots$$

Takšna iteracija pri začetnih približkih, ki so dovolj blizu parametra  $t_k$ , konvergira s kvadratično hitrostjo. (referenca na bor plestenjak? :D) Za večino primerov zato izračun približka  $t_k$ , do natančnosti reda  $10^{-12}$ , potrebuje le dva do tri korake iteracije. V praksi so takšni približki ponavadi zadovoljivi. Točke na PH krivulji pridobljene s takšnimi približki si lahko ogledamo na sliki 20.



Slika 20: 20 točk na krivulji pri ekvidistantnih  $t_i$

## 4.6 Tangenta, normala in ukrivljenost

V prejšnjem razdelku smo pokazali, da je parametrična hitrost PH krivulje (pridobljene točno s tistim integralom...)  $\sigma$  polinom. V tem razdelku bomo pa pokazali, da so tangenta, normala in ukrivljenost PH krivulje racionalne funkcije.

**Izrek 4.4.** *Tangenta, normala in ukrivljenost PH krivulje, glede na parameter  $t$  so izražene kot racionalne funkcije*

$$\mathbf{t} = \frac{(u^2 - v^2, 2uv)}{\sigma}, \quad \mathbf{n} = \frac{(2uv, v^2 - u^2)}{\sigma}, \quad \kappa = 2 \frac{uv' - u'v}{\sigma^2}.$$

Izrek tudi dokažimo.

*Dokaz.* Dokaza za tangento in normalo sta enostavna, saj hodograf REFNAHODOGRAFA le vstavimo v enačbi in upoštevamo, da je polinom  $\sigma = u^2 + v^2$  da dobimo željeno. Dokažimo za ukrivljenost  $\kappa$ . V enačbo za ukrivljenost  $\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}}$  vstavimo hodograf REFNAHODOGRAFA, ter druga odvoda  $x'' = 2(uu' - vv')$  in  $y'' = 2(u'v + uv')$ .

$$\begin{aligned} \kappa &= 2 \frac{(u^2 - v^2)(u'v + uv') - 2uv(uu' - vv')}{(\sigma^2)^{3/2}} \\ &= 2 \frac{u^2vu' + u^3v' - v^3u' - v^2uv' - 2u^2vu' + 2uv^2v'}{\sigma^3} \\ &= 2 \frac{(u^2 - v^2 + 2v^2)uv' - (-u^2 + v^2 + 2u^2)u'v}{\sigma^3} \\ &= 2 \frac{(u^2 + v^2)uv' - (u^2 + v^2)u'v}{\sigma^3} \\ &= 2 \frac{(u^2 + v^2)(uv' - u'v)}{\sigma^3} = 2 \frac{uv' - u'v}{\sigma^2} \end{aligned}$$

□

## 4.7 Racionalne odmične krivulje

Odmična krivulja z razdaljo  $d$  od krivulje  $\mathbf{r}(t)$ , je krivulja podana s parametrizacijo

$$\mathbf{r}_d(t) = \mathbf{r}(t) + d\mathbf{n}(t).$$

Takšne krivulje v splošnem niso racionalne, saj v imenovalcu enačbe za smer normale  $\mathbf{n}(t)$  nastopa koren. V prejšnjem razdelku smo pokazali, da je smer normale  $\mathbf{n}(t)$  za PH krivulje racionalna funkcija, iz česar sledi, da je tudi odmična krivulja PH krivulje racionalna funkcija. Še več, takšno odmično krivuljo lahko izrazimo kot racionalno Bezierjevo krivuljo s kontrolnimi točkami, ki jih lahko izrazimo s kontrolnimi točkami PH krivulje. Zapišimo kontrolne točke PH krivulje  $\mathbf{r}(t)$  v homogenih koordinatah

$$\mathbf{P}_k = (W_k, X_k, Y_k) = (1, x_k, y_k), \quad k = 0, 1, \dots, n.$$

Preme difference takšnih točk izgledajo takole

$$\Delta \mathbf{P}_k = \mathbf{P}_{k+1} - \mathbf{P}_k = (0, \Delta x_k, \Delta y_k), \quad k = 0, 1, \dots, n-1.$$

Označimo še pravokotno smer  $\Delta \mathbf{P}_k^\perp = (0, \Delta y_k, -\Delta x_k)$ . Parametrizacijo odmične krivulje lahko izrazimo kot

$$\mathbf{r}_d(t) = \left( \frac{X(t)}{W(t)}, \frac{Y(t)}{W(t)} \right),$$

kjer so polinomi  $W, X$  in  $Y$  stopnje  $2n-1$ , njihovi koeficienti (v Bernsteinovi bazi??)

$$\mathbf{O}_k = (W_k, X_k, Y_k), \quad k = 0, \dots, 2n-1$$

pa definirajo kontrolne točke odmične racionalne Bezierjeve krivulje v homogenih koordinatah. Homogene koordinate odmične krivulje  $\mathbf{O}_k$  lahko v zaključeni obliki izrazimo s kontrolnimi točkami prvotne krivulje

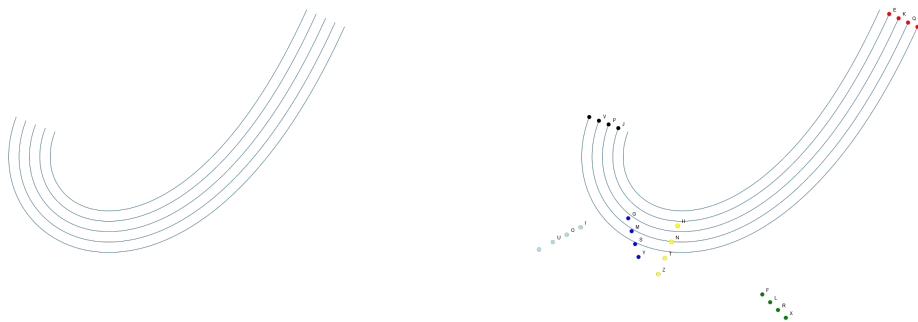
$$\mathbf{O}_k = \sum_{j=\max(0, k-n)}^{\min(n-1, k)} \frac{\binom{n-1}{j} \binom{n}{k-j}}{\binom{2n-1}{k}} (\sigma_j \mathbf{P}_{k-j} + d\mathbf{n} \Delta \mathbf{P}_j^\perp), \quad k = 0, \dots, 2n-1.$$

Za pokušino si sedaj oglejmo kako izgledajo kontrolne točke odmične krivulje kubične PH krivulje. **Ostalih stopenj ne bomo razpisali, saj bomo v nadaljevanju predstavili zapis, ki bo vse to olajšal.**

$$\begin{aligned} \mathbf{O}_0 &= \sigma_0 \mathbf{P}_0 + 3d\Delta \mathbf{P}_0^\perp, \\ \mathbf{O}_1 &= \frac{1}{5} [2\sigma_1 \mathbf{P}_0 + 3\sigma_0 \mathbf{P}_1 + 3d(3\Delta \mathbf{P}_0^\perp + 2\Delta \mathbf{P}_1^\perp)], \\ \mathbf{O}_2 &= \frac{1}{10} [\sigma_2 \mathbf{P}_0 + 6\sigma_1 \mathbf{P}_1 + 3\sigma_0 \mathbf{P}_2 + 3d(3\Delta \mathbf{P}_0^\perp + 6\Delta \mathbf{P}_1^\perp + \Delta \mathbf{P}_2^\perp)], \\ \mathbf{O}_3 &= \frac{1}{10} [3\sigma_2 \mathbf{P}_1 + 6\sigma_1 \mathbf{P}_2 + 3\sigma_0 \mathbf{P}_3 + 3d(\Delta \mathbf{P}_0^\perp + 6\Delta \mathbf{P}_1^\perp + 3\Delta \mathbf{P}_2^\perp)], \\ \mathbf{O}_4 &= \frac{1}{5} [3\sigma_2 \mathbf{P}_2 + 2\sigma_1 \mathbf{P}_3 + 3d(2\Delta \mathbf{P}_1^\perp + 3\Delta \mathbf{P}_2^\perp)], \\ \mathbf{O}_5 &= \sigma_2 \mathbf{P}_3 + 3d\Delta \mathbf{P}_2^\perp \end{aligned}$$



Tukaj lahko razpišem  $O_k$  za kvintične in kubične PH. Dodam slike različnih offsetov in kontrolne poligone. Lahko dodam v program, da se narisejo te crtkane crte po katerih se premikajo kontrolne točke te odmične krivulje... Na sliki 21 lahko vidimo, da se kontrolne točke odmične krivulje



Slika 21: Odmične krivulje (levo) skupaj z njihovimi kontrolnimi točkami (desno)

v odvisnosti od razdalje premikajo po premicah.

## 5 Orodje za grafični prikaz konceptov

Vsi koncepti predstavljeni v magistrskem delu so tudi implementirani na spletni strani. Slikovno gradivo skozi magistrsko delo je bilo ustvarjeno s pomočjo orodja. Za risanje grafov sem uporabil odprtokodno knjižnico JSxGraph. Za oblikovanje spletne strani sem uporabil ogrodje bootstrap. Za ogrodje same spletne strani pa React.

### 5.1 Implementacija konceptov magistrskega dela

### 5.2 Orodja

Na levi strani vsakega grafa se nahaja kartica 22 s funkcionalnostmi za vse grafe. V naslednjih podrazdelkih bomo predstavili posamezno funkcionalnost.

#### 5.2.1 Izvoz slik

Uporabnik lahko s stiskom na gumb SVG oziroma PNG izvozi sliko v izbranem formatu. Pri izvozu v SVG format, zaradi pomankljivosti knjižnice JSXGraph, izgubimo oznake točk.

#### 5.2.2 Ponastavitev grafa

Na kartici se nahaja priročen gumb za ponastavljanje grafa. Ob pritisku se graf ponastavi na izbrano prednastavitev.

#### 5.2.3 Prikaz mreže

Če uporabnik ne želi prikazovati mreže koordinatnega sistema, lahko za to uporabi preklopnik za prikaz mreže.

#### 5.2.4 Povečava

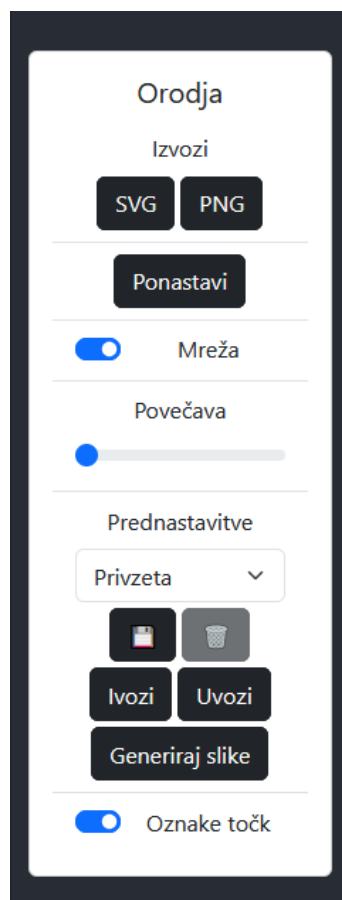
Drsnik za povečavo omogoča povečavo elementov na grafu.

#### 5.2.5 Prednastavitve

Uporabnik lahko na posameznem grafu pripravi željeni prikaz in ga nato s poljubnim imenom shrani kot prednastavitev. Prednastavitve se shranjujejo lokalno v uporabnikovem brskalniku. Če uporabnik želi svoje prednastavitve prenesti na drug računalnik, lahko to stori tako, da pritisne gumb izvozi. S pritiskom na gumb se sproži prenos datoteke s podatki o prednastavitvah. Na drugem računalniku lahko nato datoteko uvozi s pritiskom na gumb uvozi. Uporabnik ima tudi možnost generirati slike svojih prednastavitev. Ob pritisku na gumb 'Generiraj slike' začne orodje preklapljati med prednastavitvami ter izvažati slike. Slike izvozi pod imenom prednastavitve, v PNG formatu.

#### 5.2.6 Preklopnik za oznake točk

Preklopnik uporabniku omogoča izklop oznak točk na grafu.



Slika 22: Kartica z orodji

### 5.3 Ukazi izbrane krivulje

Na desni strani vsakega grafa se, v primeru, da imamo izbrano krivuljo, prikazujejo ukazi izbrane krivulje. V naslednjih podrazdelkih bomo predstavili katere ukaze premore kateri tip krivulje.

#### 5.3.1 Vse krivulje

Vse krivulje omogočajo preklapljanje prikazovanja kontrolnega poligona, konveksne ovojnice kontrolnih točk, ter kontrolnih točk samih.

#### 5.3.2 Bezierjeva krivulja

Bezierjeva krivulja premore poleg osnovnih ukazov še ukaze za subdivizijo, ekstrapolacijo, skrčitev, višanje stopnje, ter prikazovanje deCasteljaujeve sheme. Premik miške na posamezni ukaz na grafu predstavi nekakšen predogled rezultata ukaza.

#### 5.3.3 Racionalna Bezierjeva Krivulja

Poleg funkcionalnosti bezierjeve krivulje ima še dodane ukaze za prikaz in spreminjanje uteži, ter drsnik za preklapljanje forme uteži.

#### **5.3.4 PH Bezierjeva krivulja**

PH Bezierjeve krivulje nimajo ukazov, ki so jih imele Bezierjeve krivulje, saj se pri nekaterih ukazih izgubi lastnost PH. Premorejo pa naslednje ukaze. Prikazovanje odmičnih krivulj in spreminjanje odmika. Prikaz kontrolnih točk odmik krivulj ter prikazovanje premic na katerih ležijo. Dodajanje/odstranjevanje odmik krivulj. Koordinatni sistem s hodografi, ki jih premikamo za premikanje krivulje. Gumba za povečanje ter pomanjšanje krivulje.

#### **5.3.5 Zlepki**

Zlepki imajo le ukaze osnovne krivulje

## 6 Integrali po $\omega$ -kompleksih

### 6.1 Definicija

**Definicija 6.1.** Neskončno zaporedje kompleksnih števil, označeno z  $\omega = (\omega_1, \omega_2, \dots)$ , se imenuje  $\omega$ -kompleks.<sup>1</sup>

Črni blok zgoraj je tam namenoma. Označuje, da L<sup>A</sup>T<sub>E</sub>X ni znal vrstice prelomiti pravilno in vas na to opozarja. Preoblikujte stavek ali mu pomagajte deliti problematično besedo z ukazom `\hyphenation{an-ti-ko-mu-ta-ti-ven}` v preambuli.

**Trditev 6.2** (Znano ime ali avtor). *Obstaja vsaj en  $\omega$ -kompleks.*

*Dokaz.* Naštejmo nekaj primerov:

$$\begin{aligned}\omega &= (0, 0, 0, &= \dots), & (6.1) \\ \omega &= (1, i, &= -1, -i, 1, \dots), \\ \omega &= (0, &= 1, 2, 3, \dots). & \square\end{aligned}$$

---

<sup>1</sup>To ime je izmišljeno.

