

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

1	Uvod	1
2	Bezierjeve krivulje	1
2.1	Bernsteinovi polinomi	1
2.2	Večdimenzionalne oznake	3
2.3	Bezierjeve krivulje	3
2.4	Decasteljau	4
2.5	Metode Bezierjevih krivulj	4
2.5.1	Subdivizija	5
2.5.2	Ekstrapolacija	6
2.5.3	Višanje stopnje	6
2.6	Racionalne Bezierjeve krivulje	9
2.6.1	Metode racionalnih Bezierjevih krivulj	11
2.6.2	Decasteljaujev algoritem za racionalne Bezierjeve krivulje . . .	11
2.6.3	Izsek krožnice	11
3	Zlepki Bezierjevih krivulj	13
3.1	Alfa parametrizacije	15
4	PH Krivulje	16
4.1	Dolžina krivulje kot racionalna funkcija	16
4.2	Polinomska parametrična hitrost	16
4.3	Kontrolne točke Bezierjevih PH krivulj	18
4.4	Parametrična hitrost in dolžina krivulje	20
4.5	Enakomerna parametrizacija	21
4.6	Tangenta, normala in ukrivljenost	22
4.7	Racionalne odmične krivulje	23
5	Orodje za grafični prikaz konceptov	24
5.1	Implementacija konceptov magistrskega dela	24
6	Integrali po ω-kompleksih	25
6.1	Definicija	25

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.

2 Bezierjeve krivulje

V tem razdelku bomo predstavili Bezierjeve krivulje in njihove ključne lastnosti za rabo v CAGD sistemih. Začeli bomo z Bernsteinovimi polinomi, ki jih bomo uporabili pri definiciji Bezierjevih krivulj. Predstavili bomo Decasteljaujev algoritem, ki omogoča stabilno računanje točk Bezierjevih krivulj. Nadaljevali pa bomo z nekaj metodami na Bezierjevih krivuljah, ki so ključne za njihovo rabo v računalniško podprtem grafičnem oblikovanju.

2.1 Bernsteinovi polinomi

Bernsteinove polinome je najprej uporabil Sergei Bernstein pri dokazu Weierstrasovega izreka. Kasneje jih je Pierre Bezier uporabil pri definiciji Bezierjeve krivulje. V tem razdelku bomo predstavili nekaj njihovih osnovnih lastnosti, ki jih bomo uporabili pri Bezierjevih krivuljah. Dejmo jih najprej definirati.

Definicija 2.1. Za pozitivno naravno število n je i -ti *Bernsteinov bazni polinom* enak

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

Linearni kombinaciji takšnih polinomov t.j. $\sum_{i=0}^n \beta_i B_i^n(t)$, pravimo *Bernsteinov polinom* stopnje n . V izreku 2.2 naštejemo nekaj lastnosti Bernsteinovih baznih polinomov, ki so ključne za rabo v CAD sistemih (**zato da so bez krivulje sploh uporabne**).

Izrek 2.2. *Lastnosti Bernsteinovih polinomov*

1. $B_i^n(t) = 0$ za $i < 0$ ali $i > n$, interpolacija končnih točk
2. $B_i^n(t) \geq 0$ za $t \in [0, 1]$, pozitivnost
3. $B_i^n(0) = \delta_{i,0}$ in $B_i^n(1) = \delta_{i,n}$, kjer je $\delta_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$
4. $B_i^n(1-t) = B_{n-i}^n(t)$, simetrija
5. $\sum_{i=0}^n B_i^n(t) = 1$, razčlenitev enote
6. $B_i^n(t) = (1-t)B_i^{n-1} + tB_{i-1}^{n-1}$
7. $B_i^n(t) = n(B_{i-1}^{n-1} - B_i^{n-1})$
8. $\int B_i^n(t) dt = \frac{1}{n+1} \sum_{j=i+1}^{n+1} B_j^{n+1}(t)$

9. Naj bo $f(t) = \sum_{i=0}^m B_i^m(t)$ in $g(t) = \sum_{i=0}^n B_i^n(t)$. 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. Točki (1) in (2) očitno izhajata iz lastnosti binomskega simbola. Dokažimo ostale.

(3) $B_0^n(0) = \dots$

(4) Namesto spremenljivke t v enačbo za bernsteinov bazni polinom $B_i^n(t)$ vstavimo izraz $1 - t$ in uporabimo lastnost binomskega simbola $\binom{n}{i} = \binom{n}{n-i}$, dobimo

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

(5) Za $1 = 1^n = (1-t+t)^n = ((1-t)+t)^n$ uporabimo binomski izrek, dobimo

$$((1-t)+t)^n = \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} = \sum_{i=0}^n B_i^n(t).$$

(6) Uporabili bomo lastnost binomskega simbola $\binom{n-1}{i} + \binom{n-1}{i-1} = \binom{n}{i}$.

$$\begin{aligned} (1-t)B_i^{n-1} + tB_{i-1}^{n-1} &= \\ &= (1-t) \binom{n-1}{i} t^i (1-t)^{n-i-1} + t \binom{n-1}{i-1} t^{i-1} (1-t)^{n-i} = \\ &= \binom{n-1}{i} t^i (1-t)^{n-i} + \binom{n-1}{i-1} t^i (1-t)^{n-i} = \\ &= \binom{n}{i} t^i (1-t)^{n-i} = \\ &= B_i^n(t) \end{aligned} \tag{2.1}$$

Dodaj dokaze!!

(9) Naj bosta $f(t) = \sum_{i=0}^m \alpha_i B_i^m(t)$ in $g(t) = \sum_{i=0}^n \beta_i B_i^n(t)$ Bernsteinova polinoma.

$$\begin{aligned} f(t)g(t) &= \sum_{i=0}^m \alpha_i B_i^m(t) \sum_{j=0}^n \beta_j B_j^n(t) = \sum_{k=0}^{m+n} \sum_{l=0}^k \alpha_l B_l^m(t) \beta_{k-l} B_{k-l}^n(t) \\ &= \sum_{k=0}^{m+n} \sum_{l=0}^k \alpha_l \binom{m}{l} t^l (1-t)^{m-l} \beta_{k-l} \binom{n}{k-l} t^{k-l} (1-t)^{n-k+l} \\ &= \sum_{k=0}^{m+n} \sum_{l=0}^k \alpha_l \beta_{k-l} \binom{m}{l} \binom{n}{k-l} t^k (1-t)^{m+n-k} \\ &= \sum_{k=0}^{m+n} \left(\sum_{l=0}^k \alpha_l \beta_{k-l} \frac{\binom{m}{l} \binom{n}{k-l}}{\binom{m+n}{k}} \right) \binom{m+n}{k} t^k (1-t)^{m+n-k}. \end{aligned}$$

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

2.2 Večdimenzionalne oznake

Z željo po krajših, bolj preglednih zapisih, bomo uvedli večdimenzionalne oznake. Večdimenzionalnost bomo ponazarjali z odebelitvijo črke. Tako bomo večdimenzionalnost točke označili z $\mathbf{x} = (x_0, x_1, \dots, x_n)$, večdimenzionalnost funkcije $f : \mathbb{R} \rightarrow \mathbb{R}^{n+1}$ pa z $\mathbf{f}(x) = (f_0(x), f_1(x), \dots, f_n(x))$.

2.3 Bezierjeve krivulje

Če v Bernsteinov polinom n -te stopnje namesto skalarjev β_i vstavimo točke $\mathbf{p}_i \in \mathbb{R}^d$, dobimo predpis za parametrizacijo Bezierjeve krivulje.

Definicija 2.3. Bezierjeva krivulja n -te stopnje je podana s točkami \mathbf{p}_i 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*, poligonu, ki ga dobimo, če povežemo točke \mathbf{p}_i in \mathbf{p}_{i+1} za $i = 0, \dots, n-1$, ter točki \mathbf{p}_0 in \mathbf{p}_n pa pravimo *kontrolni poligon*. **Opomba:** pri slikovnem prikazovanju raznih konceptov Bezierjevih krivulj, se bomo omejili na $d = 2$, torej na Bezierjeve krivulje v ravnini.

Izrek 2.4. Lastnosti Bezierjevih krivulj

1. $\mathbf{B}_n(0) = \mathbf{p}_0$ in $\mathbf{B}_n(1) = \mathbf{p}_n$, interpolacija končnih točk
2. $\phi(\sum_{i=0}^n \mathbf{p}_i B_i^n(t)) = \sum_{i=0}^n \phi(\mathbf{p}_i) B_i^n(t)$, afina invarianca
3. Krivulja leži znotraj konveksne ovojnice svojih kontrolnih točk.

Preden izrek dokažemo, povejmo zakaj so zgornje lastnosti pomembne za CAGD sisteme. Interpolacija končnih točk omogoča, da uporabniku damo kontrolo nad tem, kje se krivulja začne in kje konča. Afina invarianca je pomembna za to, da lahko uporabnik transformira krivuljo, saj se uporabnikove transformacije krivulje enostavno prevedejo v afine preslikave kontrolnih točk. Tretja lastnost pa uporabniku s kontrolnimi točkami omogoča kontrolo nad krivuljo, kjer je krivulja zmerom v bližini svojih kontrolnih točk.

Dokaz.

(1) $\mathbf{B}_n(0) = \sum_{i=0}^n \mathbf{p}_i B_i^n(t)(0) = \sum_{i=0}^n \mathbf{p}_i \delta_{0,i} = \mathbf{p}_0$. Enako lahko naredimo tudi za $\mathbf{B}_n(1)$.

(2) Naj bo ϕ afina preslikava, velja torej $\phi(x) = A\mathbf{x} + \mathbf{b}$.

$$\begin{aligned} \phi\left(\sum_{i=0}^n \mathbf{p}_i B_i^n(t)(t)\right) &= A\left(\sum_{i=0}^n \mathbf{p}_i B_i^n(t)(t)\right) + \mathbf{b} &&= \sum_{i=0}^n A\mathbf{p}_i B_i^n(t)(t) + \mathbf{b} \\ &= \sum_{i=0}^n A\mathbf{p}_i B_i^n(t)(t) + \sum_{i=0}^n \mathbf{b} B_i^n(t)(t) &&= \sum_{i=0}^n (A\mathbf{p}_i + \mathbf{b}) B_i^n(t)(t) \\ &= \sum_{i=0}^n \phi(\mathbf{p}_i) B_i^n(t) \end{aligned}$$

(3) Konveksna ovojnica kontrolnih točk Bezierjeve krivulje je množica vseh konveksnih kombinacij teh točk t.j. $\sum_{i=0}^n \lambda_i \mathbf{p}_i$, kjer so λ_i pozitivna realna števila za katere velja $\lambda_0 + \lambda_1 + \dots + \lambda_n = 1$. Ker so Bernsteinovi polinomi za poljuben $t \in [0, 1]$ razčlenitev enote in velja $B_i^n(t) \geq 0$, lahko zapišemo $\lambda_i = B_i^n(t)$. \square

2.4 Decasteljau

Računanje Bernsteinovih polinomov direktno preko njihovih enačb je precej nestabilno****, za rabo v računalništvu pa je pomembno, da uporabljamo stabilne metode računanja. S pomočjo Decasteljauevega algoritma lahko računamo točke Bezierjevih krivulj stabilno****, zanj pa potrebujemo naslednji izrek.

Izrek 2.5. *Naj bo $\mathbf{B}(t)_{[\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n]}$ parametrizacija Bezierjeve krivulje n -te stopnje s kontrolnimi točkami $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$. Potem lahko njene točke rekurzivno računamo s pomočjo naslednjega izraza*

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

Izrek tudi dokažimo.

Dokaz.

$$\begin{aligned} (1-t)\mathbf{B}(t)_{[\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{n-1}]} + t\mathbf{B}(t)_{[\mathbf{p}_1, \dots, \mathbf{p}_n]} &= \\ &= (1-t) \sum_{i=0}^{n-1} \mathbf{p}_i B_i^{n-1}(t) + t \sum_{i=0}^{n-1} \mathbf{p}_{i+1} B_i^{n-1} \\ &= (1-t) \sum_{i=0}^{n-1} \mathbf{p}_i B_i^{n-1}(t) + t \sum_{i=1}^n \mathbf{p}_i B_{i-1}^{n-1} \\ &= \mathbf{p}_0(1-t)B_0^{n-1}(t) + \sum_{i=1}^{n-1} \mathbf{p}_i(1-t)B_i^{n-1} + \sum_{i=1}^{n-1} \mathbf{p}_i t B_{i-1}^{n-1} + \mathbf{p}_n B_{n-1}^{n-1}(t) \\ &= \mathbf{p}_0(1-t)B_0^{n-1}(t) + \sum_{i=1}^{n-1} \left((1-t)B_i^{n-1} + t B_{i-1}^{n-1} \right) \mathbf{p}_i + \mathbf{p}_n B_{n-1}^{n-1}(t) \\ &= \mathbf{p}_0 B_0^n(t) + \sum_{i=1}^{n-1} \mathbf{p}_i B_i^n(t) + \mathbf{p}_n B_n^n(t) \\ &= \sum_{i=0}^n \mathbf{p}_i B_i^n(t) \end{aligned}$$

\square

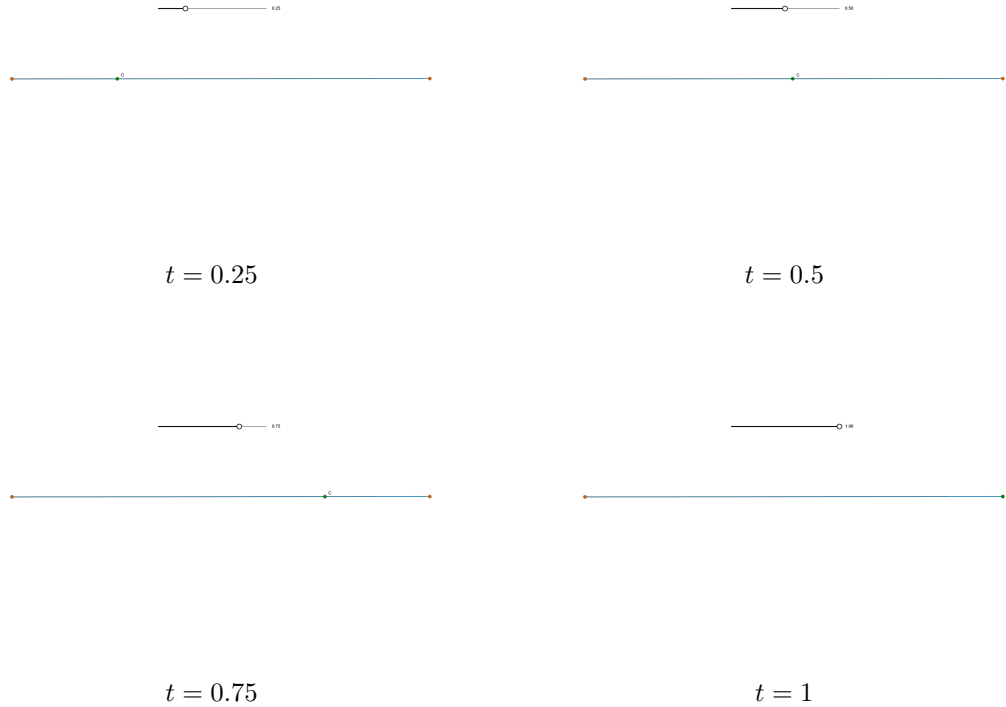
Decasteljauev algoritem ima tudi geometrijski pomen. Pri $n = 1$ se algoritem prevede na interpolacijo dveh točk, kar lahko vidimo na sliki1. Prvi višjih n , pa lahko na slikah 2, 3 in 4 vidimo interpolacijo sosednjih točk na r -tem nivoju.

2.5 Metode Bezierjevih krivulj

V tem razdelku bomo predstavili nekaj metod Bezierjevih krivulj, ki so uporabne pri grafičnem oblikovanju.

Algoritem 1 Decasteljau

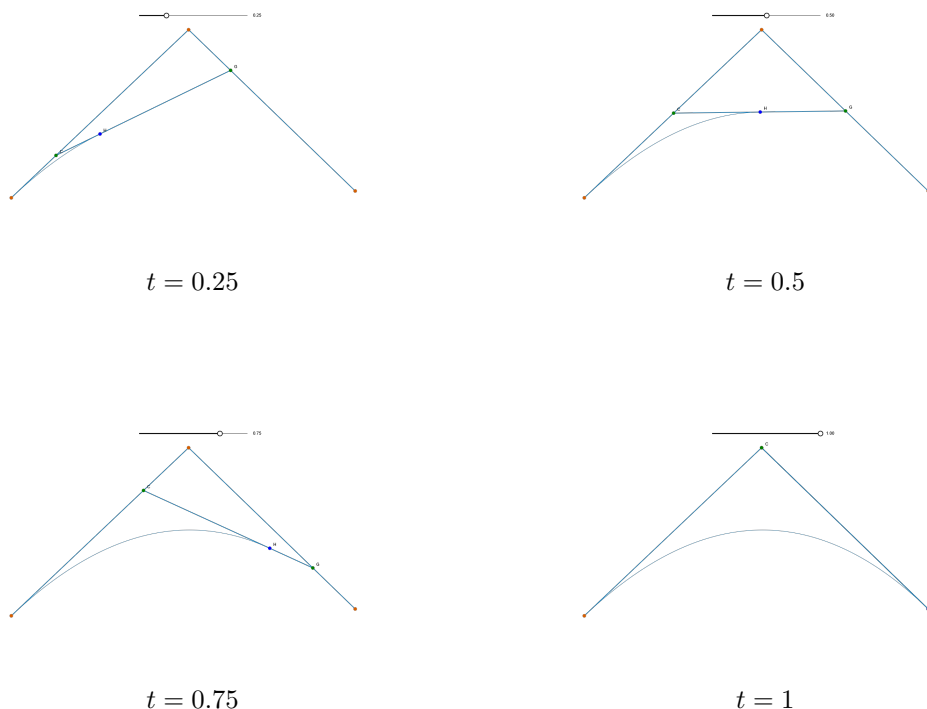
```
 $\mathbf{p} \leftarrow \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$ 
for  $i = 0, 1, \dots, n$  do
   $\mathbf{p}_i^0(t) = \mathbf{p}_i$ 
end for
for  $r = 1, 2, \dots, n$  do
  for  $i = 0, 1, \dots, n - r$  do
     $\mathbf{p}_i^r(t) = (1 - t)\mathbf{p}_i^{r-1}(t) + t\mathbf{p}_{i+1}^{r-1}(t)$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



Slika 1: Decasteljaujev algoritem za $n = 1$

2.5.1 Subdivizija

Recimo, da smo se kot računalniški grafik znašli v situaciji, ko bi želeli obdržati le en kos Bezierjeve krivulje B . Naj bo to kos krivulje, ki ga dobimo tako, da za prvotno krivuljo B omejimo parameter t na interval $[0, t_0]$ za neko fiksno realno število $t_0 < 1$. Označimo ga z B_{t_0} , zanj pa velja $B_{t_0}(t) = B(t_0 t)$. Vrnimo se nazaj k sliki sheme Decasteljaujevega algoritma 5. Na njej lahko opazimo, da pri parametru t_0 točke označene z A, F, J, M in O tvorijo kontrolni poligon bezierjeve krivulje B_{t_0} . Pokažimo, da je to res.



Slika 2: Decasteljaujev algoritem za $n = 2$

$$B_{t_0}(t) = B(t_0 t) = \sum_{i=0}^n \mathbf{p}_i \binom{n}{i} (t_0 t)^i (1 - t t_0)^{n-i}$$

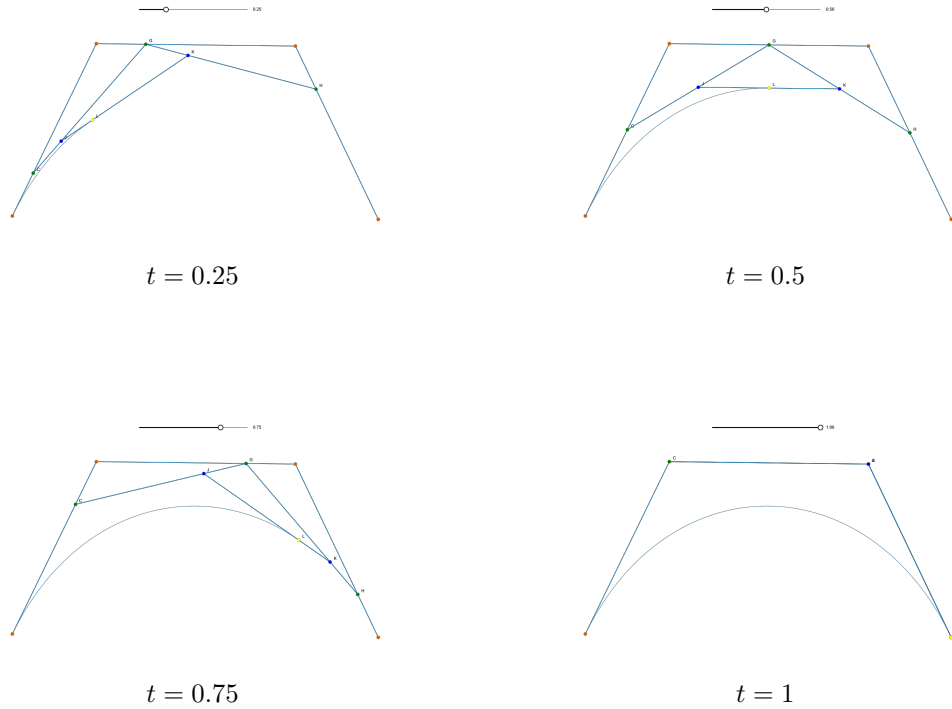
$$\sum_{i=0}^n \mathbf{p}_i t_0^i \left(\frac{1 - t t_0}{1 - t} \right)^{n-i} \binom{n}{i} t^i (1 - t)^{n-i}$$

2.5.2 Ekstrapolacija

Motivacija: Zelimo podaljsati krivuljo..... Kr neki. Realno je to bolj kot ne nekaj, kar lahko kasneje uporabimo pri izpeljavi cesa, Graficno je pa to skoraj useless in se pokaze samo kot nekaj, kar tudi deluje.

2.5.3 Višanje stopnje

TODO: DOKAZ NI PRAVILEN, nekje sem neki zajebal, malce zamakjeno bi moralo bit! Nekateri algoritmi, ki delajo z Bezierjevimi krivuljami, za vhod potrebujejo dve Bezierjevi krivulji iste stopnje. Recimo, da imamo Bezierjevo krivuljo B stopnje n , ki jo želimo spraviti na stopnjo $n + 1$. Ker velja $1 - t + t = 1$ lahko Bezierjevo krivuljo $B(t)$ zapišemo tudi kot $B(t) = (1 - t)B(t) + tB(t) = \sum_{i=0}^n \mathbf{p}_i (1 - t)B_i^n(t) +$



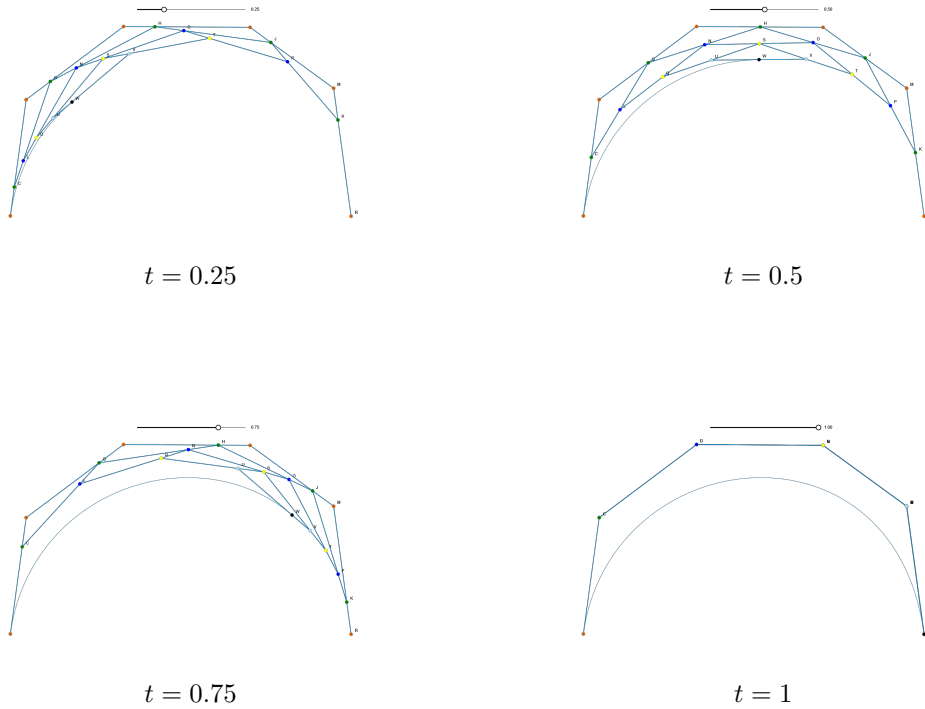
Slika 3: Decasteljaujev algoritem za $n = 3$

$\sum_{i=0}^n \mathbf{p}_i t B_i^n(t)$. Razpišimo najprej prvi del.

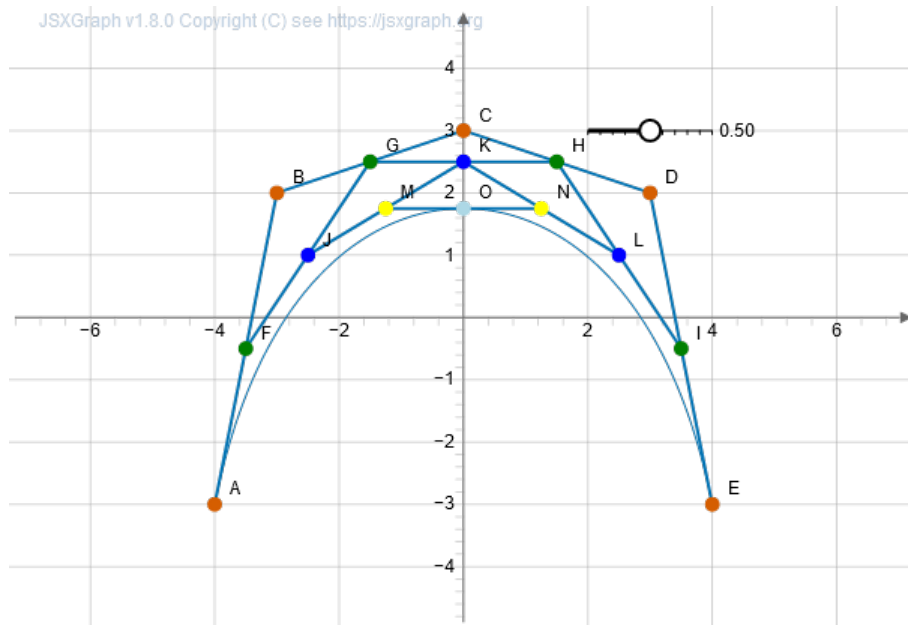
$$\begin{aligned}
 (1-t)B_i^n(t) &= (1-t) \binom{n}{i} t^i (1-t)^{n-i} \\
 &= \frac{n!}{(n-i)!i!} t^i (1-t)^{n+1-i} \\
 &= \frac{n+1-i}{n+1} \frac{(n+1)!}{(n+1-i)!i!} t^i (1-t)^{n+1-i} \\
 &= \frac{n+1-i}{n+1} \binom{n+1}{i} t^i (1-t)^{n+1-i} \\
 &= \frac{n+1-i}{n+1} B_i^{n+1}(t)
 \end{aligned}$$

Sedaj pa še za drugega.

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



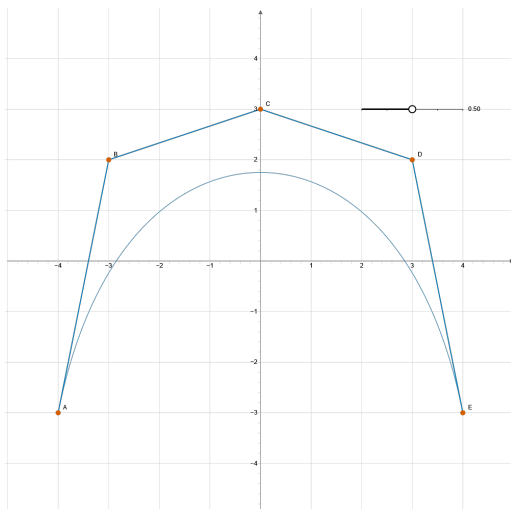
Slika 4: Decasteljaujev algoritem za $n = 5$



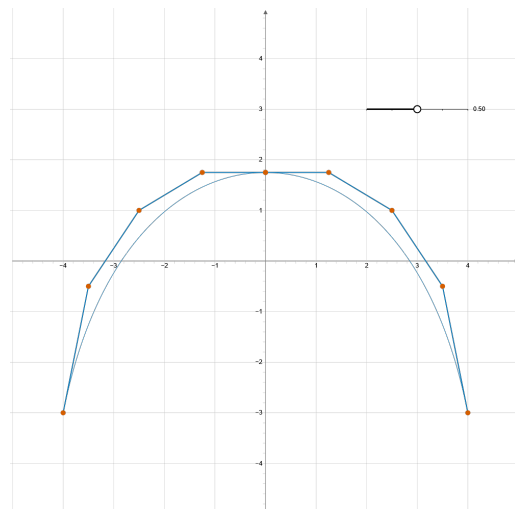
Slika 5: Subdivizija

Vstavimo v $B(t)$.

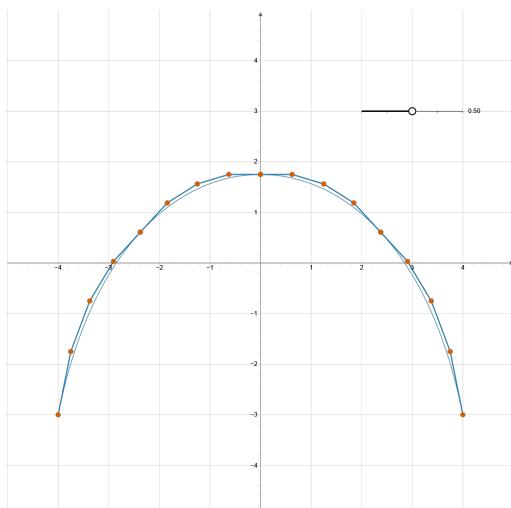
$$\begin{aligned}
 B(t) &= \sum_{i=0}^n \mathbf{p}_i \frac{n+1-i}{n+1} B_i^{n+1}(t) + \sum_{i=0}^n \mathbf{p}_i \frac{i+1}{n+1} B_{i+1}^{n+1}(t) \\
 &= \sum_{i=0}^n \mathbf{p}_i \frac{n+1-i}{n+1} B_i^{n+1}(t) + \sum_{i=1}^{n+1} \mathbf{p}_{i-1} \frac{i}{n+1} B_i^{n+1}(t) \\
 &= \mathbf{p}_0 + \sum_{i=1}^n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



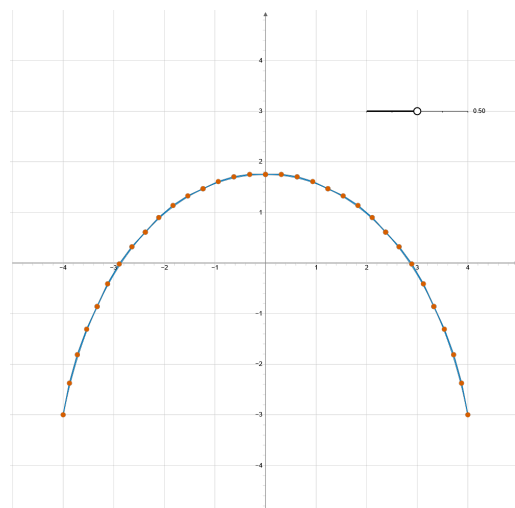
(a) Začetna krivulja s kontrolnim polinomom



(b) 2 krivulji, pridobljeni s subdivizijo



(c) 4 krivulje, pridobljene s subdivizijo



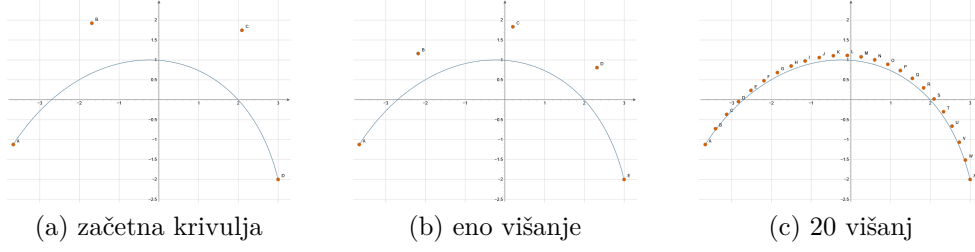
(d) 8 krivulj, pridobljenih s subdivizijo

Slika 6: Ponavljanje subdivizije na krivulji

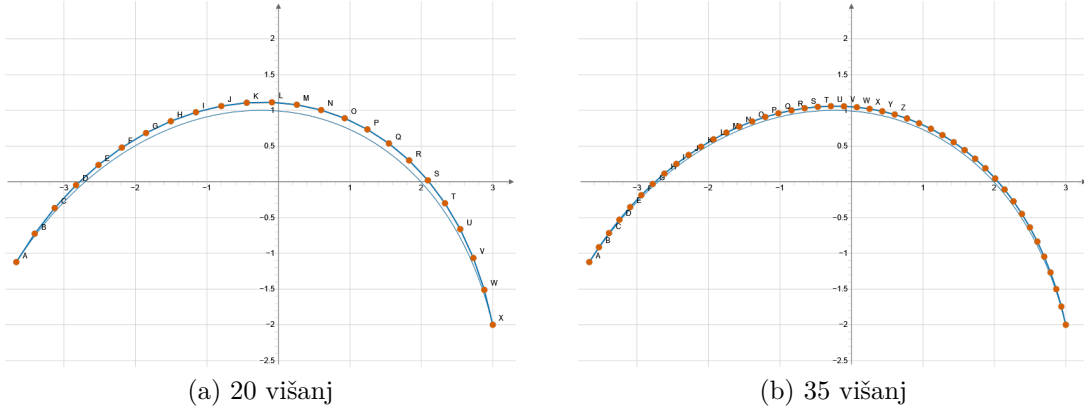
Oglejmo si kako višanje stopnje izgleda na neki krivulji. Na sliki 7 imamo na prvem grafu narisano začetno Bezierjevo krivuljo stopnje 3. Na drugem grafu, smo stopnjo začetne krivulje zvišali za 1. Na tretjem, pa smo naredili 20 višanj stopnje začetne krivulje. Krivulja je na vseh grafih enaka, imamo le več kontrolnih točk. Opaziti je tudi možno, da so kontrolne točke z vsakim višanjem bližje začetni krivulji, njihov kontrolni poligon pa se zato z vsakim višanjem bolj prilega začetni krivulji. Slednje je možno videti na sliki 8.

2.6 Racionalne Bezierjeve krivulje

Vseh krivulj se ne da opisati z navadnimi Bezierjevimi krivuljami. Med njimi so tudi takšne, ki so za CAGD sisteme zelo pomembne, na primer razni izseki krožnice. Za opis takšnih krivulj, lahko posežemo po racionalnih Bezierjevih krivuljah. *Racionalno Bezierjevo krivuljo* stopnje n v \mathbb{R}^d dobimo tako, da polinomske Bezierjevo



Slika 7: Višanje Bezierjeve krivulje



Slika 8: Prileganje kontrolnega poligona

krivuljo stopnje n v \mathbb{R}^{d+1} projiciramo na hiperravnino $w = 1$. Projekcijo definiramo s predpisom $(w, \mathbf{x}) \rightarrow (1, \frac{\mathbf{x}}{w})$.

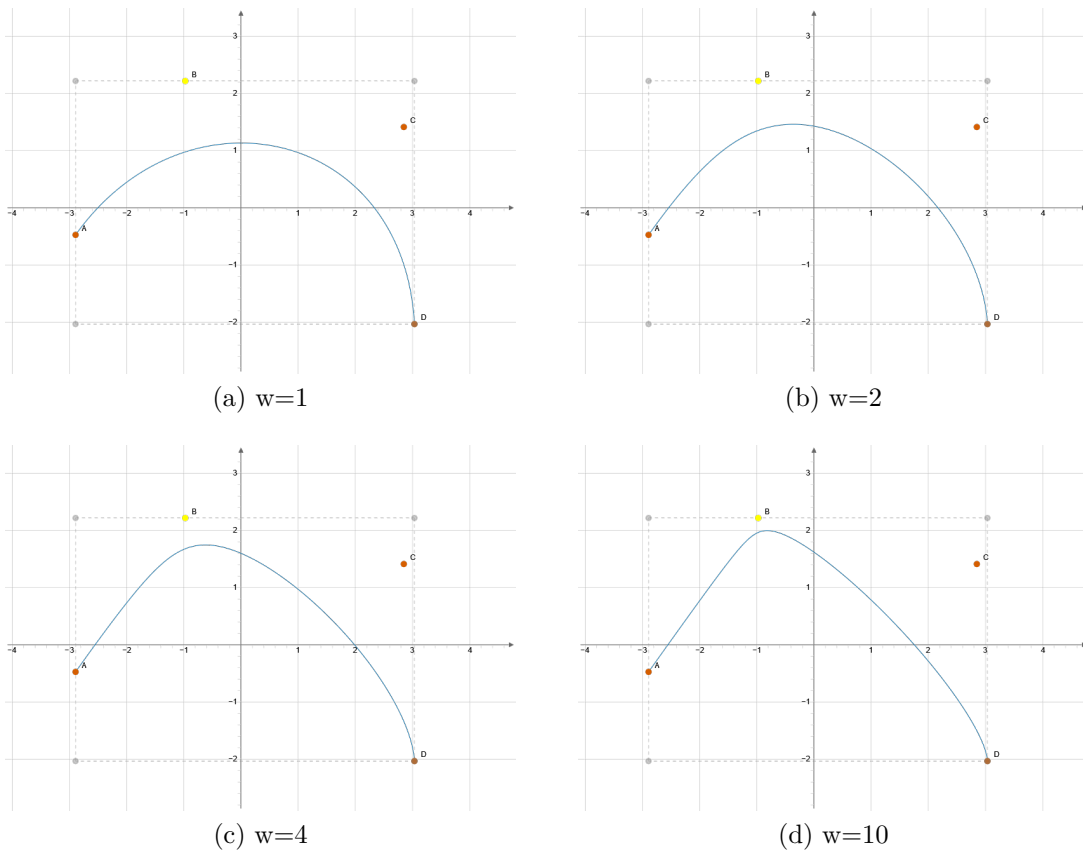
Definicija 2.6. Parametrizacija $\mathbf{R} : [0, 1] \rightarrow \mathbb{R}^d$ Bezierjeve krivulje stopnje n je tako določena s predpisom

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

.

Točkam \mathbf{p}_i tudi tukaj pravimo kontrolne točke, realnim številom w_i pa pravimo *uteži*. Uteži so prosti parametri, ki jih lahko uporabimo pri oblikovanju. Da bi se izognili težavam pri deljenju z 0 ponavadi privzamemo, da so vse uteži pozitivne. Vpliv uteži si pogledjmo na sliki 9. 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. Lastnosti Bezierjevih krivulj, ki smo jih podali v izreku 2.4 veljajo tudi za racionalne Bezierjeve krivulje s pozitivnimi utežmi. Interpolacijo točk lahko dokažemo na podoben način, kakor smo to storili pri dokazu izreka 2.4. Da dokažemo, da je racionalna Bezierjeva krivulja afino invariantna, ter da leži znotraj konveksne ovojnice svojih kontrolnih točk, pa posežemo po naslednjem zapisu.

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



Slika 9: Vpliv uteži

Če pri dokazu izreka 2.4 namesto Bernsteinovih polinomov $B_i^n(t)$ vstavimo funkcijo N_i^n iz zgornjega zapisa, dobimo dokaz lastnosti za racionalne Bezierjeve krivulje.

2.6.1 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 $\in \mathbb{R}^d$.

2.6.2 Decasteljaujev algoritem za racionalne Bezierjeve krivulje

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

2.6.3 Izsek krožnice

V uvodu podrazdelka smo povedali, da se nekaterih krivulj ne da opisati z navadnimi Bezierjevimi krivuljami. Kot enega izmed primerov smo podali izseke krožnice. V tem podpodrazdelku bomo izpeljali takšne kontrolne točke in uteži racionalne Bezierjeve krivulje, da bo krivulja predstavljala izsek krožnice s polmerom r in

Algoritem 2 Racionalni Decasteljaujev algoritem

```

p ← p0, p1, ..., pn
w ← w0, w1, ..., wn
for i = 0, 1, ..., n do
  pi0(t) = pi
  wi0(t) = wi
end for
for r = 1, 2, ..., n do
  for i = 0, 1, ..., n - r do
    wir(t) = (1 - t)wir-1(t) + twi+1r-1(t)
    pir(t) = (1 - t) $\frac{\mathbf{w}_i^{r-1}(t)}{\mathbf{w}_i^r(t)} \mathbf{p}_i^{r-1}(t) + t \frac{\mathbf{w}_{i+1}^{r-1}(t)}{\mathbf{w}_{i+1}^r(t)} \mathbf{p}_{i+1}^{r-1}(t)$ 
  end for
end for
return p0n(t)
  
```

središčem $\mathbf{0} = (0, 0)$. Izsek krožnice bomo predstavljali s kvadratičnimi racionalnimi Bezierjevimi krivuljami. **Ker smo izbrali kvadratične krivulje, te definirajo parabolo.** Za konstrukcijo imamo na voljo 3 kontrolne točke $\mathbf{p}_0, \mathbf{p}_1$ in \mathbf{p}_2 . Ker so racionalne Bezierjeve krivulje afino invariantne, se lahko osredotočimo na primer, kjer točka \mathbf{p}_1 leži na abscisi. Bezierjeve krivulje interpolirajo končne točke, zato morata točki \mathbf{p}_0 in \mathbf{p}_2 ležati na krožnici. Po prejšnji predpostavki mora veljati $\mathbf{p}_0 = (r, 0)$ in $\mathbf{p}_2 = (r \cos(\alpha), r \sin(\alpha))$, kjer je α kot, ki ga točka \mathbf{p}_1 kot vektor tvori z absciso. Poskusimo sedaj upoštevati simetrijo izseka čez premico z naklonom $\frac{\alpha}{2}$. Želimo enaki uteži točk \mathbf{p}_0 in \mathbf{p}_2 , iz izreka REFNAIZREKDAPRVAFIKSNA1 sledi, da lahko nastavimo $w_1 = w_3 = 1$. Simetrijo upoštevajmo tudi pri iskanju pozicije točke \mathbf{p}_2 . Točko postavimo na premico med točkama \mathbf{p}_0 in \mathbf{p}_2 t.j. $\mathbf{p}_1 = \left(R \cos\left(\frac{\alpha}{2}\right), R \sin\left(\frac{\alpha}{2}\right)\right)$. Iščemo sedaj le še R in w_1 .

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

Po komponentah

$$x(t) = \frac{r(1-t)^2 + w_1 R \cos\left(\frac{\alpha}{2}\right) 2t(1-t) + r \cos(\alpha) t^2}{(1-t)^2 + w_1 2t(1-t) + t^2}$$

$$y(t) = \frac{w_1 2t(1-t) R \sin\left(\frac{\alpha}{2}\right) + r \sin(\alpha) t^2}{(1-t)^2 + w_1 2t(1-t) + t^2}$$

Zaradi simetrije mora krivulja v $t = 0.5$ interpolirati točko $\left(r \cos\left(\frac{\alpha}{2}\right), r \sin\left(\frac{\alpha}{2}\right)\right)$.

$$x\left(\frac{1}{2}\right) = \frac{r \frac{1}{4} + w_1 R \cos\left(\frac{\alpha}{2}\right) \frac{1}{2} + r \cos(\alpha) \frac{1}{4}}{\frac{1}{4} + w_1 \frac{1}{2} + \frac{1}{4}} = \frac{\frac{r}{2} + w_1 R \cos\left(\frac{\alpha}{2}\right) + \frac{r}{2} \cos(\alpha)}{1 + w_1} = r \cos\left(\frac{\alpha}{2}\right)$$

$$y\left(\frac{1}{2}\right) = \frac{\frac{w_1}{2} R \sin\left(\frac{\alpha}{2}\right) + \frac{r}{4} \sin(\alpha)}{\frac{1}{4} + w_1 \frac{1}{2} + \frac{1}{4}} = \frac{w_1 R \sin\left(\frac{\alpha}{2}\right) + \frac{r}{2} \sin(\alpha)}{1 + w_1} = r \sin\left(\frac{\alpha}{2}\right)$$

NEXT

$$\begin{aligned}\frac{r}{2} + w_1 R \cos\left(\frac{\alpha}{2}\right) + \frac{r}{2} \cos(\alpha) &= (1 + w_1) r \cos\left(\frac{\alpha}{2}\right) \\ w_1 R \cos\left(\frac{\alpha}{2}\right) - w_1 r \cos\left(\frac{\alpha}{2}\right) &= r \cos\left(\frac{\alpha}{2}\right) - \frac{r}{2} - \frac{r}{2} \cos(\alpha) \\ w_1(R - r) \cos\left(\frac{\alpha}{2}\right) &= r \left(\cos\left(\frac{\alpha}{2}\right) - \frac{1}{2} - \frac{1}{2} \cos(\alpha) \right)\end{aligned}$$

NEXT

$$\begin{aligned}w_1 R \sin\left(\frac{\alpha}{2}\right) + \frac{r}{2} \sin(\alpha) &= (1 + w_1) r \sin\left(\frac{\alpha}{2}\right) \\ w_1(R - r) \sin\left(\frac{\alpha}{2}\right) &= r \sin\left(\frac{\alpha}{2}\right) - \frac{r}{2} \sin(\alpha)\end{aligned}$$

$$\begin{aligned}\frac{r}{2} + w_1 R \cos\left(\frac{\alpha}{2}\right) + \frac{r}{2} \cos(\alpha) &= (1 + w_1) r \cos\left(\frac{\alpha}{2}\right) \\ w_1 R \cos\left(\frac{\alpha}{2}\right) - w_1 r \cos\left(\frac{\alpha}{2}\right) &= r \cos\left(\frac{\alpha}{2}\right) - \frac{r}{2} \cos(\alpha) - \frac{r}{2} \\ w_1(R - r) \cos\left(\frac{\alpha}{2}\right) &= r \cos\left(\frac{\alpha}{2}\right) - \frac{r}{2} \cos(\alpha) - \frac{r}{2} \\ w_1 &= \frac{r \cos\left(\frac{\alpha}{2}\right) - \frac{r}{2} \cos(\alpha) - \frac{r}{2}}{(R - r) \cos\left(\frac{\alpha}{2}\right)} \\ w_1 &= \frac{r \cos\left(\frac{\alpha}{2}\right) - \frac{r}{2} \cos(\alpha) - \frac{r}{2}}{\left(R = \frac{r}{\cos\left(\frac{\alpha}{2}\right)} - r\right) \cos\left(\frac{\alpha}{2}\right)}\end{aligned}$$

Tangenta na krožnico je v točki $(r, 0)$ enaka $(0, r)$. Odvod krivulje v točki $(r, 0)$ je enak $2(\mathbf{p}_1 - \mathbf{p}_0) = 2\left(R \sin\left(\frac{\alpha}{2}\right) - r, R \cos\left(\frac{\alpha}{2}\right)\right)$. Iz česar sledi $R = \frac{r}{\cos\left(\frac{\alpha}{2}\right)}$.

$$\begin{aligned}w_1 \left(\frac{r}{\sin\left(\frac{\alpha}{2}\right)} - r \right) \sin\left(\frac{\alpha}{2}\right) &= r \sin\left(\frac{\alpha}{2}\right) - \frac{r}{2} \sin(\alpha) \\ w_1 &= \frac{r \sin\left(\frac{\alpha}{2}\right) - \frac{r}{2} \sin(\alpha)}{\left(\frac{r}{\sin\left(\frac{\alpha}{2}\right)} - r \right) \sin\left(\frac{\alpha}{2}\right)}\end{aligned}$$

Izračunajmo sedaj za $r = 1$ in $\alpha = \frac{\pi}{2}$. $R = \frac{1}{\cos\left(\frac{\pi}{4}\right)} = \sqrt{2}$.

$$w_1 = \frac{\frac{\sqrt{2}-1}{2}}{(\sqrt{2}-1)\frac{\sqrt{2}}{2}} = \frac{\sqrt{2}-1}{(\sqrt{2}-1)\sqrt{2}} = \frac{\sqrt{2}}{2}$$

3 Zlepki Bezierjevih krivulj

Če si ponovno ogledamo Decasteljaujev algoritem 2.4, 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 $\mathbf{S}|_{[u_{l-1}, u_l]} \in \mathbb{P}_n^d$.

Želimo si tudi, da bi naši zlepi 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 10. Vidimo lahko, da takšen zlepek v CAGD sistemih ne bo najbolj uporaben, saj ima v stični točki t.i. 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 $\mathbf{p}_n^{(1)} = \mathbf{p}_0^{(2)} = \frac{\mathbf{p}_1^{(2)} + \mathbf{p}_{n-1}^{(1)}}{2}$. Na sliki 11 si lahko ogledamo takšen zlepek. 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

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

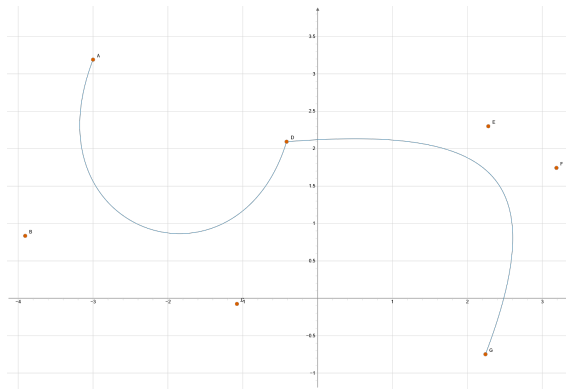
n -krat zvezno odvedljiv.

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

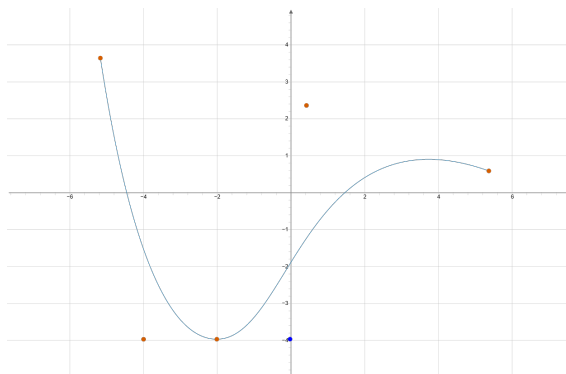
1. G^0 , natanko tedaj, ko je zvezen v analitičnem smislu.

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 β_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 12.



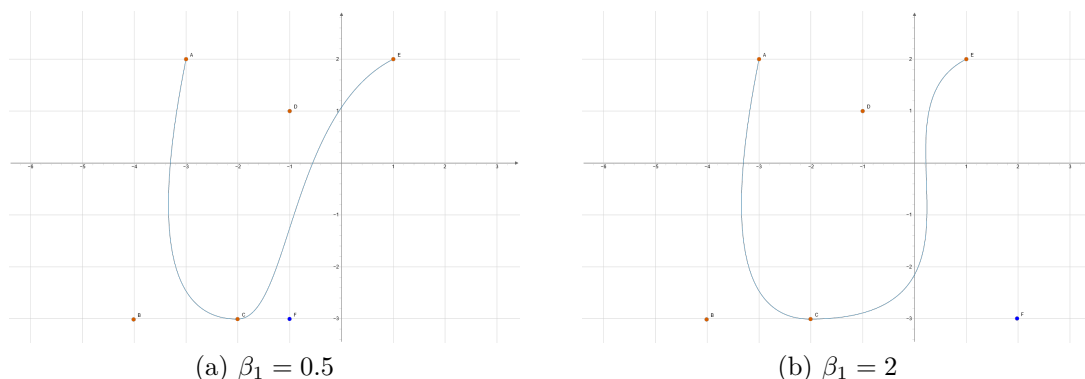
Slika 10: C^0 zlepek dveh Bezierjevih krivulj



Slika 11: C^1 zlepek dveh Bezierjevih krivulj

3.1 Alfa parametrizacije

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...).



Slika 12: Vpliv izbora β_1 pri G^1 zlepku

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 preme 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, t.j. 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' asociamo 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 λ označimo stopnjo polinoma w , s številom μ 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 λ 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 8 iz izreka 2.2. Če

upoštevamo še, da so Bernsteinovi polinomi razčlenitev enote dobimo

$$\begin{aligned}
x(t) &= x_0(B_0^3(t) + B_1^3(t) + B_2^3(t) + B_3^3(t)) \\
&\quad + \frac{1}{3}(u_0^2 - v_0^2)(B_1^3(t) + B_2^3(t) + B_3^3(t)) \\
&\quad + \frac{1}{3}(u_0u_1 - v_0v_1)(B_2^3(t) + B_3^3(t)) \\
&\quad + \frac{1}{3}(u_1^2 - v_1^2)B_3^3(t) \\
y(t) &= y_0(B_0^3(t) + B_1^3(t) + B_2^3(t) + B_3^3(t)) \\
&\quad + \frac{1}{3}2u_0v_0(B_1^3(t) + B_2^3(t) + B_3^3(t)) \\
&\quad + \frac{1}{3}(u_0v_1 + u_1v_0)(B_2^3(t) + B_3^3(t)) \\
&\quad + \frac{1}{3}2u_1v_1B_3^3(t),
\end{aligned}$$

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*

$$\Delta = (u_2v_0 - u_0v_2)^2 - 4(u_0v_1 - u_1v_0)(u_1v_1 - u_2v_1).$$

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 σ 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 2.2, lahko koeficiente σ_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 9. 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.

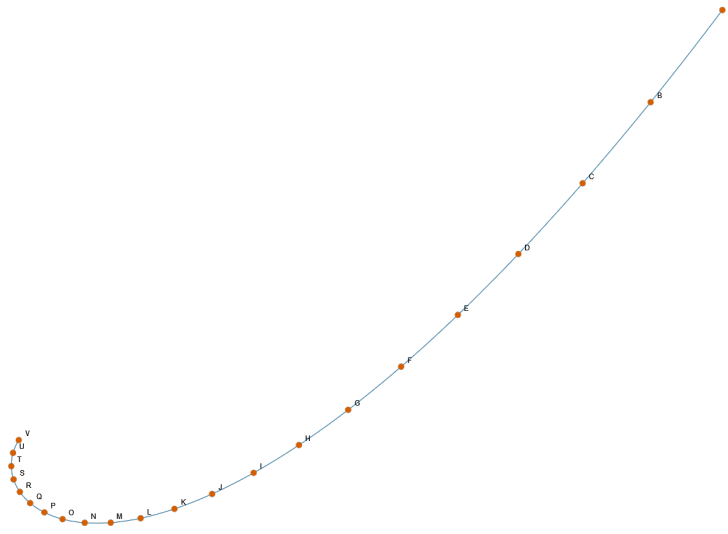
4.5 Enakomerna parametrizacija

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

Krivuljo želimo s parametri $\{t_i\}_{i=0}^N$ razdeliti na N kosov. Če to naredimo tako, da parametre t_i postavimo ekvidistantno, t.j. $\Delta t = t_i - t_{i-1}$ 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 13. Čeprav parametrična hitrost PH krivulje niti ni konstantna, lahko 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 14.



Slika 13: 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...) σ 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*

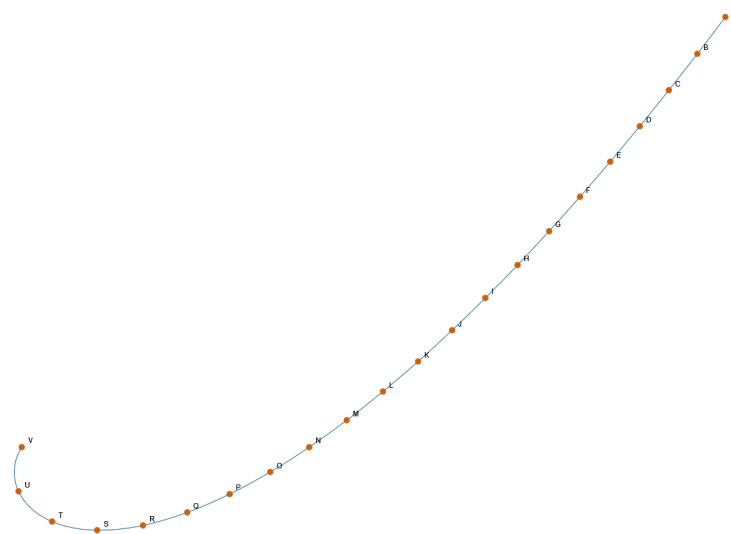
$$\mathbf{t} = \frac{(u^2 - v^2, 2uv)}{\sigma}, \quad \mathbf{n} = \frac{(2uv, v^2 - u^2)}{\sigma}, \quad \kappa = 2 \frac{uv' - u'v}{\sigma^2}.$$

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 κ . 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 $y'' = 2(u'v + uv')$.

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

□



Slika 14: 20 točk na krivulji pri ekvidistantnih t_i

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??)

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

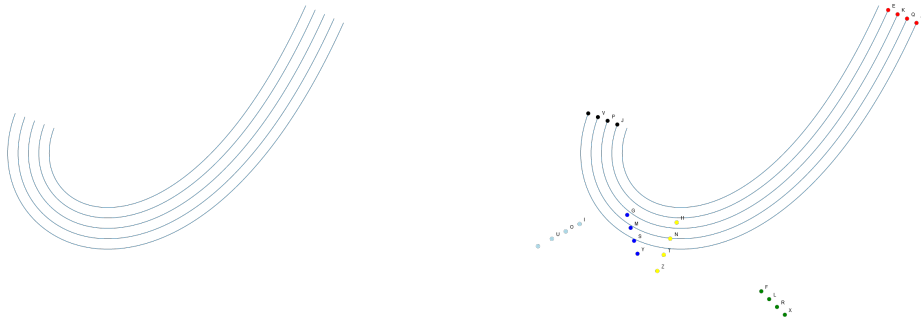
pa definirajo kontrolne točke odmične racionalne Bezierjeve krivulje v homogenih koordinatah. Homogene koordinate odmične krivulje \mathbf{O}_k lahko v zaključeni obliki izrazimo s kontrolnimi točkami prvotne krivulje

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

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.**

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

Tukaj lahko razpišem \mathbf{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 15 lahko vidimo, da se kontrolne točke odmične krivulje



Slika 15: 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 graf sem uporabil odprtokodno knjižnico JsxGraph. Za oblikovanje bootstrap. Za ogrodje pa React.

5.1 Implementacija konceptov magistrskega dela

6 Integrali po ω -kompleksih

6.1 Definicija

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

Črni blok zgoraj je tam namenoma. Označuje, da L^AT_EX 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 ω -kompleks.*

Dokaz. Naštejmo nekaj primerov:

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

¹To ime je izmišljeno.

