

UNIVERZA V LJUBLJANI  
FAKULTETA ZA MATEMATIKO IN FIZIKO

Matematika – 1. stopnja

Urška Bobnar

**Računanje presečišč ravninskih Bézierovih krivulj**

Delo diplomskega seminarja

Mentorica: doc. dr. Marjetka Krajnc

Ljubljana, 2014

## KAZALO

1. Uvod	4
2. Splošno o Bézierovih krivuljah	5
2.1. De Casteljaouov algoritem	5
2.2. Lastnosti Bernsteinovih polinomov in Bézierovih krivulj	8
2.3. Razcvetna oblika Bézierove krivulje	11
2.4. Subdivizija	12
2.5. Spreminjanje stopnje krivulje	13
2.6. Racionalna Bézierova krivulja	15
2.7. Eksplicitna oblika	15
3. Računanje presečišč ravninskih Bézierovih krivulj	15
3.1. Bézierov subdivizijski algoritem	16
3.2. Implicitizacija	17
3.3. Bézierovo izrezovanje	25
3.4. Časovna zahtevnost	29
Literatura	29

# Računanje presečišč ravninskih Bézierovih krivulj

## POVZETEK

Bézierove krivulje predstavljajo eno izmed osnovnih orodij, s katerim delamo v računalniško podprtem geometrijskem oblikovanju. S to posebno obliko krivulj lahko aproksimiramo poljubne oblike. Zaradi svojih lastnosti so enostavne za uporabo tudi med oblikovalci z malo matematičnega znanja. Pri delu s krivuljami večkrat naletimo na problem, kako enostavno in hitro izračunati presečišča med različnimi Bézierovimi krivuljami. Iskanje presečišč ravninskih krivulj je raziskano področje, ki najde svojo uporabo tako v numerični matematiki kot tudi v robotiki.

V tem diplomskem delu so najprej predstavljene Bézierove krivulje, njihove lastnosti ter de Casteljauov algoritem, s katerim na geometrijski način definiramo krivuljo. Nato so obravnavani Bézierov subdivizijski algoritem, implicitizacija in Bézierovo izrezovanje, ki so najpogosteje uporabljene metode za izračun presečišč ravninskih Bézierovih krivulj.

## Computing the intersection of planar Bézier curves

### ABSTRACT

Bézier curves are one of the main tools used in computer-aided geometric design. The special curve forms can be used to approximate any shapes. Due to their properties, they are being used also among graphic designers with little mathematical knowledge. However, when working with the curves, one often needs to calculate the intersections between different Bézier curves in a simple and quick manner. Many studies have been conducted on finding intersections between planar curves, since this is a field which finds its use in numerical mathematics and robotics.

The thesis first describes the Bézier curves, their properties and definitions; further, it analyses the De Casteljau's algorithm to define the curve geometrically. Finally, the thesis focuses on the Bézier subdivision algorithm, implicitization and Bézier clipping, which are the most commonly used methods for computing intersections of planar Bézier curves.

**Math. Subj. Class. (2010):** 14Q05, 65D17, 68U07

**Ključne besede:** Bézierova krivulja, de Casteljauov algoritem, subdivizija, implicitizacija, Bézierov subdivizijski algoritem, Bézierovo izrezovanje

**Keywords:** Bézier curve, De Casteljau's algorithm, subdivision, implicitization, Bézier subdivision algorithm, Bézier clipping

## 1. UVOD

Bézierove krivulje tvorijo poseben razred polinomskih parametričnih krivulj. Te krivulje imajo posebno vlogo v računalniški grafiki in računalniško podprtem geometrijskem oblikovanju, saj veliko programov za oblikovanje (npr. Adobe Illustrator, CorelDRAW ...) uporablja Bézierove krivulje in njihove lastnosti pri grafičnem oblikovanju, prav tako je s pomočjo teh krivulj definiranih veliko pisav.

Ime so dobile po dr. Pierru Bézieru, inženirju avtomobilske družbe Renault. Ta je leta 1962 objavil svoje delo, v katerem je opisal posebno obliko krivulje, ki jo je uporabljal za oblikovanje avtomobilov. V istem času je neodvisno in na drug način do teh krivulj prišel tudi Paul de Casteljau, ki je bil matematik v Citroënovi avtomobilski družbi. Leta 1959 je napisal članek, ki ga ni objavil in so ga kasneje odkrili. V tem članku je predstavil numerično stabilen algoritem za izračun točk na Bézierovi krivulji, ki ga danes imenujemo de Casteljauov algoritem.

Oba avtorja sta želela s to krivuljo doseči čim enostavnejši način za oblikovanje avtomobilov, ki bi bil prijazen do oblikovalcev in od njih ne bi zahteval veliko matematičnega znanja. Kot bomo videli v nadaljevanju, so Bézierove krivulje definirane z Bernsteinovimi polinomi, ki so poznani že od leta 1912, v grafiko pa so bili vpeljani 50 let kasneje [7, 8, 9].

V diplomskem delu so na začetku predstavljene Bézierove krivulje, osnovne definicije in lastnosti povezane s to obliko krivulj, geometrijski algoritem, ki nas pripelje do njih ter različne oblike krivulj [1, 6]. V drugem delu so predstavljeni različni algoritmi za računanje presečišč ravninskih Bézierovih krivulj. Natančneje so opisani Bézierov subdivizijski algoritem, Bézierovo izrezovanje in implicitizacija, ki sodijo med algoritme, ki se najpogosteje uporabljajo pri računanju presečišč. V zaključku je predstavljena primerjava časovne zahtevnosti teh algoritmov [3].

Poleg uporabe v računalniški grafiki imajo Bézierove krivulje pomembno vlogo tudi v robotiki. Potovanje robota, gibanje robotske roke, glave itd., opišemo kot potovanje po Bézierovi krivulji. Pojavi se vprašanje ali se robota srečata in kdaj, ali se poti robotskih rok križata in podobno. To je enako problemu iskanja presečišč Bézierovih krivulj.

Za začetek si pogledjmo definicijo splošne Bézierove krivulje. Krivuljo stopnje  $n$  definiramo z  $n + 1$  kontrolnimi točkami na sledeč način.

**Definicija 1.1.** *Ravninska Bézierova krivulja stopnje  $n$  je parametrična krivulja  $\mathbf{b} : [0, 1] \rightarrow \mathbb{R}^2$ , definirana s predpisom*

$$\mathbf{b}(t) = \sum_{i=0}^n B_i^n(t) \mathbf{b}_i,$$

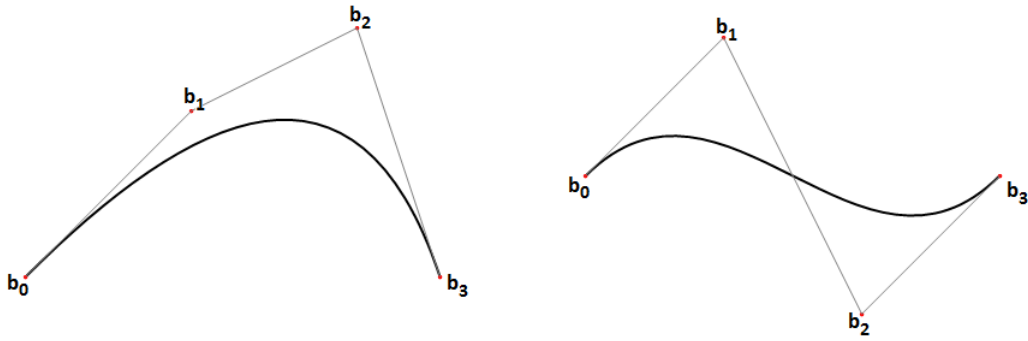
kjer so  $\mathbf{b}_i \in \mathbb{R}^2$  kontrolne točke in

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i, \quad i = 0, 1, \dots, n,$$

Bernsteinovi bazni polinomi.

**Opomba 1.2.** Za Bernsteinove polinome velja tudi, da je  $B_0^0(t) \equiv 1$  in  $B_i^n(t) \equiv 0$  za  $i \notin \{0, \dots, n\}$ .

Na sliki 1 sta prikazana primera dveh kubičnih Bézierovih krivulj.



SLIKA 1. Primera dveh kubičnih Bézierovih krivulj.

## 2. SPLOŠNO O BÉZIEROVIH KRIVULJAH

**2.1. De Casteljaouov algoritem.** De Casteljaouov algoritem, s katerim geometrijsko definiramo Bézierovo krivuljo, sodi med najbolj osnovne algoritme pri delu s krivuljami in ploskvami. Preprosto ponavljanje linearne interpolacije nas pripelje do postopka, ki nam tudi poda povezavo med geometrijsko in algebraično definicijo krivulje.

Najprej si pogledjmo konstrukcijo parabole, ki nas bo pripeljala do Bézierove oblike krivulje. Naj bodo  $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$  poljubne 3 točke v ravnini  $\mathbb{R}^2$  in  $t \in [0, 1]$ . Z linearno interpolacijo konstruiramo točke

$$\mathbf{b}_0^1(t) = (1 - t)\mathbf{b}_0 + t\mathbf{b}_1$$

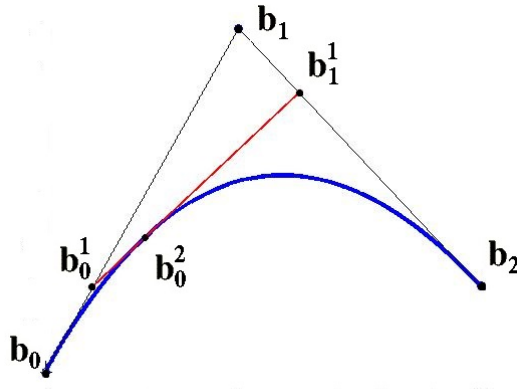
$$\mathbf{b}_1^1(t) = (1 - t)\mathbf{b}_1 + t\mathbf{b}_2$$

$$\mathbf{b}_0^2(t) = (1 - t)\mathbf{b}_0^1 + t\mathbf{b}_1^1,$$

vstavimo prvi dve enačbi v zadnjo in dobimo naslednjo zvezo

$$\mathbf{b}_0^2(t) = (1 - t)^2\mathbf{b}_0 + 2t(1 - t)\mathbf{b}_1 + t^2\mathbf{b}_2,$$

ki predstavlja parametrično obliko parabole. Na sliki 2 je prikazana konstrukcija parabole pri parametru  $t = 0.25$ .



SLIKA 2. Konstrukcija točke na paraboli pri parametru  $t = 1/4$ .

Posplošimo zdaj zgornji algoritem na splošnih  $n + 1$  točk  $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$  v  $\mathbb{R}^2$ , ki nas bo pripeljal do krivulje stopnje  $n$ . Algoritem na  $r$ -tem koraku konstruira  $i$ -to kontrolno točko po formuli

$$\mathbf{b}_i^r(t) = (1 - t)\mathbf{b}_i^{r-1}(t) + t\mathbf{b}_{i+1}^{r-1}(t),$$

za  $r = 1, \dots, n$  in  $i = 0, \dots, n - r$ . Pri tem so točke  $\mathbf{b}_i^0(t) \equiv \mathbf{b}_i$  in na zadnjem koraku dobimo  $\mathbf{b}_0^n(t)$ , ki predstavlja točko na Bézierovi krivulji pri parametru  $t$ . Prepričajmo se, da to res drži.

**Trditev 2.1.** *Točke na  $r$ -tem koraku algoritma se izražajo v Bernsteinovi bazi kot*

$$\mathbf{b}_i^r(t) = \sum_{k=i}^{i+r} \mathbf{b}_k B_{k-i}^r(t),$$

za  $r = 1, \dots, n$  in  $i = 0, \dots, n - r$ .

*Dokaz.* Dokaz bomo naredili z indukcijo na  $r$ . Pri  $r = 0$  dobimo

$$\mathbf{b}_i^0(t) = \sum_{k=i}^i \mathbf{b}_k B_{k-i}^0(t) = \mathbf{b}_i B_0^0(t) = \mathbf{b}_i,$$

kar so ravno kontrolne točke. Predpostavimo sedaj, da formula velja za neki  $r < n$ . Poglejmo  $(r + 1)$ -vo točko, ki se po algoritmu izraža kot

$$\mathbf{b}_i^{r+1}(t) = (1 - t)\mathbf{b}_i^r(t) + t\mathbf{b}_{i+1}^r(t).$$

Da bi lahko dokazali trditev za  $r + 1$ , si pogledjmo najprej zvezo, ki velja za Bernsteinove polinome. In sicer velja

$$B_i^n(t) = (1 - t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t),$$

kar enostavno dokažemo, saj je

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

Z uporabo indukcijske predpostavke in zgornje zveze med Bernsteinovimi polinomi dobimo

$$\begin{aligned}
\mathbf{b}_i^{r+1}(t) &= (1-t) \sum_{k=i}^{i+r} \mathbf{b}_k B_{k-i}^r(t) + t \sum_{k=i+1}^{i+1+r} \mathbf{b}_k B_{k-i-1}^r(t) \\
&= \mathbf{b}_i(1-t)B_0^r(t) + \sum_{k=i+1}^{i+r} \mathbf{b}_k((1-t)B_{k-i}^r(t) + tB_{k-i-1}^r(t)) + t\mathbf{b}_{i+1+r}B_r^r(t) \\
&= \mathbf{b}_i B_0^{r+1}(t) + \sum_{k=i+1}^{i+r} \mathbf{b}_k B_{k-i}^{r+1}(t) + \mathbf{b}_{i+1+r} B_{r+1}^{r+1}(t) \\
&= \sum_{k=i}^{i+1+r} \mathbf{b}_k B_{k-i}^{r+1}(t).
\end{aligned}$$

□

Če v trditvi 2.1 izberemo  $r = n$  in  $i = 0$ , dobimo

$$\mathbf{b}_0^n(t) = \sum_{j=0}^n \mathbf{b}_j B_j^n(t).$$

Vidimo, da lahko točko na zadnjem koraku de Casteljauovega algoritma zapišemo v Bernsteinovi bazi. S tem smo tudi dokazali, da točka  $\mathbf{b}_0^n$  res leži na Bézierovi krivulji. Ker algoritem temelji na ponavljajoči se linearni interpolaciji, je numerično stabilen, saj je  $\mathbf{b}^n(t)$  vedno baricentrična kombinacija točk. Pri konstrukciji krivulje na ravninskih točkah vedno dobimo ravninsko krivuljo.

**Definicija 2.2.** Poligon  $\mathbf{P}$ , sestavljen iz točk  $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ , imenujemo *Bézierov poligon* oz. *kontrolni poligon* krivulje  $\mathbf{b}$ , njegove komponente  $\mathbf{b}_i$  pa imenujemo *kontrolne točke* krivulje  $\mathbf{b}$ . Zapišemo tudi  $\mathbf{b} = \mathcal{B}[\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n] = \mathcal{B}[\mathbf{P}]$ . Tak zapis definira  $\mathcal{B}$  kot linearni operator, ki povezuje Bézierovo krivuljo z njenim kontrolnim poligonom. Rečemo tudi, da je krivulja  $\mathcal{B}[\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n]$  *Bernstein-Bézierova aproksimacija* kontrolnega poligona.

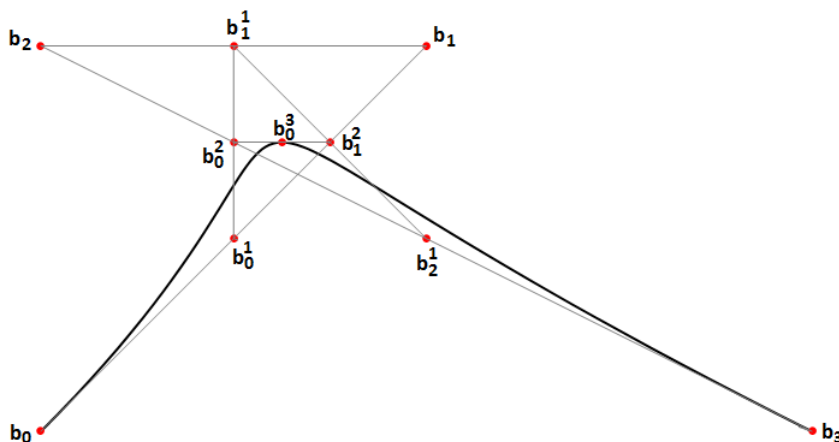
**Opomba 2.3.** Krivulja bo v delu označena kot  $\mathbf{b}$ , če pa bo poudarjena stopnja krivulje, bo ta označena kot  $\mathbf{b}^n$ .

Točke, dobljene na vsakem koraku algoritma, lahko zapišemo v trikotno obliko, ki jo imenujemo *de Casteljauova shema*. V primeru Bézierove krivulje stopnje  $n$  je shema oblike

$$\begin{array}{ccccccc}
& & & & & & \mathbf{b}_0^0 \\
& & & & & & \mathbf{b}_1^0 & \mathbf{b}_0^1 \\
& & & & & & \mathbf{b}_2^0 & \mathbf{b}_1^1 & \cdots \\
& & & & & & \vdots & \vdots & \cdots & \mathbf{b}_0^{n-1} \\
& & & & & & \mathbf{b}_n^0 & \mathbf{b}_{n-1}^1 & \cdots & \mathbf{b}_1^{n-1} & \mathbf{b}_0^n
\end{array}$$

**Primer 2.4.** Poglejmo primer sheme za kubično Bézierovo krivuljo, ki jo vidimo na sliki 3 in jo definirajo kontrolne točke

$$\mathbf{b}_0 = (0, 0), \mathbf{b}_1 = (2, 2), \mathbf{b}_2 = (0, 2), \mathbf{b}_3 = (4, 0).$$



SLIKA 3. Kubična krivulja z označenimi točkami, ki jih dobimo pri de Casteljauovem algoritmu in so zapisane v shemi.

Pri parametru  $t = 0.5$  dobimo

$$\begin{array}{cccc} (0, 0) & & & \\ (2, 2) & (1, 1) & & \\ (0, 2) & (1, 2) & (1, 1.5) & \\ (4, 0) & (2, 1) & (1.5, 1.5) & (1.25, 1.5). \end{array}$$

Iz sheme preberemo, da je  $\mathbf{b}(0.5) = (1.25, 1.5)$ .

**2.2. Lastnosti Bernsteinovih polinomov in Bézierovih krivulj.** Za začetek si pogledjmo nekaj zvez in lastnosti, ki veljajo za Bernsteinove polinome:

**Lema 2.5.** *Za Bernsteinove polinome velja naslednja zveza*

$$B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t).$$

To smo dokazali že v dokazu trditve 2.1, enačba (1).

**Lema 2.6** (Razčlenitev enote). *Bernsteinovi polinomi se seštejejo v 1,*

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

*Dokaz.* Lemo enostavno dokažemo na sledeč način:

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

**Lema 2.7.** *Parameter  $t$  lahko izrazimo v Bernsteinovi bazi, saj velja naslednja zveza*

$$\sum_{i=0}^n \frac{i}{n} B_i^n(t) = t.$$



*Dokaz.* Razpišemo člen v vsoti in z zamikom indeksa ter uporabo lastnosti particije enote dobimo želeni rezultat:

$$\begin{aligned} \sum_{i=0}^n \frac{i}{n} B_i^n(t) &= \sum_{i=0}^n \frac{i}{n} \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} = \sum_{i=1}^n \binom{n-1}{i-1} (1-t)^{n-i} t^i \\ &= \sum_{j=0}^{n-1} \binom{n-1}{j} (1-t)^{n-1-j} t^{j+1} = t \sum_{j=0}^{n-1} B_j^{n-1}(t) \\ &= t. \end{aligned} \quad \square$$

Naslednje lastnosti Bézierovih krivulj sledijo iz de Casteljauovega algoritma in lastnosti Bernsteinovih polinomov.

- Krivulja je *afino invariantna*.

Afina invarianca pomeni, da dobimo enak rezultat, če najprej točke na krivulji  $\mathbf{b}$  izračunamo in jih nato preslikamo z afino preslikavo, ali pa če najprej afino preslikamo kontrolne točke in potem na preslikanih kontrolnih točkah izračunamo točke na krivulji. Invarianca je posledica de Casteljauovega algoritma, ki uporablja linearno interpolacijo, ta pa je afino invariantna. Ta lastnost je uporabna, ko želimo na primer rotirati krivuljo. Če izrišemo krivuljo na kontrolnih točkah in naredimo 100 izračunov točk na krivulji ter jo želimo rotirati, moramo preslikati vseh 100 izračunanih točk. Lahko pa najprej rotiramo začetne kontrolne točke in na njih izrišemo krivuljo, kar nam prihrani kar nekaj časa.

- *Interpolacija kontrolnih točk.*

Bézierova krivulja se začne pri  $t = 0$  v točki  $\mathbf{b}_0$  in konča pri  $t = 1$  v točki  $\mathbf{b}_n$ . To je zelo pomembno pri oblikovanju, saj imamo kontrolo nad začetkom in koncem. Rečemo, da krivulja interpolira začetno in končno točko.

- *Invarianca glede na afino transformacijo parametra.*

Napačno je mišljenje, da je krivulja definirana samo na intervalu  $[0, 1]$ , čeprav je ta interval najpogosteje uporabljen. De Casteljauov algoritem namreč uporablja samo razmerja in zato interval, na katerem je krivulja definirana, ni pomemben. Krivulja  $\mathbf{b} : [a, b] \rightarrow \mathbb{R}^2$  je definirana s predpisom

$$\mathbf{b}(u) = \sum_{i=0}^n B_i^n \left( \frac{u-a}{b-a} \right) \mathbf{b}_i, \quad u \in [a, b].$$

Pri de Casteljauovem algoritmu vzamemo  $t = \frac{u-a}{b-a}$  in računamo

$$\mathbf{b}_i^r(u) = \frac{b-u}{b-a} \mathbf{b}_i^{r-1}(u) + \frac{u-a}{b-a} \mathbf{b}_{i+1}^{r-1}(u),$$

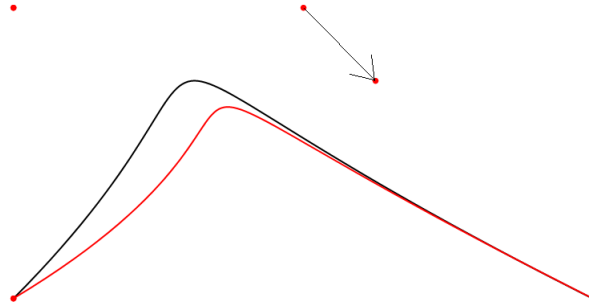
za  $r = 0, 1, \dots, n$  in  $i = 0, 1, \dots, n-r$ . Prehod med intervaloma  $[0, 1]$  in  $[a, b]$  je linearna reparametrizacija intervala.

- *Reparametrizacija.* Reparametrizacija krivulje pomeni, da spremenimo kako je krivulja parametrizirana (sprememba vrednosti parametra je določena vsaki točki na krivulji). To naredimo s substitucijo parametra, npr. imamo krivuljo  $\mathbf{b}(t)$  in če naredimo zamenjavo  $t = f(s)$ , dobimo krivuljo  $\mathbf{c}(s)$ . Krivulji sta geometrijsko enaki in imata iste točke, vendar razmerje med vrednostjo parametra in točkami na krivulji sta različni. [3, poglavje 2.12]

- *Oblikovanje.*

Bézierova krivulja v nekem smislu sledi svojim kontrolnim točkam, saj se s

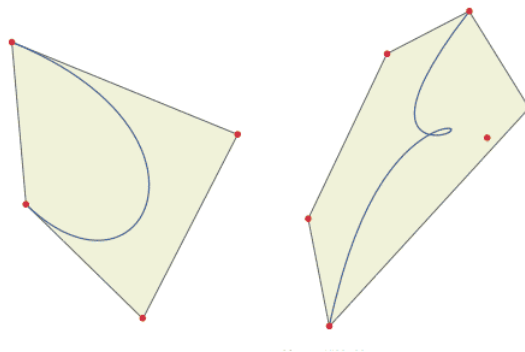
premikanjem kontrolnih točk premika tudi krivulja. S premikom ene same kontrolne točke se bo krivulja najbolj spremenila v lokalni okolici te točke. To nam je v veliko pomoč pri oblikovanju krivulje. Če želimo samo majhno spremembo oblike krivulje, to dosežemo že s premikom najbližje kontrolne točke. Primer vidimo na sliki 4.



SLIKA 4. Sprememba krivulje pri premiku ene kontrolne točke.

- *Konveksna ovojnica.*

Konveksno ovojnico si predstavljamo kot območje znotraj lika, katerega oglišča so kontrolne točke. Za  $t \in [0, 1]$  leži krivulja v konveksni ovojnici kontrolnih točk, saj je vsaka točka konveksna kombinacija prejšnjih dveh točk in zato nikoli ne dobimo točke izven konveksne ovojnice. Ta lastnost sledi iz dejstva, da je vsota Bernsteinovih polinomov enaka 1 in da so ti nenegativni za  $t \in [0, 1]$ . Posledica tega je tudi dejstvo, da ravninske kontrolne točke vedno generirajo ravninsko krivuljo. To je pomembna lastnost, ki nam pomaga predvsem pri računanju presečišč krivulj. Na sliki 5 je označeno območje konveksne ovojnice.



SLIKA 5. Sivo območje predstavlja konveksno ovojnico krivulje.

- *Simetrija.*

Ni pomemben vrstni red označevanja točk, ali začnemo točke označevati od prve kontrolne točke naprej ali nazaj od zadnje kontrolne točke, saj sta krivulji s kontrolnimi točkami  $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$  ali  $\mathbf{b}_n, \mathbf{b}_{n-1}, \dots, \mathbf{b}_0$  enaki, razlika je le v orientaciji (v katero smer potujemo po krivulji). Točk med seboj pa ne smemo popolnoma premešati, saj pri tem dobimo neko drugo krivuljo.

2.3. **Razcvetna oblika Bézierove krivulje.** Pri izvajanju de Casteljauovega algoritma običajno za parameter  $t$  vzamemo vedno isto vrednost. Če pa posplošimo ta algoritem in na  $k$ -tem koraku namesto parametra  $t$  vzamemo parameter  $t_k$ , dobimo tako imenovani *razcvet krivulje*. Ta posplošitev nam pomaga pri nekaterih operacijah na Bézierovih krivuljah. Za začetek si pogledjmo primer razcveta.

**Primer 2.8.** Razcvet kubične krivulje, zapisan v de Casteljauovi shemi

$$\begin{array}{l} \mathbf{b}_0 \\ \mathbf{b}_1 \quad \mathbf{b}_0^1[t_1] \\ \mathbf{b}_2 \quad \mathbf{b}_1^1[t_1] \quad \mathbf{b}_0^2[t_1, t_2] \\ \mathbf{b}_3 \quad \mathbf{b}_2^1[t_1] \quad \mathbf{b}_1^2[t_1, t_2] \quad \mathbf{b}_0^3[t_1, t_2, t_3], \end{array}$$

in končna točka

$$\begin{aligned} \mathbf{b}_0^3[t_1, t_2, t_3] = & \mathbf{b}_0(1-t_1)(1-t_2)(1-t_3) + \mathbf{b}_1(t_1+t_2+t_3-2t_1t_2-2t_1t_3-2t_2t_3+ \\ & + 3t_1t_2t_3) + \mathbf{b}_2(t_1t_2+t_1t_3+t_2t_3-3t_1t_2t_3) + \mathbf{b}_3t_1t_2t_3, \end{aligned}$$

ki je funkcija treh neodvisnih spremenljivk.

Posplošitev na krivuljo stopnje  $n$  imenujemo razcvetna oblika Bézierove krivulje  $\mathbf{b}$ . Originalno krivuljo dobimo tako, da za parametre  $t_1, \dots, t_n$  vzamemo parameter  $t$ . Razcvet je simetrična polinomska funkcija, saj ni odvisna od vrstnega reda argumentov. Ni pomembno kako izbiramo vrstni red parametrov, ker na koncu je funkcija vedno ista.

**Primer 2.9.** V posebnem primeru iz razcveta Bézierove krivulje dobimo kontrolne točke

$$\begin{aligned} \mathbf{b}[0, 0, 0] &= \mathbf{b}_0, \\ \mathbf{b}[0, 0, 1] &= \mathbf{b}_1, \\ \mathbf{b}[0, 1, 1] &= \mathbf{b}_2, \\ \mathbf{b}[1, 1, 1] &= \mathbf{b}_3. \end{aligned}$$

Preverimo, da to res drži

$$\begin{array}{l} \mathbf{b}_0 \\ \mathbf{b}_1 \quad \mathbf{b}_0^1[0] = \mathbf{b}_0 \\ \mathbf{b}_2 \quad \mathbf{b}_1^1[0] = \mathbf{b}_1 \quad \mathbf{b}_0^2[0, 0] = \mathbf{b}_0 \\ \mathbf{b}_3 \quad \mathbf{b}_2^1[0] = \mathbf{b}_2 \quad \mathbf{b}_1^2[0, 0] = \mathbf{b}_1 \quad \mathbf{b}_0^3[0, 0, 0] = \mathbf{b}_0, \end{array}$$

$$\begin{array}{l} \mathbf{b}_0 \\ \mathbf{b}_1 \quad \mathbf{b}_0^1[0] = \mathbf{b}_0 \\ \mathbf{b}_2 \quad \mathbf{b}_1^1[0] = \mathbf{b}_1 \quad \mathbf{b}_0^2[0, 0] = \mathbf{b}_0 \\ \mathbf{b}_3 \quad \mathbf{b}_2^1[0] = \mathbf{b}_2 \quad \mathbf{b}_1^2[0, 0] = \mathbf{b}_1 \quad \mathbf{b}_0^3[0, 0, 1] = \mathbf{b}_1. \end{array}$$

Podobno naredimo za  $\mathbf{b}_2$  in  $\mathbf{b}_3$ .

V splošnem lahko katerokoli kontrolno točko krivulje izrazimo z razcvetom z enkami in ničlami. Natančno povedano, lahko  $i$ -to kontrolno točko na Bézierovi krivulji stopnje  $n$  izrazimo v razcvetni obliki  $\mathbf{b}_i = \mathbf{b}[0^{\langle n-i \rangle}, 1^{(i)}]$ , kjer zgornji indeks pove, kolikokrat se v zaporedju pojavi nič oz. ena.

De Casteljauov algoritem lahko definiramo tudi s pomočjo razcveta in tako dobimo naslednjo rekurzijo

$$\mathbf{b} [0^{\langle n-r-i \rangle}, t^{\langle r \rangle}, 1^{\langle i \rangle}] = (1-t)\mathbf{b} [0^{\langle n-r-i+1 \rangle}, t^{\langle r-1 \rangle}, 1^{\langle i \rangle}] + t\mathbf{b} [0^{\langle n-r-i \rangle}, t^{\langle r-1 \rangle}, 1^{\langle i+1 \rangle}],$$

za  $r = 0, 1, \dots, n$  in  $i = 0, 1, \dots, n-r$ , točka na krivulji pa je podana kot  $\mathbf{b}[t^{\langle n \rangle}]$  [1].

Videli smo, da je med lastnostmi krivulje tudi invarianca glede na afino transformacijo parametra, ki nam pove, da lahko krivuljo gledamo na poljubnem intervalu  $[a, b]$ . Prav tako lahko tudi razcvet naredimo nad istim intervalom in kontrolne točke se izražajo kot

$$\mathbf{b}_i = \mathbf{b} [a^{\langle n-i \rangle}, b^{\langle i \rangle}], \quad i = 0, 1, \dots, n.$$

**2.4. Subdivizija.** Krivuljo običajno definiramo na enotskem intervalu  $[0, 1]$ . Zdaj pa nas zanima, kako bi izrazili samo del osnovne Bézierove krivulje, recimo na intervalu  $[0, c]$  za  $c < 1$  kot samostojno Bézierovo krivuljo iste stopnje. Poiskati moramo ustrezne kontrolne točke, na katerih bomo definirali krivuljo in to naredimo s subdivizijo.

Naj bodo  $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$  kontrolne točke osnovne krivulje, kontrolne točke iskane krivulje pa označimo s  $\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n$ . Naša želja je, da je  $\mathbf{c}_0 = \mathbf{b}_0$ ,  $\mathbf{c}_n = \mathbf{b}^n(c)$  in da je del krivulje  $\mathbf{b}^n$  nad intervalom  $[0, c]$  natanko krivulja  $\mathbf{c}^n$ . Da dobimo kontrolne točke  $\mathbf{c}_j$ , si pomagamo z razcvetno obliko krivulje. Nad intervalom  $[0, c]$  se  $j$ -ta kontrolna točka izraža s

$$\mathbf{c}_j = \mathbf{b} [0^{\langle n-j \rangle}, c^{\langle j \rangle}], \quad j = 0, 1, \dots, n.$$

S tem dobimo kontrolne točke za del krivulje. V nerazcvetni obliki pa jih zapišemo kot  $\mathbf{c}_j = \mathbf{b}_0^j(c)$ . Zaradi lastnosti simetrije, ki jo ima Bézierova krivulja, pa velja, da so kontrolne točke, ki ustrezajo krivulji na intervalu  $[c, 1]$ , podane z

$$\mathbf{b}_j^{n-j}(c), \quad j = 0, 1, \dots, n, \quad \text{oz. z} \quad \mathbf{b} [c^{\langle n-j \rangle}, 1^{\langle j \rangle}], \quad j = 0, 1, \dots, n.$$

Tako dobljena formula se imenuje *subdivizijska formula* za Bézierovo krivuljo. Algoritem je pomemben in uporaben zato, ker nam opiše, kako Bézierovo krivuljo  $\mathbf{b}$  razdelimo na dva dela, katerih unija je ravno cela krivulja.

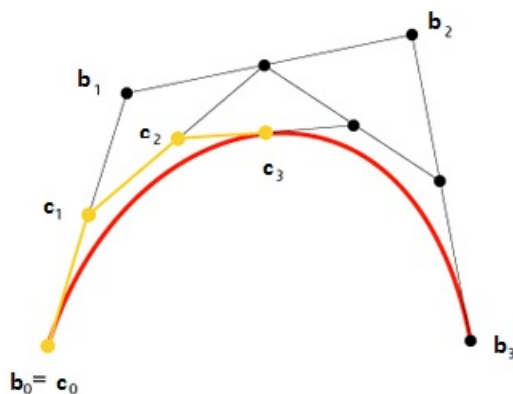
Subdivizijo lahko poljubno mnogokrat ponovimo. Vzemimo na primer parameter  $t = 1/2$ , na naslednjem koraku interval zopet razpolovimo in na  $k$ -tem koraku imamo že  $2^k$  kontrolnih poligonov. Ti skupaj opisujejo celotno prvotno Bézierovo krivuljo, vsak od njih pa le majhen del. Z naraščajočim  $k$  poligon hitro skonvergira h krivulji. Z deljenjem parametra  $t$  in risanjem primernih kontrolnih točk dobimo način za risanje krivulje.

**Primer 2.10.** Poglejmo si primer kubične krivulje, ki jo vidimo na sliki 6. Krivulja je definirana z naslednjimi kontrolnimi točkami

$$\mathbf{b}_0 = (-1, 0), \mathbf{b}_1 = (1, 5), \mathbf{b}_2 = (5, 6), \mathbf{b}_3 = (7, 0).$$

Krivuljo želimo deliti na pol in zato vzamemo za parameter  $t = 1/2$ . De Casteljauova shema pri  $t = 1/2$  je enaka

$$\begin{array}{llll} \mathbf{b}_0 = (-1, 0) & & & \\ \mathbf{b}_1 = (1, 5) & \mathbf{b}_0^1 = (0, 2.5) & & \\ \mathbf{b}_2 = (5, 6) & \mathbf{b}_1^1 = (3, 5.5) & \mathbf{b}_0^2 = (1.5, 4) & \\ \mathbf{b}_3 = (7, 0) & \mathbf{b}_2^1 = (6, 3) & \mathbf{b}_1^2 = (4.5, 4.25) & \mathbf{b}_0^3 = (3, 4.125). \end{array}$$



SLIKA 6. Primer subdivizije kubične krivulje za  $c = 1/2$ .

Zgornji postopek nam pove, da so iskane kontrolne točke prve polovice krivulje enake

$$\begin{aligned} \mathbf{c}_0 = \mathbf{b}_0 &= (-1, 0), \quad \mathbf{c}_1 = \mathbf{b}_0^1(0.5) = (0, 2.5), \\ \mathbf{c}_2 &= \mathbf{b}_0^2(0.5) = (1.5, 4), \quad \mathbf{c}_3 = \mathbf{b}_0^3(0.5) = (3, 4.125), \end{aligned}$$

ki so tudi označene na sliki 5. Kontrolne točke drugega dela krivulje pa so enake

$$\mathbf{b}_0^3(0.5) = (3, 4.125), \quad \mathbf{b}_1^2(0.5) = (4.5, 4.25), \quad \mathbf{b}_2^1(0.5) = (6, 3), \quad \mathbf{b}_3^0 = (7, 0).$$

**2.5. Spreminjanje stopnje krivulje.** Bézierovo krivuljo stopnje  $n$  lahko predstavimo kot Bézierovo krivuljo višje stopnje. Začetnim kontrolnim točkam s pravilnim dodajanjem nove kontrolne točke zvišamo stopnjo krivulje, oblika pa se ne spremeni. Natančneje stare kontrolne točke premaknemo in pravilno dodamo novo točko, samo začetna in končna kontrolna točka ostaneta na istem mestu. To dejstvo se uporablja predvsem pri oblikovanju s krivuljami. Recimo, da na nekem koraku ugotovimo, da s premikanjem danih kontrolnih točk ne dosežemo želene oblike krivulje. S postopkom višanja stopnje dodamo novo kontrolno točko in ustrezno spremenimo stare kontrolne točke. Nato pa nadaljujemo oblikovanje s premikanjem na novo izračunanih točk. Druga pomembna uporaba je naslednja. Če imamo dve krivulji različnih stopenj in želimo delati s krivuljama enakih stopenj, potem samo zvišamo stopnjo krivulji z nižjo stopnjo. Poglejmo si sedaj omenjeni postopek bolj natančno. Imamo Bézierovo krivuljo

$$\mathbf{b}(t) = \sum_{i=0}^n \mathbf{b}_i \binom{n}{i} (1-t)^{n-i} t^i.$$

Poiskati želimo take kontrolne točke  $\mathbf{b}_i^{(1)}$ ,  $i = 0, \dots, n+1$ , za krivuljo

$$\mathbf{b}^{(1)}(t) = \sum_{i=0}^{n+1} \mathbf{b}_i^{(1)} B_i^{n+1}(t)$$

stopnje  $n+1$ , da bosta prvotna krivulja in krivulja z dodano točko enaki. Prvotno enačbo krivulje  $\mathbf{b}$  pomnožimo s koeficientom  $(t + (1-t))$  in dobimo

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

Primerjamo desno stran te enačbe s parametrizacijo krivulje  $\mathbf{b}^{(1)}$  in dobimo naslednjo enakost:

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

Iz primerjave koeficientov na obeh straneh sledi

$$\mathbf{b}_i^{(1)} \binom{n+1}{i} = \mathbf{b}_i \binom{n}{i} + \mathbf{b}_{i-1} \binom{n}{i-1} \quad i = 0, \dots, n+1.$$

Kontrolne točke krivulje  $\mathbf{b}^{(1)}$ , ki je stopnje višja, so torej enake

$$\mathbf{b}_i^{(1)} = \frac{i}{n+1} \mathbf{b}_{i-1} + \left(1 - \frac{i}{n+1}\right) \mathbf{b}_i, \quad i = 0, \dots, n+1.$$

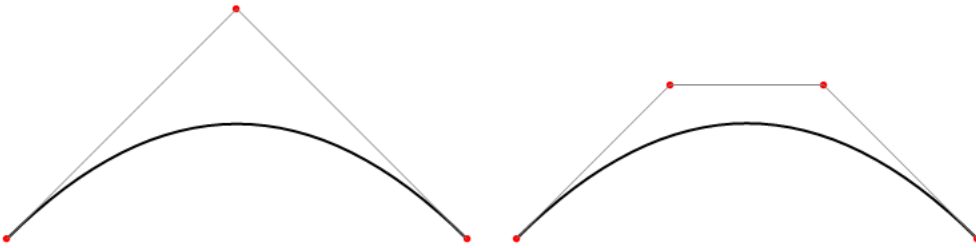
Iz zgornje zveze vidimo, da nov kontrolni poligon leži v konveksni ovojnici prejšnjega kontrolnega poligona in zato bližje krivulji. Če ta postopek ponavljamo, se kontrolni poligon vedno bolj približuje krivulji. Z zaporednim višanjem stopnje krivulje bo kontrolni poligon počasi skonvergirala h krivulji (vendar je red konvergence počasen in zato neuporaben v praksi).

**Primer 2.11.** Poglejmo si primer, kako zvišamo stopnjo Bézierovi krivulji stopnje 2. Naj bo krivulja definirana s kontrolnimi točkami

$$\mathbf{b}_0 = (2, 2), \quad \mathbf{b}_1 = (5, 5), \quad \mathbf{b}_2 = (8, 2).$$

Izračunamo nove kontrolne točke z zgornjim postopkom in dobimo

$$\begin{aligned} \mathbf{b}_0^{(1)} &= (2, 2), \\ \mathbf{b}_1^{(1)} &= \frac{1}{3} \mathbf{b}_0 + \frac{2}{3} \mathbf{b}_1 = \frac{1}{3} (2, 2) + \frac{2}{3} (5, 5) = (4, 4) \\ \mathbf{b}_2^{(1)} &= \frac{2}{3} \mathbf{b}_1 + \frac{1}{3} \mathbf{b}_2 = \frac{2}{3} (5, 5) + \frac{1}{3} (8, 2) = (6, 4) \\ \mathbf{b}_3^{(1)} &= (8, 2). \end{aligned}$$



SLIKA 7. Enaki krivulji, ki ju dobimo pri višanju stopnje: levo krivulja stopnje 2 in desno krivulja stopnje 3.

Podobno lahko izpeljemo nižanje stopnje z odvzemanom kakšne točke kontrolnega poligona. Vendar moramo paziti, saj ni nujno, da dobimo isto krivuljo, lahko le aproksimiramo dano krivuljo s krivuljo nižje stopnje.

**2.6. Racionalna Bézierova krivulja.** Do sedaj so bile vse kontrolne točke enako pomembne. Lahko pa jih razlikujemo in vsaki pripišemo skalar uteži in s tem dobimo racionalno Bézierovo krivuljo, ki jo opisuje naslednja definicija.

**Definicija 2.12.** Naj bodo  $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$  kontrolne točke,  $w_i$  utež  $i$ -te kontrolne točke in  $B_i^n$  Bernsteinovi polinomi. Potem je *racionalna Bézierova krivulja* podana z enačbo

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

Če so vse uteži enake 1 oz. so enake med seboj, preide racionalna krivulja v polinomsko Bézierovo krivuljo. Prednosti racionalne oblike krivulje pred polinomsko so sledeče:

- Večja kontrola oblike krivulje, saj že sprememba uteži vpliva na spremembo oblike krivulje.
- Pri projekciji prostorske polinomske Bézierove krivulje dobimo ravninsko racionalno krivuljo.
- Potrebne so pri natančnem opisu stožnic (polinomska Bézierova krivulja stopnje 2 predstavlja samo parabolo). Pri natančni predstavitvi krožnice potrebujemo racionalno obliko.

Racionalno Bézierovo krivuljo si lahko predstavljamo tudi kot projekcijo prostorske krivulje na ravnino (prostorsko krivuljo projiciramo na ravnino  $z = 1$ ).

Če želimo razširiti de Casteljaouv algoritem na racionalno krivuljo, najprej racionalno Bézierovo krivuljo prevedemo na ustrezno prostorsko polinomsko krivuljo. Nato izvedemo želeno »operacijo« na prostorski polinomski krivulji in na koncu preslikamo prostorsko krivuljo nazaj na ravninsko.

**2.7. Eksplicitna oblika.** Naj bo  $f : [0, 1] \rightarrow \mathbb{R}$ ,

$$f(x) = b_0 B_0^n(x) + \dots + b_n B_n^n(x) \quad b_i \in \mathbb{R}$$

funkcija zapisana v Bernsteinovi bazi. Zapis  $y = f(x)$  predstavlja eksplicitno podano krivuljo. Natančneje je eksplicitno podana krivulja graf funkcije  $f$ , to je množica točk  $\{(x, f(x)), x \in [0, 1]\}$ .

Ker velja lema 2.7, lahko eksplicitno obliko zapišemo v parametrični obliki kot

$$\mathbf{b}(t) = \sum_{i=0}^n (i/n, b_i) B_i^n(t).$$

Kot vidimo, so kontrolne točke oblike  $(i/n, b_i)$  za  $i = 0, \dots, n$  in prve koordinate točk so enakomerno razporejene na intervalu, kjer je krivulja definirana.

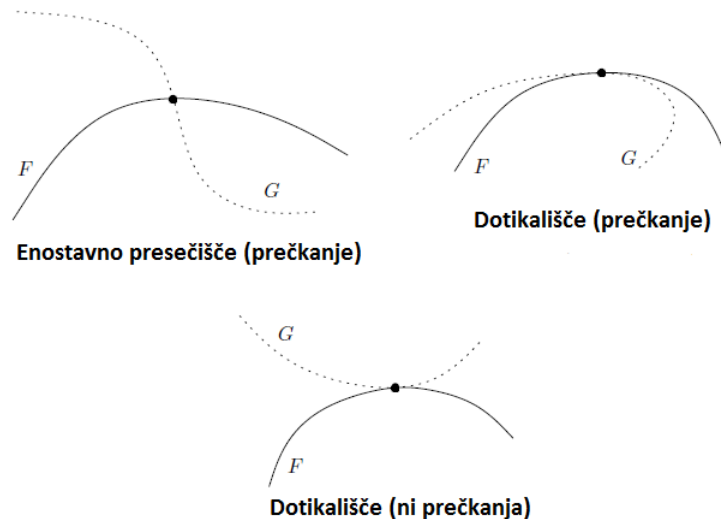
### 3. RAČUNANJE PRESEČIŠČ RAVNINSKIH BÉZIEROVIH KRIVULJ

Računanje presečišč in analiza algebraičnih krivulj in ploskev je osnovni problem na mnogih področjih geometrijskega modeliranja in robotike. Kot primer vzemimo dva robota, ki se gibljeta vsak po svoji Bézierovi krivulji. Zanima nas kje in kdaj se srečata oz. če se sploh kdaj.

**Definicija 3.1.** Algebraična krivulja je množica točk v Evklidski ravnini, katerih koordinate so ničle nekega polinoma dveh spremenljivk.

**Izrek 3.2** (Bézoutov izrek [3, poglavje 7.1]). *Ravninska algebraična krivulja stopnje  $n$  in druga ravninska algebraična krivulja stopnje  $m$  imata največ  $n \cdot m$  presečnih točk.*

**Opomba 3.3.** Pri preštevanju teh presečnih točk upoštevamo tudi kompleksne preseke, presečne večkratnosti in presečišča v neskončnosti. Če je presečišč več kot  $n \cdot m$ , potem sta krivulji enaki. Izrek lahko posplošimo tudi za računanje presečišč v višjih dimenzijah.



SLIKA 8. Primeri presečišč krivulj.

Presečišča lahko razdelimo glede na različne kriterije. Ločimo *enostavna presečišča* in *dotikališča*, ki so odvisna od tega ali tangenti na krivuljo v presečišču sovpadata ali ne, kjer tangenti sovpadata tam imamo dotikališča. Druga delitev pa je na presečišča, kjer se krivulji prečkata ali ne. Vsa presečišča imajo tudi različne večkratnosti. Na sliki 8 vidimo primere različnih presečišč.

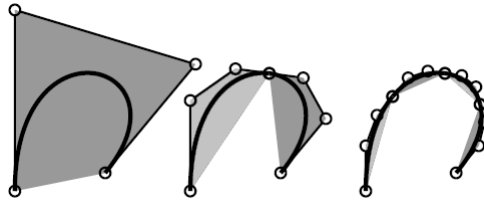
Poznamo več algoritmov za računanje presečišč. V tem delu bodo obravnavani Bézierov subdivizijski algoritem, Bézierovo izrezovanje in implicitizacija.

**3.1. Bézierov subdivizijski algoritem.** Algoritem temelji na dveh lastnostih krivulje. Prva lastnost, ki jo izkorišča algoritem za delovanje, je vsebovanost krivulje v konveksni ovojnici. Druga lastnost je subdivizija, ki deli krivuljo z uporabo de Casteljauovega algoritma. Na sliki 9 so prikazane konveksne ovojnice krivulje in delov krivulje pri subdiviziji.

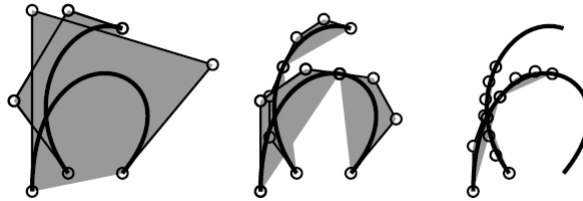
Osnovna ideja algoritma je, da gledamo preseke konveksnih ovojnic krivulj. Če se konveksni ovojnice ne sekata, se tudi krivulji ne sekata. Pare krivulj katerih konveksne ovojnice se ne sekajo, izločimo. V primeru, da se sekata, krivulji razdelimo na dva dela in postopek ponovimo na vsakem delu posebej. S ponavljanjem postopka dobimo vedno večji del, kjer se krivulji ne sekata in območje, kjer je presečišče, se manjša. Primer algoritma je prikazan na sliki 10, kjer vidimo tri korake algoritma.

Po dovolj korakih subdivizije pridemo do koraka, ko krivuljo delimo že na dovolj majhen del, da jo lahko dobro aproksimiramo s premico. Presečišče začetnih krivulj





SLIKA 9. Konveksna ovojnica Bézierove krivulje in konveksne ovojnice po delitvi na pol in na četrtine.



SLIKA 10. Trije koraki Bézierovega subdivizijskega algoritma.

je približno enako presečišču aproksimacijskih premic, ki potekata skozi začetno in končno točko ustreznega dela subdiviziranih krivulj.

Velika pomanjkljivost algoritma je ta, da je računanje konveksnih ovojnic težko. Zato v praksi namesto konveksne ovojnice uporabljamo pravokotnike. Izberemo najmanjši možni pravokotnik, v katerega lahko vložimo krivuljo in njeno konveksno ovojnico, ter izvajamo zgoraj opisani algoritem.

**3.2. Implicitizacija.** Racionalno krivuljo lahko predstavimo na različne načine. Najpogostejši obliki krivulje sta parametrična  $\mathbf{r}(t) = \left( \frac{x(t)}{w(t)}, \frac{y(t)}{w(t)} \right)$ ,  $t \in I$ , kjer so  $x(t), y(t), w(t)$  polinomi, in implicitna  $F(x, y) = 0$ , kjer je  $F$  polinomska funkcija dveh spremenljivk. Prva oblika ima prednost pri hitrem računanju koordinat na krivulji (v pomoč predvsem pri risanju), z drugo obliko pa hitro določimo, ali dana točka leži na krivulji ali ne.

Naravno vprašanje, ki si ga zastavimo, je, ali lahko enostavno prehajamo iz ene oblike v drugo. Videli bomo, da lahko parametrično racionalno krivuljo prevedemo v implicitno obliko, medtem ko obratno lahko naredimo samo za krivulje stopnje ena ali dve.

*Problem implicitizacije* zahteva, da za dano parametrično krivuljo  $\mathbf{r}$  najdemo tako funkcijo  $F$ , za katero bo veljalo  $F(x, y) = 0$  za vse točke  $(x, y)$ , ki ležijo na krivulji  $\mathbf{r}$ . Funkcija  $F$  predstavlja implicitni zapis parametrične krivulje  $\mathbf{r}$ . Drugo vprašanje, ki si ga zastavimo, je sledeče. Recimo, da za točko  $T(x_0, y_0)$  velja  $F(x_0, y_0) = 0$ . Zanima nas vrednost parametra te točke, pri katerem je  $\mathbf{r}(t_0) = (x_0, y_0)$ . Ta problem imenujemo *problem inverzije*.

Kot bomo videli v nadaljevanju, sta oba zgornja problema povezana s problemom iskanja skupnih ničel dveh polinomov. V računalniško podprto geometrijsko oblikovanje sta prišla iz algebraične geometrije in skupaj ju uporabljamo za računanje presečišč dveh parametričnih krivulj.

3.2.1. *Rezultanta*. Preden pogledamo, kako iščemo skupne ničle dveh polinomov, si pogledjmo še nekaj definicij, ki nam bodo v pomoč pri računanju. Za začetek si pogledjmo rezultanto v standardni bazi, s pomočjo katere izračunamo implicitizacijo. Spodnja definicija rezultante je za polinome enakih stopenj, lahko pa jo tudi razširimo na polinome različnih stopenj. Z umetnim dodajanjem koeficientov pri polinomih ali z višanjem stopnje krivulj pri Bézierovih krivuljah dosežemo enake stopnje polinomov. V računalniško podprtem geometrijskem oblikovanju je velikokrat zahtevano, da sta polinoma iste stopnje.

**Definicija 3.4.** Naj bosta

$$f(t) = \sum_{i=0}^n a_i t^i \quad \text{in} \quad g(t) = \sum_{i=0}^n b_i t^i$$

polinoma. *Sylvestrova rezultanta* je definirana kot determinanta  $(2n) \times (2n)$  matrike s predpisom

$$R(f, g) = \det \begin{pmatrix} a_n & \dots & a_0 & 0 & \dots & 0 \\ 0 & a_n & \dots & a_0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_n & \dots & a_0 \\ b_n & \dots & b_0 & 0 & \dots & 0 \\ 0 & b_n & \dots & b_0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & b_n & \dots & b_0 \end{pmatrix}.$$

*Bézoutova rezultanta* je enaka determinanti Bézoutove matrike  $(r_{ij})_{i,j=1,\dots,n}$ , katere elementi so podani z enakostjo

$$\frac{f(x)g(y) - f(y)g(x)}{x - y} = \sum_{i,j=1}^n r_{ij} x^{i-1} y^{j-1}$$

oz. kot

$$r_{ij} = \sum_{k=1}^{m_{ij}} (a_{j+k-1} b_{i-k} - a_{i-k} b_{j+k-1}),$$

kjer so  $m_{ij} = \min\{i, n + 1 - j\}$ ,  $i, j = 1, \dots, n$ .

**Trditev 3.5** ([2, izrek 3.5]). *Rezultanta  $R(f, g)$  je enaka nič natanko tedaj, ko imata polinoma skupno ničlo.*

**Primer 3.6.** Naj bosta

$$f(t) = a_1 t + a_0 \quad \text{in} \quad g(t) = b_1 t + b_0$$

polinoma stopnje 1. Pogledjmo si, kako dobimo Bézoutovo rezultanto za  $f$  in  $g$ . Ničlo polinoma  $f$  in ničlo polinoma  $g$  lahko enostavno najdemo in sicer sta ničli enaki

$$t = -\frac{a_0}{a_1} \quad \text{in} \quad t = -\frac{b_0}{b_1}.$$

Po trditvi 3.5 sledi, da morata biti ničli enaki, če želimo, da imata polinoma skupno ničlo, in tako dobimo

$$-\frac{a_0}{a_1} = -\frac{b_0}{b_1} \quad \text{oz.} \quad a_1 b_0 - b_1 a_0 = 0.$$

Tako je

$$R_B(f, g) = a_1b_0 - b_1a_0,$$

kar je enak rezultat, kot ga dobimo po definiciji. Podobno lahko izpeljemo tudi za polinome višjih stopenj.

**Izrek 3.7** ([4]). *Bézoutova rezultanta je do multiplikativne konstante +1 ali -1 enaka Sylvestrovei rezultanti.*

**Primer 3.8.** Poglejmo si obe obliki rezultante za polinoma stopnje 2 v standardni bazi. Naj bosta

$$f(t) = a_2t^2 + a_1t + a_0 \quad \text{in} \quad g(t) = b_2t^2 + b_1t + b_0$$

polinoma. V Sylvestrove rezultanto samo zapišemo koeficiente polinomov in ta je enaka

$$R_S(f, g) = \det \begin{pmatrix} a_2 & a_1 & a_0 & 0 \\ 0 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 \\ 0 & b_2 & b_1 & b_0 \end{pmatrix}$$

$$= a_2^2b_0^2 - a_1a_2b_0b_1 + a_0a_2b_1^2 + a_1^2b_0b_2 - 2a_0a_2b_0b_2 - a_0a_1b_1b_2 + a_0^2b_2^2.$$

Za Bézoutovo rezultanto izračunamo elemente po definiciji in tako dobimo naslednje elemente matrike

$$r_{11} = a_1b_0 - a_0b_1, \quad r_{12} = a_2b_0 - b_2a_0,$$

$$r_{21} = a_2b_0 - b_2a_0, \quad r_{22} = a_2b_1 - b_2a_1.$$

Z njimi izračunamo Bézoutovo rezultanto in dobimo

$$R_B(f, g) = \det \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} = \det \begin{pmatrix} a_1b_0 - a_0b_1 & a_2b_0 - b_2a_0 \\ a_2b_0 - b_2a_0 & a_2b_1 - b_2a_1 \end{pmatrix}$$

$$= -a_2^2b_0^2 + a_1a_2b_0b_1 - a_0a_2b_1^2 - a_1^2b_0b_2 + 2a_0a_2b_0b_2 + a_0a_1b_1b_2 - a_0^2b_2^2.$$

Vidimo, da sta rezultanti res samo drugače predznačeni.

Obe rezultanti imata svoje prednosti in slabosti. Bézoutova rezultanta je determinanta matrike velikosti  $n \times n$ , ki je simetrična, vendar je njene koeficiente težko izračunati. Po drugi strani pa lahko elemente Sylvesterove rezultante enostavno preberemo iz polinoma, je pa velikost matrike, katere determinanto moramo izračunati, dvakrat večja.

**3.2.2. Implicitizacija.** S pomočjo rezultante, ki nam pove, ali imata polinoma isto ničlo ali ne, naredimo implicitizacijo. Naj bo krivulja podana parametrično kot  $\mathbf{r}(t) = \left( \frac{x(t)}{w(t)}, \frac{y(t)}{w(t)} \right)$ , kjer so  $x(t), y(t)$  in  $w(t)$  polinomi. Definiramo dva polinoma dveh spremenljivk

$$p(x, t) = p_x(t) = w(t)x - x(t) \quad \text{in} \quad q(y, t) = q_y(t) = w(t)y - y(t).$$

Opazimo, da velja  $p_x(t) = q_y(t) = 0$  samo za vrednosti  $x, y$  in  $t$ , ki zadoščajo enačbama  $x = \frac{x(t)}{w(t)}$  in  $y = \frac{y(t)}{w(t)}$ . Na  $p_x(t)$  in  $q_y(t)$  gledamo kot na polinoma v spremenljivki  $t$ , katerih koeficienti so linearni v  $x$  oz.  $y$ .

Naj bodo

$$x(t) = \sum_{i=0}^n a_i t^i, \quad y(t) = \sum_{i=0}^n b_i t^i, \quad w(t) = \sum_{i=0}^n d_i t^i.$$

Potem je

$$p_x(t) = (d_n x - a_n)t^n + (d_{n-1}x - a_{n-1})t^{n-1} + \dots + (d_1x - a_1)t + (d_0x - a_0)$$

in

$$q_y(t) = (d_n y - a_n)t^n + (d_{n-1}y - a_{n-1})t^{n-1} + \dots + (d_1y - a_1)t + (d_0y - a_0).$$

Izračunamo rezultanto zgornjih dveh polinomov, ki nam poda polinom v spremenljivki  $x$  in  $y$  in jo označimo s  $f(x, y)$ . Za vsak par  $(x, y)$ , za katerega velja  $f(x, y) = 0$ , obstaja skupna ničla polinomov  $p_x$  in  $q_y$ . Natančneje, obstaja vrednost parametra  $t$ , pri kateri je  $p_x(t) = q_y(t) = 0$ . Drugače povedano, vsak par  $(x, y)$ , ki zadošča  $f(x, y) = 0$ , leži na parametrični krivulji in  $f(x, y) = 0$  je implicitna oblika te krivulje.

3.2.3. *Implicitizacija Bézierove krivulje.* V podrazdelku 3.3.2 smo videli pretvorbo parametrične krivulje, ki je zapisana v standardni bazi, v implicitno obliko. Prav tako želimo Bézierove krivulje, ki so zapisane v Bernsteinovi bazi, prevesti v implicitno obliko, da bomo lahko računali presečišča. Postopek smo si ogledali že v prejšnjem razdelku, vendar se pojavi vprašanje, kako zapisati in izračunati rezultanto v Bernsteinovi bazi. Seveda bi lahko Bernsteinovo bazo pretvorili v standardno bazo ter nato izračunali rezultanto. Da se izognemo preračunavanju baz, si raje pogledjmo zapis rezultante v Bernsteinovi bazi.

Spomnimo se definicije Bernsteinovih polinomov

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

in definicije Bézierove krivulje

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

Naj bo

$$u = \frac{t}{1-t}.$$

Delimo  $B_i^n(t)$  z  $(1-t)^n$  in dobimo

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

Definirajmo

$$\mathbf{b}^*(u) = \sum_{i=0}^n \mathbf{a}_i u^i, \quad \mathbf{a}_i = \binom{n}{i} \mathbf{b}_i,$$

s tem smo reparametrizirali krivuljo in tako prevedli polinom v Bernsteinovi bazi s spremenljivko  $t$  v standardno bazo v spremenljivki  $u$ . Velja, da je

$$\mathbf{b}^*(u) = \mathbf{b}(t)/(1-t)^n$$

in  $\mathbf{b}(t)$  ima ničlo pri  $t = t_0$  natanko tedaj, ko ima  $\mathbf{b}^*(u)$  ničlo pri  $u = u_0 = t_0/(1-t_0)$ . Rezultanto za  $\mathbf{b}^*(u)$  že znamo izračunati. Postopek nas pripelje do naslednje oblike implicitizacije za Bézierovo krivuljo.

**Trditev 3.9** ([2, poglavje 6]). *Za splošno racionalno Bézierovo krivuljo stopnje  $n$  je implicitna oblika enaka:*

$$f(x, y) = \det \begin{bmatrix} L_{n-1, n-1}(x, y) & \dots & L_{0, n-1}(x, y) \\ \vdots & \dots & \vdots \\ L_{n-1, 0}(x, y) & \dots & L_{0, 0}(x, y) \end{bmatrix} = 0,$$

kjer so

$$L_{i, j} = \sum_{k+m=i+j+1}^{m \leq \min(i, j)} \ell_{km}$$

in

$$\ell_{ij}(x, y) = \binom{n}{i} \binom{n}{j} w_i w_j \begin{vmatrix} x & y & 1 \\ x_i & y_i & 1 \\ x_j & y_j & 1 \end{vmatrix},$$

vrednosti  $x_i, y_i, w_i$  pa predstavljajo koordinate in uteži  $i$ -te kontrolne točke.

**Primer 3.10.** Imamo polinomske Bézierove krivulje  $\mathbf{b}$ , ki jo definirajo kontrolne točke

$$\mathbf{b}_0 = (1, 0), \quad \mathbf{b}_1 = (-1, 5), \quad \mathbf{b}_2 = (3, 0), \quad \mathbf{b}_3 = (10, 3).$$

Vse uteži so torej enake 1. Elementi matrike, ki jo potrebujemo za implicitno obliko, so sledeči:

$$\ell_{32}(x, y) = \binom{3}{3} \binom{3}{2} w_3 w_2 \begin{vmatrix} x & y & 1 \\ x_3 & y_3 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = 3 \begin{vmatrix} x & y & 1 \\ 10 & 3 & 1 \\ 3 & 0 & 1 \end{vmatrix} = -27 + 9x - 21y,$$

$$\ell_{31}(x, y) = 3 \begin{vmatrix} x & y & 1 \\ 10 & 3 & 1 \\ -1 & 5 & 1 \end{vmatrix} = 159 - 6x - 33y,$$

$$\ell_{30}(x, y) = \begin{vmatrix} x & y & 1 \\ 10 & 3 & 1 \\ 1 & 0 & 1 \end{vmatrix} = 3x - 9y - 3,$$

$$\ell_{21}(x, y) = 9 \begin{vmatrix} x & y & 1 \\ 3 & 0 & 1 \\ -1 & 5 & 1 \end{vmatrix} = -45x - 36y + 135,$$

$$\ell_{20}(x, y) = 3 \begin{vmatrix} x & y & 1 \\ 3 & 0 & 1 \\ 1 & 0 & 1 \end{vmatrix} = -6y,$$

$$\ell_{10}(x, y) = 3 \begin{vmatrix} x & y & 1 \\ -1 & 5 & 1 \\ 1 & 0 & 1 \end{vmatrix} = 15x + 6y - 15.$$

Implicitna oblika krivulje je tako enaka

$$\begin{aligned}
f(x, y) &= \begin{vmatrix} \ell_{32}(x, y) & \ell_{31}(x, y) & \ell_{30}(x, y) \\ \ell_{31}(x, y) & \ell_{30}(x, y) + \ell_{21}(x, y) & \ell_{20}(x, y) \\ \ell_{30}(x, y) & \ell_{20}(x, y) & \ell_{10}(x, y) \end{vmatrix} \\
&= \begin{vmatrix} -27 + 9x - 21y & 159 - 6x - 33y & 3x - 9y - 3 \\ 159 - 6x - 33y & 3x - 9y - 3 - 45x - 36y + 135 & -6y \\ 3x - 9y - 3 & -6y & 15x + 6y - 15 \end{vmatrix} \\
&= 431487 - 493371x + 67716x^2 - 5832x^3 - 308124y + 160866xy \\
&\quad - 2916x^2y + 64476y^2 - 486xy^2 - 27y^3 = 0.
\end{aligned}$$

3.2.4. *Inverzija.* Vzemimo točko  $T(x_0, y_0)$ , za katero vemo, da leži na parametrični krivulji  $\mathbf{r}$ . Zanima nas parameter  $t_0$ , tako da bo  $\mathbf{r}(t_0) = (x_0, y_0)$ . Rešitev tega problema sledi iz naslednje definicije in izreka.

**Definicija 3.11.** Naj bo  $R(f, g)$  rezultanta polinomov  $f$  in  $g$ . *Kofaktor*  $R_{(i,j)}(f, g)$  je poddeterminanta pripadajoče matrike, ki ji izbrišemo  $i$ -to vrstico in  $j$ -ti stolpec, pomnožena s predznakom  $(-1)^{i+j}$ .

**Izrek 3.12** ([5, izrek 1.1.2]). *Označimo z  $R_B(p_x, q_y)$  Bézoutovo determinanto. Če imata polinoma  $p_x(t)$  in  $q_y(t)$  stopnje  $n$  natanko eno skupno ničlo  $t_0$ , potem je*

$$t_0 = \frac{R_{B(n-1, n-2)}(p_x, q_y)}{R_{B(n-1, n-1)}(p_x, q_y)}.$$

*Splošneje: Če imata polinoma  $f(t), g(t)$  natanko  $j$  skupnih ničel, potem te ničle lahko najdemo tako, da izvedemo Gaussovo eliminacijo na Bézoutovi matriki. Po končani eliminaciji bo imela zadnja neničelna vrstica obliko  $(0, \dots, 0, h_j, \dots, h_1, h_0)$ ,  $h_j \neq 0$  in skupna ničla polinomov  $f(t), g(t)$  bo ničla polinoma*

$$h(t) = \sum_{k=0}^j t^k h_k.$$

3.2.5. *Inverzija Bézierove krivulje.* V razdelku 3.3.4. smo videli inverzijo za polinome v standardni bazi. Zopet se pojavi vprašanje, kako to prevesti na Bernsteinovo bazo. Podobno kot pri implicitizaciji, ko smo prevedli polinome iz Bernsteinove baze v standardno bazo, izračunamo po izreku 3.10 inverzijo za

$$\mathbf{b}^*(u) = \sum_{i=0}^n \binom{n}{i} \mathbf{b}_i u^i,$$

in dobimo parameter  $u_0$ . Spomnimo se, da smo  $u$  definirali kot

$$u = \frac{t}{1-t}.$$

Iskani parameter  $t_0$  je tako enak

$$t_0 = \frac{u_0}{1+u_0}.$$

Paziti moramo na ničle pri  $t = 1$  (takrat dobimo končno točko krivulje), saj pride pri  $u$  do deljenja z 0.

**Primer 3.13.** Poglejmo si splošen primer implicitizacije in inverzije za Bézierovo krivuljo stopnje 2. Implicitno obliko dobimo po trditvi 3.9 in označimo rezultanto, ki jo računamo, z  $R_B(b_x, b_y)$ .

$$f(x, y) = R_B(b_x, b_y) = \begin{vmatrix} \ell_{21}(x, y) & \ell_{20}(x, y) \\ \ell_{20}(x, y) & \ell_{10}(x, y) \end{vmatrix} = 0.$$

Za izračun inverzije uporabimo izrek in dobimo

$$u = \frac{R_{B(1,0)}(b_x, b_y)}{R_{B(1,1)}(b_x, b_y)} = \frac{-\ell_{20}}{\ell_{21}}.$$

Iz zveze med  $u$  in  $t$  dobimo, da je

$$t = \frac{\ell_{20}}{\ell_{20} - \ell_{21}}.$$

**Primer 3.14.** Inverzija Bézierove krivulje  $\mathbf{b}$  s kontrolnimi točkami kot v primeru 3.10 je sledeča. Uporabimo rezultanto, ki smo jo izračunali že v primeru 3.10 in po izreku za inverzijo sledi, da je

$$\begin{aligned} u &= \frac{R_{B(2,1)}(b_x, b_y)}{R_{B(2,2)}(b_x, b_y)} = \frac{- \begin{vmatrix} l_{32}(x, y) & l_{30}(x, y) \\ l_{31}(x, y) & l_{20}(x, y) \end{vmatrix}}{\begin{vmatrix} l_{32}(x, y) & l_{31}(x, y) \\ l_{31}(x, y) & l_{30}(x, y) + l_2(x, y) \end{vmatrix}} \\ &= \frac{- \begin{vmatrix} -27 + 9x - 21y & 3x - 9y - 3 \\ 159 - 6x - 33y & -6y \end{vmatrix}}{\begin{vmatrix} -27 + 9x - 21y & 159 - 6x - 33y \\ 159 - 6x - 33y & 3x - 9y - 3 - 45x - 36y + 135 \end{vmatrix}} \\ &= \frac{477 - 495x + 18x^2 + 1494y - 9xy - 171y^2}{-28845 + 4230x - 414x^2 + 8937y + 81xy - 144y^2}. \end{aligned}$$

Iz zveze med  $u$  in  $t$  dobimo, da je

$$t = \frac{u}{1 + u} = \frac{-53 - 2x^2 - 166y + 19y^2 + x(55 + y)}{3152 + 44x^2 - 1159y + 35y^2 - x(415 + 8y)}.$$

3.2.6. *Računanje presečišč.* Zdaj, ko poznamo metode za pretvarjanje krivulj iz parametrične v implicitno obliko in računanje parametrov, si pogledimo, kako nam to pomaga pri računanju presečišč dveh krivulj. Če imamo dve parametrični krivulji, potem eno spremenimo v implicitno obliko po postopku, ki smo ga spoznali. Ko imamo eno krivuljo v implicitni obliki  $f(x, y) = 0$  in drugo v parametrični obliki  $(x(t), y(t))$ , zamenjamo vse  $x$  in  $y$  v implicitni obliki z  $x = x(t)$  in  $y = y(t)$ . S to substitucijo dobimo polinom oblike  $g(t) = f(x(t), y(t))$ , čigar ničle so vrednosti parametra  $t$  v presečnih točkah. Koordinate teh točk enostavno izračunamo iz parametrične enačbe druge krivulje, medtem ko vrednosti parametra na prvi krivulji izračunamo po formuli za inverzijo.

**Primer 3.15.** Poglejmo si primer računanja presečišč dveh Bézierovih krivulj s pomočjo implicitizacije. Naj bo  $\mathbf{b}$  prva Bézierova krivulja, podana z naslednjimi kontrolnimi točkami

$$\mathbf{b}_0 = (1, 0), \mathbf{b}_1 = (-1, 5), \mathbf{b}_2 = (3, 0), \mathbf{b}_3 = (10, 3).$$

Njeno implicitno obliko in inverzijo smo že izračunali v primeru 3.10 in primeru 3.14. Drugo Bézierovo krivuljo  $\mathbf{c} = (c_x, c_y)$  naj definirajo naslednje kontrolne točke

$$\mathbf{c}_0 = (0, 0), \quad \mathbf{c}_1 = (0, 5), \quad \mathbf{c}_2 = (10, 4), \quad \mathbf{c}_3 = (5, 3),$$

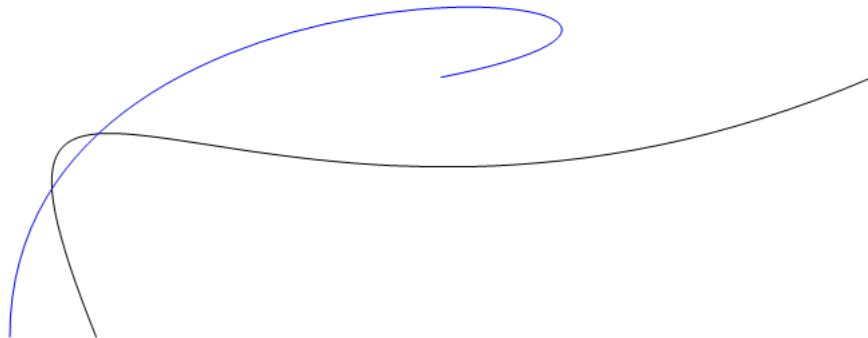
njena parametrična oblika je enaka

$$c_x(u) = 30(1-u)u^2 + 5u^3, \quad c_y(u) = 15(1-u)^2u + 12(1-u)u^2 + 3u^3.$$

Po Bézoutovem izreku imata ti dve krivulji največ 9 presečišč. Iz slike 11 je razvidno, da imata krivulji natanko dve presečišči. V implicitno obliko  $f(x, y) = 0$  prve krivulje vstavimo parametrično obliko druge krivulje  $x = c_x(u)$  ter  $y = c_y(u)$  in dobimo naslednji polinom stopnje 9:

$$f(x, y) = f(c_x(u), c_y(u)) = 81(5327 - 57060u + 64842u^2 + 592186u^3 - 700056u^4 - 536016u^5 - 464080u^6 + 3468096u^7 - 3359232u^8 + 995328u^9).$$

Ničle tega polinoma nam dajo parameter druge krivulje, kjer se ti dve krivulji sekata. Dobimo 6 kompleksnih ničel, ničlo, ki ne leži na intervalu  $[0, 1]$ , in ničli  $u_1 = 0.135387$  in  $u_2 = 0.202739$ .



SLIKA 11. Krivulja  $\mathbf{b}$  (črna) in krivulja  $\mathbf{c}$  (modra) iz primera 3.15.

Iz parametrične oblike druge krivulje enostavno dobimo presečni točki

$$T_1 = (0.487849, 1.71576) \text{ in } T_2 = (1.02476, 2.35123).$$

Sedaj potrebujemo samo še parametra presečnih točk prve krivulje, ki ju izračunamo s formulo za inverzijo in tako dobimo parametra presečnih točk prve krivulje  $t_1 = 0.156531$  in  $t_2 = 0.338395$ . Vsi rezultati so zapisani tudi v tabeli 1.

Presečna točka	Vrednost parametra $t$	Vrednost parametra $u$
(0.487849, 1.71576)	0.156531	0.135387
(1.02476, 2.35123)	0.338395	0.202739

TABELA 1. Presečišča krivulj iz primera 3.15.



**3.3. Bézierovo izrezovanje.** Bézierovo izrezovanje je uporabno tudi pri reševanju nekaterih drugih problemov kot so npr. iskanje presečišč algebrskih krivulj, računanje tangentskih presečišč itd. Konvergenca tega algoritma je kvadratična [3]. Kot že ime pove, bo ta metoda iskanja presečišč poiskala območje, ki ne seka krivulje in ga zavrgla ter nadaljevala iskanje presečišč na manjšem območju. V nadaljevanju bo predstavljenih nekaj metod, ki se potrebujejo pri izvedbi algoritma.

**3.3.1. Pas, ki obdaja krivuljo.** Najprej pogledjmo, kako poiščemo območje, ki omejuje krivuljo. Algoritem za iskanje presečišč začne z računanjem območja, ki omejuje eno od obeh Bézierovih krivulj.

Označimo z  $L$  premico, ki poteka med začetno kontrolno točko  $\mathbf{b}_0$  in končno kontrolno točko  $\mathbf{b}_n$ , in je definirana z naslednjo implicitno enačbo

$$ax + by + c = 0, \quad a^2 + b^2 = 1.$$

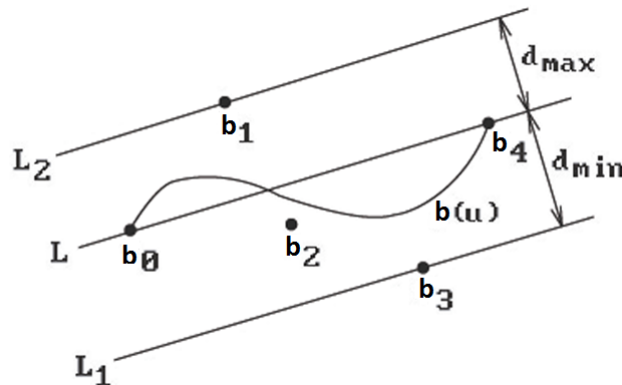
Predznačena razdalja  $d(x, y)$  od katerekoli točke  $(x, y)$  do premice  $L$  je enaka

$$d(x, y) = ax + by + c.$$

Označimo z  $d_i = d(x_i, y_i)$  razdaljo med  $L$  in kontrolno točko  $\mathbf{b}_i = (x_i, y_i)$ . Pas, ki omejuje dano Bézierovo krivuljo, je množica točk v ravnini, definirana kot

$$P = \{(x, y) | d_{min} \leq d(x, y) \leq d_{max}\},$$

kjer je  $d_{min} = \min\{d_0, \dots, d_n\}$  in  $d_{max} = \max\{d_0, \dots, d_n\}$ . Primer pasu, ki omejuje krivuljo, vidimo na sliki 12.



SLIKA 12. Pas, ki obdaja Bézierovo krivuljo.

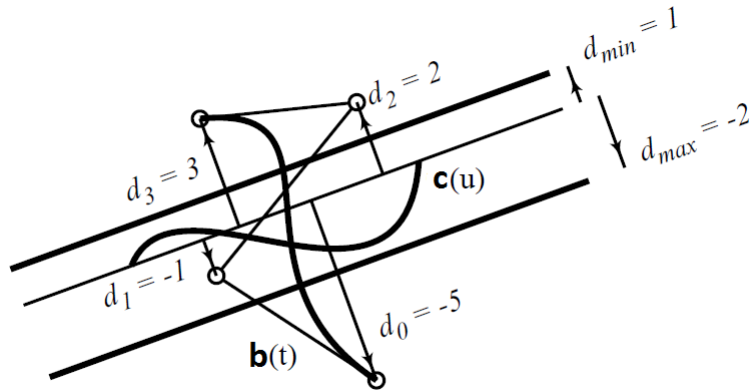
**3.3.2. Izrezovanje.** Recimo, da imamo dve polinomski kubični Bézierovi krivulji  $\mathbf{b}(t)$  in  $\mathbf{c}(u)$  ter pas  $P$ , ki pripada krivulji  $\mathbf{c}(u)$ . Pogledjmo si, kako poiščemo interval  $I$ , za katerega  $\mathbf{b}(t)$  leži zunaj  $P$  in posledično za katerega  $\mathbf{b}(t)$  ne seka  $\mathbf{c}(u)$ . Primer izrezovanja vidimo na sliki 3.3.2.

Krivulja  $\mathbf{b}$  naj bo podana parametrično s predpisom

$$\mathbf{b}(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n(t),$$

kjer so  $\mathbf{b}_i = (x_i, y_i)$  kontrolne točke in  $B_i^n$  Bernsteinovi polinomi. Če je premica  $L$  skozi točki  $\mathbf{c}_0$  in  $\mathbf{c}_n$  definirana z

$$ax + by + c = 0 \quad (a^2 + b^2 = 1),$$



SLIKA 13. Bézierovo izrezovanje.

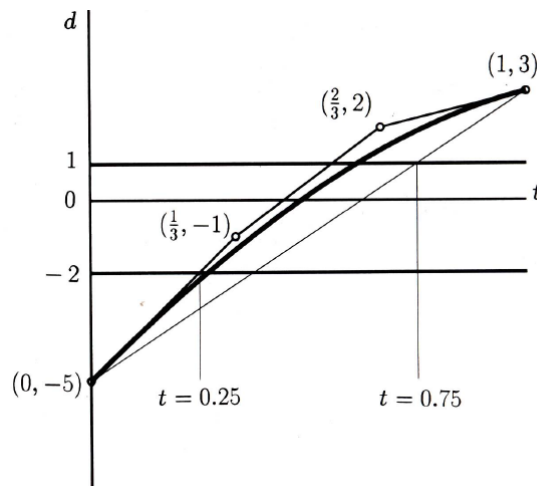
potem je razdalja  $d(t)$  od katerekoli točke na  $\mathbf{b}(t)$  do  $L$  podana kot

$$d(t) = \sum_{i=0}^n d_i B_i^n(t), \quad d_i = ax_i + by_i + c.$$

Razdalja  $d(t)$  je enaka 0 za natanko tiste vrednosti  $t$ , kjer  $\mathbf{b}$  seka  $L$ . Oznaka  $d_i$  predstavlja razdaljo med  $\mathbf{b}_i$  in  $L$ . Funkcija  $d(t)$  je polinomska v Bernsteinovi obliki in jo lahko predstavimo kot eksplicitno Bézierovo krivuljo, ki meri razdaljo med krivuljo in pasom  $P$ , z enačbo

$$\mathbf{D}(t) = (t, d(t)) = \sum_{i=0}^n (i/n, d_i) B_i^n(t) = \sum_{i=0}^n \mathbf{D}_i B_i^n(t).$$

Enačbo dobimo z upoštevanjem izražanja  $t$  v Bernsteinovi bazi po lemi 2.7 in iz funkcije, ki meri razdaljo med krivuljo  $\mathbf{b}(t)$  in premico  $L$ . Primer eksplicitne Bézierove krivulje vidimo na sliki 14.



SLIKA 14. Eksplicitno podana krivulja  $\mathbf{D}$ .

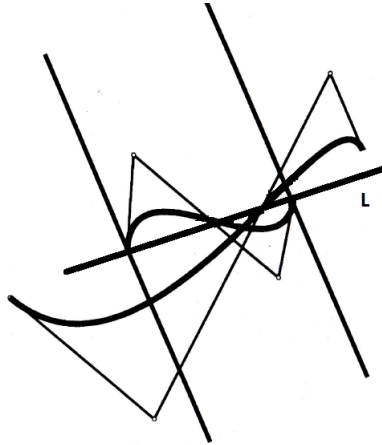
Vrednost parametra  $t$ , za katerega  $\mathbf{b}(t)$  leži zunaj pasu  $P$ , ustreza vrednosti parametra  $t$ , za katerega  $\mathbf{D}(t)$  leži nad  $d_{max}$  ali pod  $d_{min}$ . Torej lahko določimo interval

za parameter  $t$ , za katerega  $\mathbf{b}(t)$  zagotovo leži zunaj  $P$ , tako da določimo vrednost parametra  $t$ , za katerega konveksna ovojnica  $\mathbf{D}(t)$  leži nad  $d_{max}$  ali pod  $d_{min}$ .

Bézierovo izrezovanje je končano z dvakratno subdivizijo krivulje  $\mathbf{b}$  (uporabimo de Castaljaouov algoritem), pri tem pa odstranimo dela krivulje  $\mathbf{b}$ , za katera je  $t < d_{min}$  in  $t > d_{max}$ . Ta postopek deluje za polinomsko Bézierovo krivuljo, medtem ko za racionalno Bézierovo krivuljo generiranje eksplicitne Bézierove krivulje ni tako enostavno.

3.3.3. *Iteracija.* Pogledajmo si sedaj povezavo prejšnjih dveh postopkov in iskanje presečišč. Območje ene krivulje, ki zagotovo ne seka druge krivulje, lahko prepoznamo in zavržemo. Algoritem za iskanje presečišč Bézierovih krivulj iterativno ponavlja postopke Bézierovega izrezovanja enkrat za prvo, drugič za drugo krivuljo, dokler sta  $t_{min}$  in  $t_{max}$  različna ter tudi  $u_{min}$  in  $u_{max}$  različna do neke predpostavljene tolerance.

3.3.4. *Ostale definicije pasu  $P$ .* Definicija pasu, ki omejuje krivuljo, kot smo jo definirali na začetku, ni nujno vedno najbolj optimalna. Načeloma lahko vzamemo katerikoli pas, ki omejuje krivuljo, in z njim režemo dele krivulje, kjer zagotovo ni presečišč. Prej smo vzeli pas, ki je vzporeden premici  $L$  med točkama  $\mathbf{b}_0$  in  $\mathbf{b}_n$ , drugi način pa je, da vzamemo pas med pravokotnicama na premico med točkama  $\mathbf{b}_0$  in  $\mathbf{b}_n$ . Primer vidimo na sliki 15. V nekaterih primerih lahko s tako definicijo pasu odrežemo večji del krivulje, ki ga zavržemo. V algoritmu lahko tako pogledamo oba pasova in potem vzamemo tistega, ki nam pusti manjši del krivulje, s katerim delamo naprej.



SLIKA 15. Druga definicija pasu, ki omejuje krivuljo.

3.3.5. *Racionalne krivulje.* Pri racionalnih Bézierovih krivuljah moramo zgornji algoritem malo popraviti.

Naj bo  $\mathbf{b}$  racionalna Bézierova krivulja, definirana z

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

kjer so  $\mathbf{b}_i = (x_i, y_i)$  kontrolne točke z utežmi  $w_i$ . Funkcija, ki v algoritmu meri razdaljo, ima sedaj naslednjo obliko

$$d(t) = \sum_{i=0}^n d_i B_i^n(t), \quad d_i = w_i(ax_i + by_i + c),$$

kjer  $d_i$  predstavlja razdaljo med  $\mathbf{b}_i$  in  $L$ . Presek krivulje in pasu, ki omejuje krivuljo, zdaj ne more biti predstavljen kot  $P = \{(x, y) = \mathbf{b}(t) | d_{min} \leq d(x, y) \leq d_{max}\}$ , ampak mora biti izrez narejen neodvisno za zgornjo in spodnjo mejo pasu, ki omejuje krivuljo. Tako mora za območje parametra  $t$  veljati

$$d(t) < d_{max} \text{ in } d_{min} < d(t).$$

Vstavimo zgornjo funkcijo, ki meri razdaljo, preoblikujemo in tako dobimo

$$\sum_{i=0}^n (w_i(ax_i + by_i + c) - d_{max}) B_i^n(t) > 0$$

in

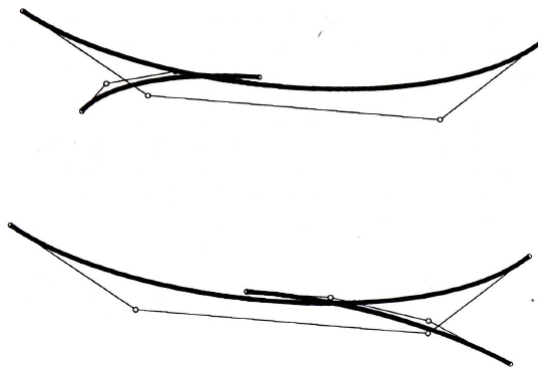
$$\sum_{i=0}^n (w_i(ax_i + by_i + c) + d_{min}) B_i^n(t) < 0.$$

To območje dobimo z uporabo Bézierovega izrezovanja kot pri polinomski krivulji.

3.3.6. *Večkrat sekajoči krivulji.* V primeru, ki ga vidimo na sliki 16, kjer se dve krivulji sekata v več kot eni presečni točki, metoda Bézierovega izrezovanja ne deluje, saj ne konvergira k nobeni presečni točki. Problem je v tem, da končne točke vsake krivulje ležijo v pasu druge krivulje in zato ne moremo izrezati dela krivulje, kjer ni preseka. Ta problem rešimo tako, da eno od krivulj razdelimo na pol in potem algoritem izvedemo za vsako polovico z drugo krivuljo posebej. Pas samo polovice krivulje nam nato izreže območje brez presečišč in algoritem lahko nemoteno izvajamo naprej. Primer delitve je predstavljen na sliki 17.



SLIKA 16. Krivulji z dvema presečiščema.



SLIKA 17. Delitev krivulje.

3.4. **Časovna zahtevnost.** Izkaže se, da imajo ti algoritmi različno časovno zahtevnost, ki pa je tudi odvisna od implementacije in zmožnosti računalnika. V tabeli 2 so predstavljeni relativni časi za izvedbo algoritmov, povzeti iz [3], kratica ‘Subdiv’ predstavlja Bézierov subdivizijski algoritem, ‘Impl’ - implicitizacijo ter ‘Izrez’ - Bézierovo izrezovanje.

Primer	Stopnja krivulje	Subdiv	Impl	Izrez
1	3	15	1	2.5
2	3	6	1	1.8
3	5	5	1.7	1
4	10	4	/	1

TABELA 2. Časovne zahtevnosti algoritmov.

Implicitizacija je uporabna samo za krivulje stopnje manjše od pet in z uporabo aritmetike z dvojno natančnostjo. Za višje stopnje je velika verjetnost, da ta algoritem postane zelo počasen in nam ne vrne dobrega rezultata. Za krivulje s stopnjo manjšo kot pet je implicitizacija 1-3 krat hitrejša od Bézierovega izrezovanja, ta pa je 2-10 krat hitrejši od Bézierovega subdivizijskega algoritma. Na splošno je najboljši algoritem za iskanje presečišč Bézierovih ravninskih krivulj stopnje večje kot štiri Bézierovo izrezovanje [3].

#### LITERATURA

- [1] G. Farin, *Curves and surfaces for computer aided geometric design: a practical guide*, 3rd ed., Computer Science and Scientific Computing **XVII**, Academic Press, Boston, 1993.
- [2] R. N. Goldman, T. W. Sederberg in D. C. Anderson, *Vector elimination: A technique for the implicitization, inversion and intersection of planar parametric rational polynomial curves*, *Comput. Aided Geom. Design* **1** (1984) 327–356.
- [3] T. W. Sederberg, *Computer aided geometric design courses notes*, verzija 23. 10. 2012, [ogled 15. 11. 2013], dostopno na <http://tom.cs.byu.edu/~557/text/cagd.pdf>.
- [4] B. Sturmfels, *Introduction to Resultants*, v: Applications of computational algebraic geometry (ur. D. A. Cox in B. Sturmfels), *Proceedings of Symposia in Applied Mathematics* **53**, Amer. Math. Soc., Providence, 1997, str. 25–39.
- [5] E. Žagar, *Racionalne parametrične krivulje*, diplomsko delo, Fakulteta za matematiko in fiziko, Univerza v Ljubljani, 1994.
- [6] E. Žagar, *Interpolacija s parametričnimi polinomskimi krivuljami*, verzija 6. 12. 2010, [ogled 15. 10. 2013], dostopno na <http://ucilnica1213.fmf.uni-lj.si/course/view.php?id=24>.
- [7] *Bézier curve*, [ogled 6. 11. 2013], dostopno na [http://en.wikipedia.org/wiki/Bezier\\_curve](http://en.wikipedia.org/wiki/Bezier_curve).
- [8] *Pierre Bézier*, [ogled 16. 1. 2014], dostopno na [http://en.wikipedia.org/wiki/Pierre\\_B%C3%A9zier](http://en.wikipedia.org/wiki/Pierre_B%C3%A9zier).
- [9] *Paul de Casteljau*, [ogled 16. 1. 2014], dostopno na [http://en.wikipedia.org/wiki/Paul\\_de\\_Casteljau](http://en.wikipedia.org/wiki/Paul_de_Casteljau).