

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

Računalništvo in matematika – 2. stopnja

Kevin Štampar

**ORODJE ZA UVOD V BEZIERJEVE
KRIVULJE**

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	2
2.3	Bezierjeve krivulje	2
2.4	Decasteljau	3
2.5	Metode Bezierjevih krivulj	4
2.5.1	Subdivizija	4
2.5.2	Ekstrapolacija	4
2.5.3	Dvig stopnje	4
2.6	Racionalne Bezierjeve krivulje	6
2.6.1	Metode racionalnih Bezierjevih krivulj	8
2.6.2	Decasteljaujev algoritem za racionalne Bezierjeve krivulje	8
3	Zlepki Bezierjevih krivulj	8
3.1	C0	9
3.2	C1	9
3.3	C2	9
3.4	G1	9
3.5	Alfa parametrizacije	9
4	PH Krivulje	9
4.1	Dolžina krivulje kot racionalna funkcija	9
4.2	Polinomska parametrična hitrost	10
4.3	Kontrolne točke Bezierjevih PH krivulj	11
4.4	Racionalni odmik krivulje	12
4.5	Enakomerna parametrizacija	12
5	Orodje za uvod v Bezierjeve krivulje - Bezeg	12
5.1	Implementacija konceptov magistrskega dela	12
6	Integrali po ω-kompleksih	13
6.1	Definicija	13
7	Tehnični napotki za pisanje	13
7.1	Sklicevanje in citiranje	13
7.2	Okrajšave	13
7.3	Vstavljanje slik	13
7.4	Kako narediti stvarno kazalo	14
7.5	Navajanje literature	14
	Literatura	15

Program dela

Mentor naj napiše program dela skupaj z osnovno literaturo.

Podpis mentorja:

Orodje za uvod v Bezierjeve krivulje

POVZETEK

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

Tool for introduction into Bezier 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 osnove Bezierjevih krivulj. Začeli bomo z Bernsteinovimi polinomi, ki jih bomo uporabili pri definiciji Bezierjevih krivulj. Predstavili bomo Decasteljaujev algoritem, ki je ključen za stabilen način računanja 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 so ključne za delovanje Bezierjevih krivulj. i -ti *Bernsteinov bazni polinom* stopnje n definiramo kot $b_{i,n}(t) := \binom{n}{i} t^i (1-t)^{n-i}$. Linearni kombinaciji takšnih polinomov t.j. $B_n(t) := \sum_{i=0}^n \beta_i b_{i,n}(t)$, pravimo *Bernsteinov polinom* stopnje n . V izreku 2.1 naštejemo nekaj lastnosti Bernsteinovih polinomov.

Izrek 2.1. *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. $b_{n,i}(t) = (1-t)b_{n-1,i}(t) + tb_{n-1,i-1}(t)$

7. $b'_{i,n}(t) = n(b_{i-1,n-1}(t) - b_{i,n-1}(t))$ in $B'_n(t) = n \sum_{i=0}^{n-1} (\beta_{i+1} - \beta_i) b_{i,n-1}(t)$

Dokaz. Točki (1) in (2) očitno izhajata iz lastnosti binomskega simbola. Dokažimo ostale.

(3) 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

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

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

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

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

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

(6) Dodaj dokaz! Lahko tudi večdimenzionalno ane □

2.2 Večdimenzionalne oznake

Z željo po krajših, bolj preglednih zapisih, bomo uvedli večdimenzionalne oznake. Večdimenzionalnost bomo ponazarjali z odebelitvijo črke. Tako bomo večdimenzionalne točke označili z $\mathbf{x} = (x_0, x_1, \dots, x_n)$, večdimenzionalne 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 stopnje n namesto realnega števila β_i vstavimo točke $\mathbf{p}_i \in \mathbb{R}^d$, dobimo predpis za parametrizacijo *Bezierjeve krivulje* stopnje n v \mathbb{R}^d t.j. $\mathbf{B}_n(t) = \sum_{i=0}^n \mathbf{p}_i b_{i,n}(t)$. 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 *kontrolni poligon*. Opomba: pri slikovnem prikazovanju raznih konceptov Bezierjevih krivulj, se bomo omejili na $d = 2$, torej na Bezierjeve krivulje v ravnini.

Izrek 2.2. *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 potrebe grafičnega oblikovanja. Interpolacija končnih točk in lastnost (3) sta pomembni, saj omogočata enostavno kontrolo krivulje. Afina invarianca pa je pomembna za enostavno premikanje, rotiranje itd. krivulje, saj lahko krivuljo transformiramo tako, da transformacijo uporabimo na kontrolnih točkah.

Dokaz.

- (1) $\mathbf{B}_n(0) = \sum_{i=0}^n \mathbf{p}_i b_{n,i}(0) = \sum_{i=0}^n \mathbf{p}_i \delta_{0,i} = \mathbf{p}_0$. Enako lahko naredimo tudi za $\mathbf{B}_n(1)$.
(2) Naj bo ϕ afina preslikava, velja torej $\phi(x) = A\mathbf{x} + \mathbf{b}$.

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

- (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 $b_{n,i}(t) \geq 0$, lahko zapišemo $\lambda_i = b_{n,i}(t)$. \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 2.3. *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.

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



□

Algoritem 1 Decasteljau

```

p ← p0, p1, ..., pn
for  $i = 0, 1, \dots, n$  do
    pi0(t) = pi
end for
for  $r = 1, 2, \dots, n$  do
    for  $i = 0, 1, \dots, n - r$  do
        pir(t) = (1 - t)pir-1(t) + tpi+1r-1(t)
    end for
end for
return p0n(t)

```

Decasteljaujev algoritem ima tudi geometrijski pomen. Predstavlja namreč nekakšno zaporedno interpolacijo točk.

2.5 Metode Bezierjevih krivulj

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

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 algoritma1. 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.

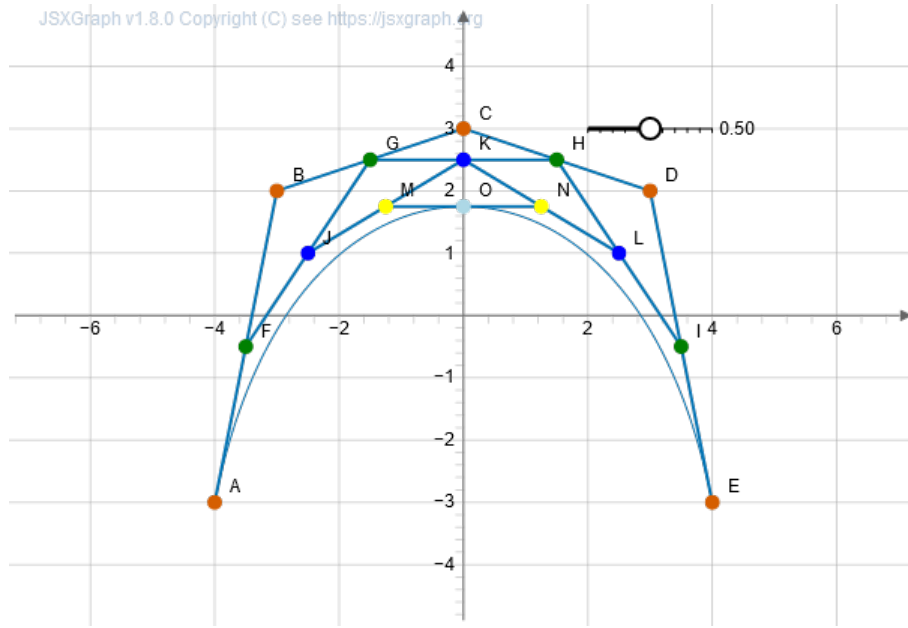
$$\begin{aligned}
 B_{t_0}(t) &= B(t_0 t) = \sum_{i=0}^n \mathbf{p}_i \binom{n}{i} (t_0 t)^i (1 - t t_0)^{n-i} \\
 &= \sum_{i=0}^n \mathbf{p}_i t_0^i \left(\frac{1 - t t_0}{1 - t} \right)^{n-i} \binom{n}{i} t^i (1 - t)^{n-i}
 \end{aligned}$$

2.5.2 Ekstrapolacija

Motivacija: Zelimo podaljšati krivuljo

2.5.3 Dvig 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



Slika 1: Subdivizija

krivuljo $B(t)$ zapišemo tudi kot $B(t) = (1-t)B(t) + tB(t) = \sum_{i=0}^n \mathbf{p}_i(1-t)b_{i,n}(t) + \sum_{i=0}^n \mathbf{p}_i t b_{i,n}(t)$. Razpišimo najprej prvi del.

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

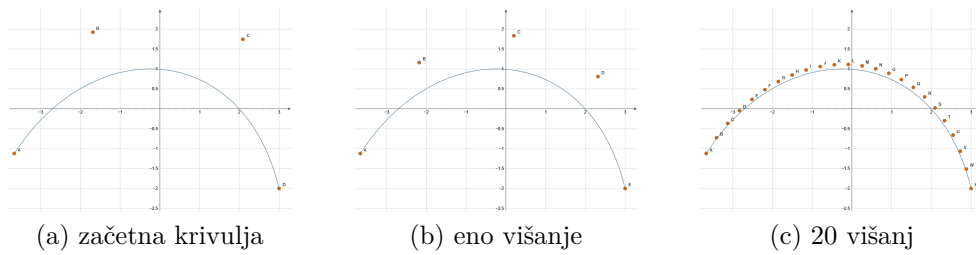
Sedaj pa še za drugega.

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

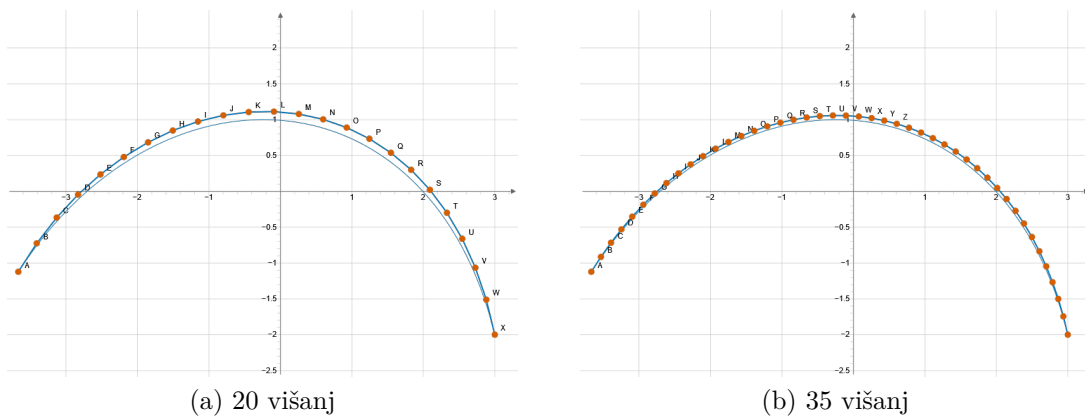
Vstavimo v $B(t)$.

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

Oglejmo si kako dvig stopnje izgleda na neki krivulji. Na sliki2 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 sliki3.



Slika 2: Višanje Bezierjeve krivulje

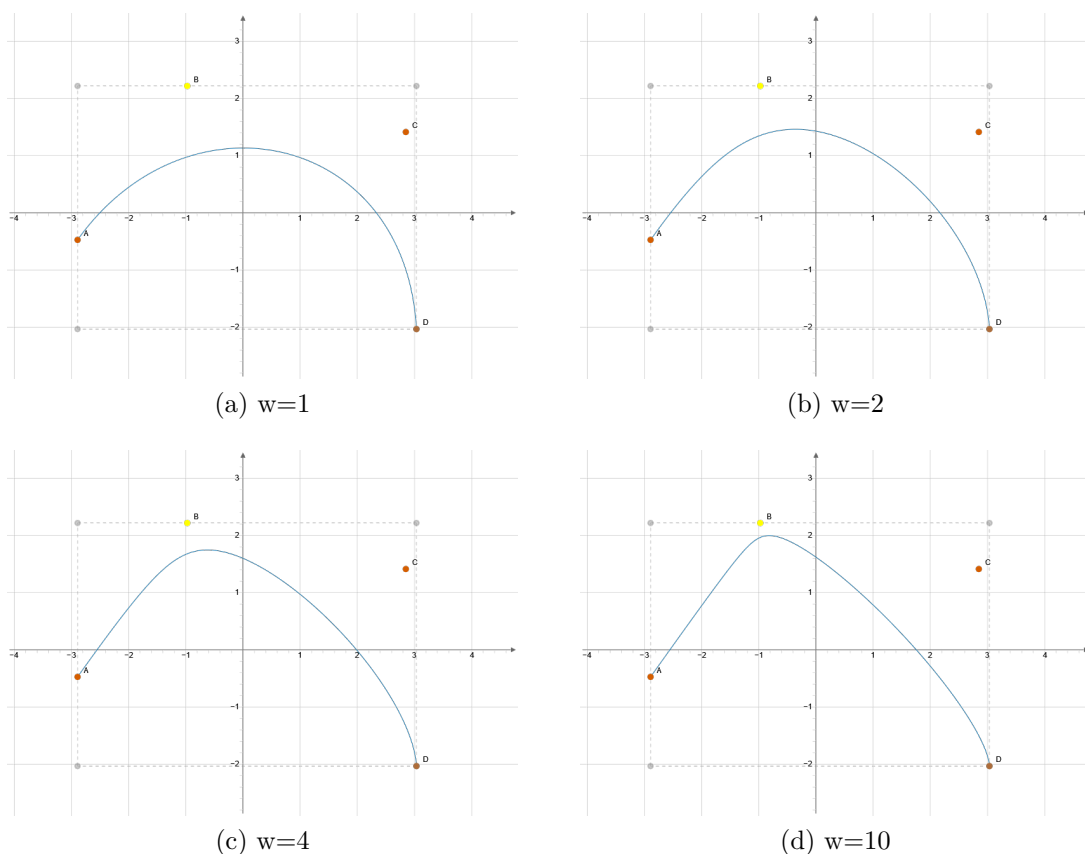


Slika 3: Prileganje kontrolnega poligona

2.6 Racionalne Bezierjeve krivulje

Vseh krivulj se ne da opisati z navadnimi Bezierjevimi krivuljami. Med njimi so tudi takšne, ki so za računalniško grafiko 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 krivuljo stopnje n v \mathbb{R}^{d+1} projiciramo na hiperravnino $w = 1$. Projekcijo definiramo s predpisom $(w, \mathbf{x}) \rightarrow (1, \frac{\mathbf{x}}{w})$. 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 sliki4. 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.



Slika 4: Vpliv uteži

Lastnosti Bezierjevih krivulj, ki smo jih podali v izreku2.2 veljajo tudi za racionalne Bezierjeve krivulje s pozitivnimi utežmi. Interpolacijo točk lahko dokažemo na podoben način, kakor smo to storili pri dokazu izreka2.2. 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 2.2 namesto Bernsteinovih polinomov $b_{n,i}$ 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.

Algoritem 2 Racionalni Decasteljaujev algoritem

```

P  $\leftarrow$  p0, p1, ..., pn
w  $\leftarrow$  w0, w1, ..., wn
for  $i = 0, 1, \dots, n$  do
    pi0(t) = pi
    wi0(t) = wi
end for
for  $r = 1, 2, \dots, n$  do
    for  $i = 0, 1, \dots, n - r$  do
        wir(t) = (1 - t)wir-1(t) + twi+1r-1(t)
        pir(t) = (1 - t) $\frac{w_i^{r-1}(t)}{w_i^r(t)}$ pir-1(t) + t $\frac{w_{i+1}^{r-1}(t)}{w_{i+1}^r(t)}$ pi+1r-1(t)
    end for
end for
return p0n(t)

```

3 Zlepki Bezierjevih krivulj

Računanje Bezierjevih točk krivulj je pri visokih številih n lahko zamudno. Če si ponovno ogledamo Decasteljaujev algoritem 2.4, lahko hitro opazimo, da je časovna kompleksnost algoritma $O(n^2)$. Bezierjeve krivulje z veliko kontrolnimi točkami so zato v grafičnem oblikovanju neuporabne. Da bi ohranili čim večjo kontrolo krivulj s čim manj računanja, posežemo po zlepkih Bezierjevih krivulj.

Definicija 3.1. Zlepek $s : [a, b] \rightarrow \mathbb{R}$ stopnje n nad zaporedjem stičnih točk

$$a = u_0 < u_1 < \dots < u_{m-1} < u_m = b$$

je odsekoma polinomska funkcija, za katero velja $s|_{[u_{l-1}, u_l]} \in \mathbb{P}_n$.

Definicija 3.2. Zlepek krivulj stopnje n je krivulja s parametrizacijo $\mathbf{S} : [a, b] \rightarrow \mathbb{R}^d$ nad zaporedjem stičnih točk

$$a = u_0 < u_1 < \dots < u_{m-1} < u_m = b$$

je odsekoma polinomska krivulja, za katere velja $S_i|_{[u_{i-1}, u_i]} \in \mathbb{P}_n$, kjer so polinomi S_i komponente....*****

Želimo si tudi, da bi bili naši zlepki gladki. Krivulja je na posameznih odsekih polinomska in zato tudi gladka, problem je le v stičnih točkah. V naslednjih podrazdelkih bomo predstavili nekaj osnovnih zlepkov Bezierjevih krivulj različnih stopenj gladkosti????.

$$S_{[u_{i-1}, u_i]} \in \mathbb{P}_n$$

,

3.1 C0

3.2 C1

3.3 C2

3.4 G1

3.5 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?

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 $v(t)$. Idealno bi bilo, da je hitrost $v(t) = 1$ za vsak $t \in [0, t_0]$, saj bi v tem primeru veljalo $s(t) = t$. Takšna parametrizacija je zlahka dosegljiva za ravne črte(daljice?***), 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 $x'(t) + y'^2(t)$ popoln kvadrat. V takšnem primeru se namreč koren v integrandu pokrajša, pod integralom pa ostane polinom. Krivuljam za katere to drži pravimo *krivulje s pitagorejskimi hodografi* oziroma *PH krivulje*. To so krivulje, ki jim lahko dolžino izračunamo tako, da evalviramo polinom.

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?)

Za pomoč pri konstrukciji bezierjevih PH krivulj bomo potrebovali izrek, ki nam bo pomagal pri generaciji polinomov, ki tvorijo pitagorejske trojice.

Izrek 4.1. *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 relativno praštevilska 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 polinomi $\tilde{a}(t) = \frac{a(t)}{w(t)}$, $\tilde{b}(t) = \frac{b(t)}{w(t)}$, $\tilde{c}(t) = \frac{c(t)}{w(t)}$ relativno praštevilski 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 relativno praštevilska 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). □

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 relativno praštevilska polinoma u, v in polinom w v izraza

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

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 ravno črto.
3. Če sta polinoma u in v konstantna ter vsaj en neničelen, polinom w pa ni konstanten. Potem hodograf spet definira ravno 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 krivule 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, za katere velja $\mathbf{r}(t) \neq 0$ pri vsakem realnem številu t . Tako dobljene PH krivulje so lihe stopnje $n = 2\mu + 1$.

4.4 Racionalni odmik krivulje

4.5 Enakomerna parametrizacija

5 Orodje za uvod v Bezierjeve krivulje - Bezeg

Vsi koncepti predstavljeni v magistrskem delu so tudi implementirani na spletni strani. Za graf sem uporabil `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:

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

□

7 Tehnični napotki za pisanje

7.1 Sklicevanje in citiranje

Za sklice uporabljamo `\ref`, za sklice na enačbe `\eqref`, za citate `\cite`. Pri sklicevanju in citiranju sklicano številko povežemo s prejšnjo besedo z nedeljivim presledkom `~`, kot npr. iz `trditve~\ref{trd:obstoj-omega}` vidimo.

Primer 7.1. Zaporedje (6.1) iz dokaza trditve 6.2 na strani 13 lahko najdemo tudi v Spletni enciklopediji zaporedij [oeis]. Citiramo lahko tudi bolj natančno [lebedev2009introduction] ◇

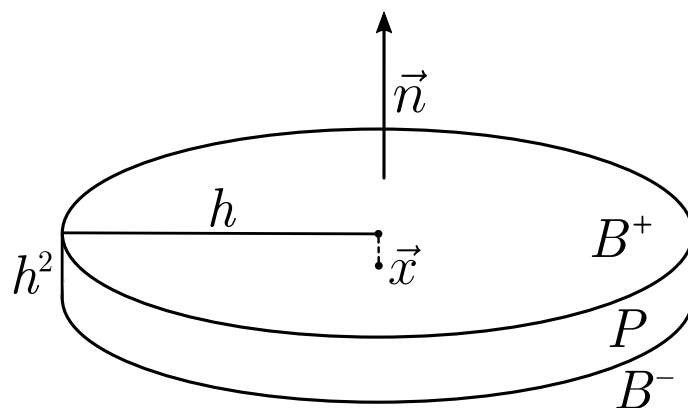
7.2 Okrajšave

Pri uporabi okrajšav L^AT_EX za piko vstavi predolg presledek, kot npr. tukaj. Zato se za vsako piko, ki ni konec stavka doda presledek običajne širine z ukazom `_`, kot npr. tukaj. Primerjaj z okrajšavo zgoraj za razliko.

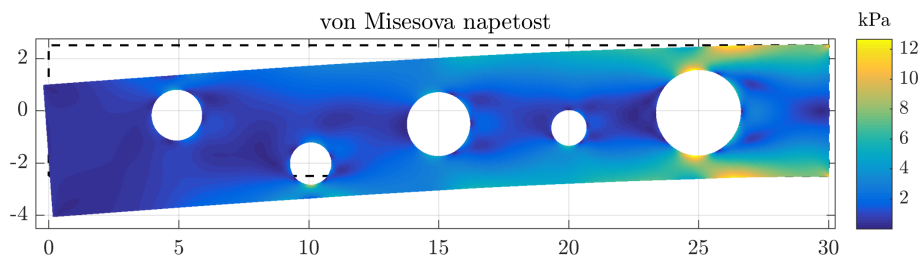
7.3 Vstavljanje slik

Sliko vstavimo v plavajočem okolju `figure`. Plavajoča okolja *plavajo* po tekstu, in jih lahko postavimo na vrh strani z opsijskim parametrom `'t'`, na lokacijo, kjer je v kodi s `'h'`, in če to ne deluje, potem pa lahko rečete L^AT_EXu, da ga *res* želite tukaj, kjer ste napisali, s `'h!'`. Lepo je da so vstavljene slike vektorske (recimo `.pdf` ali `.eps` ali `.svg`) ali pa `.png` visoke resolucije (več kot 300 dpi). Pod vsako sliko je napis in na vsako sliko se skličemo v besedilu. Primer vektorske slike je na sliki 5. Vektorsko sliko prepoznate tako, da močno zoomate v sliko, in še vedno ostane gladka. Več informacij je na voljo na https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions. Če so slike bitne, kot na primer slika 6, poskrbite, da so v dovolj visoki resoluciji.

¹To ime je izmišljeno.



Slika 5: Primer vektorske slike z oznakami v enaki pisavi, kot jo uporablja \LaTeX . Narejena je s programom Inkscape, \LaTeX oznake so importane v Inkscape iz pomožnega PDF.



Slika 6: Primer bitne slike, izvožene iz Matlab. Poskrbite, da so slike v dovolj visoki resoluciji in da ne vsebujejo prosojnih elementov (to zahteva PDF/A-1b format).

7.4 Kako narediti stvarno kazalo

Dodate ukaze `\index{polje}` na besede, kjer je pojavijo, kot tukaj . Več o stvarnih kazalih je na voljo na <https://en.wikibooks.org/wiki/LaTeX/Indexing>.

7.5 Navajanje literature

Članke citiramo z uporabo `\cite{label}`, `\cite[text]{label}` ali pa več naenkrat s `\cite{\label1, label2}`. Tudi tukaj predhodno besedo in citat povežemo z nedeljivim presledkom `~`. Na primer `[chen2006meshless, liu2001point]`, ali pa `[kibriya2007empirical],[kibriya2007empirical]`, ali pa `[trobec2015parallel]`, `[pereira2016con]`. Vnosi iz `.bib` datoteke, ki niso citirani, se ne prikažejo v seznamu literature, zato jih tukaj citiram. `[vene2000categorical]`, `[gregoric2017stopniceni]`, `[slak2015induktivni]`, `[nsphere]`, `[kearsley1975linearly]`, `[STtemplate]`, `[NunbergerTand]`, `[vanoosten2008realizab`

Stvarno kazalo

tukaj, 14