

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga
Bézierove in Gordon-Coons krpe
(Bézier and Gordon-Coons patches)

Ime in priimek: Irena Čavdek
Študijski program: Matematika
Mentor: doc. dr. Vito Vitrih

Koper, september 2014

Ključna dokumentacijska informacija

Ime in PRIIMEK: Irena ČAVDEK

Naslov zaključne naloge: Bézierove in Gordon-Coons krpe

Kraj: Koper

Leto: 2014

Število listov: 46

Število slik: 12

Število prilog: 4

Število strani prilog: 8

Število referenc: 8

Mentor: doc. dr. Vito Vitrih

Ključne besede: Bézier, Gordon-Coons, krpa, krivulja, CAGD

Math. Subj. Class. (2010): 65D17, 65D18, 65D05, 68U07, 68U05

Izveček:

V zaključni nalogi so predstavljene Bézierove in Gordon-Coons krpe, ki so pomembne v računalniško podprtem geometrijskem oblikovanju (CAGD). Najprej so opisane Bézierove krivulje: podani so de Casteljauev algoritem, Bernsteinova oblika polinomskih Bézierovih krivulj in dokazane lastnosti teh krivulj. Sledi opis Bézierovih krp, kjer si podpoglavja sledijo v podobnem vrstnem redu kot pri krivuljah, za lažjo primerjavo krivulj in ploskev. V zadnjem poglavju so obravnavane Gordon-Coons krpe, podana je njihova izpeljava in težave, na katere naletimo pri tem. V zaključni nalogi so priložene tudi programske kode v jeziku Octave, s katerimi smo prišli do nekaterih rezultatov v nalogi. Večina snovi je povzeta po [1].

Key words documentation

Name and SURNAME: Irena ČAVDEK

Title of final project paper: Bézier and Gordon-Coons patches

Place: Koper

Year: 2014

Number of pages: 46

Number of figures: 12

Number of appendices: 4

Number of appendix pages: 8

Number of references: 8

Mentor: Assist. Prof. Vito Vitrih, PhD

Keywords: Bézier, Gordon-Coons, patch, curve, CAGD

Math. Subj. Class. (2010): 65D17, 65D18, 65D05, 68U07, 68U05

Abstract:

In the thesis Bézier and Gordon-Coons patches, which are important in Computer Aided Geometric Design (CAGD), are described. Firstly, we describe Bézier curves: we present de Casteljau algorithm, the Bernstein form of Bézier curve and provide proofs for some of the curve's properties. Third chapter, which is about Bézier patches, is divided into similar sections as previous chapter, which allows us to compare curves and surfaces in a better way. In the last chapter, Gordon-Coons patches, their definitions and some problems, which occur related to interpolation, are presented. In the thesis, some examples of Octave's code, that we used to get some of the results in the thesis, are included. The basic literature used is [1].

Zahvala

Iskreno se zahvaljujem mentorju doc. dr. Vitu Vitrihu za pomoč, nasvete in potrpežljivost pri izdelavi zaključne naloge.

Rada bi se zahvalila tudi družini za izkazano podporo in spodbudo v času študija.

Kazalo vsebine

| | | |
|----------|--|-----------|
| 1 | Uvod | 1 |
| 2 | Bézierove krivulje | 3 |
| 2.1 | De Casteljauev algoritem | 3 |
| 2.2 | Bernsteinova oblika Bézierove krivulje | 4 |
| 2.3 | Lastnosti polinomskih Bézierovih krivulj | 7 |
| 2.4 | Lastnosti odvajanja | 9 |
| 3 | Bézierove krpe | 11 |
| 3.1 | Bilinearna interpolacija | 11 |
| 3.2 | De Casteljauev algoritem | 11 |
| 3.3 | Bernsteinova oblika Bézierovih krp | 13 |
| 3.4 | Lastnosti Bézierovih krp | 14 |
| 3.5 | Odvodi | 15 |
| 4 | Gordon-Coons krpe | 18 |
| 4.1 | Bilinearno ujemanje | 18 |
| 4.2 | Delno bikubično ujemanje | 22 |
| 4.3 | Bikubično ujemanje | 24 |
| 4.4 | Kompatibilnost | 24 |
| 4.5 | Gordonove krpe | 26 |
| 5 | Zaključek | 28 |
| 6 | Literatura | 29 |

Kazalo slik

| | | |
|----|---|----|
| 1 | Bézierova krivulja s kontrolnim poligonom iz primera 2.4, skonstruirana z de Casteljauevim algoritmom, in točka na krivulji pri parametru $t = 0.5$. | 5 |
| 2 | Primer hiperboličnega paraboloida. | 12 |
| 3 | Slika ploskve s kontrolnim poligonom iz primera 3.2 in točka na ploskvi pri parametrih $u = v = 0.5$ | 13 |
| 4 | Štiri krivulje in Coonsova krpa, ki jo definirajo. | 18 |
| 5 | Ploskev \mathbf{r} definirana s krivuljama \mathbf{c} in \mathbf{d} | 19 |
| 6 | Slike prikazuje ploskve \mathbf{r}_c , \mathbf{r}_d , \mathbf{r}_{cd} in Coonsovo krpo \mathbf{x} iz primera 4.3 . . . | 21 |
| 7 | Primer diskretne Coonsove krpe za $m = n = 4$ | 22 |
| 8 | Čeprav so robne krivulje, ki definirajo Coonsovi krpi, gladke, pa ploskev, sestavljena iz teh krp, ni gladka. | 23 |
| 9 | Kubični Hermiteovi polinomi na intervalu $[0, 1]$ | 23 |
| 10 | Potrebni podatki za interpolacijo. | 25 |
| 11 | Mreža krivulj, ki jih želimo interpolirati. | 26 |
| 12 | Gordonova krpa, ki interpolira mrežo krivulj na sliki 11. | 27 |

Kazalo prilog

A Priloga

B Priloga

C Priloga

D Priloga

Seznam kratic

| | |
|-----------------|--|
| <i>itd.</i> | in tako dalje |
| <i>npr.</i> | na primer |
| <i>oz.</i> | oziroma |
| <i>CAD</i> | Computer Aided Design |
| <i>CAGD</i> | Computer Aided Geometric Design |
| <i>SIGGRAPH</i> | Steven A. Coons Award for Outstanding Contributions to Computer Graphics |

1 Uvod

Z razvojem tehnologije in računalnikov se je v 50. letih prejšnjega stoletja pojavilo vprašanje, kako načrte na papirju izdelati s pomočjo računalniško vodenih strojev. Odgovor so odkrili raziskovalci v avtomobilski industriji. P. de Casteljaeu in P. Bézier sta v Franciji neodvisno razvila zapis parametričnih krivulj, ki temelji na Bernsteinovih polinomih. Čeprav je de Casteljaeu prišel do odkritja prvi (leta 1958), je bilo le-to objavljeno za Bézierovim, zato so krivulje poimenovane po Bézieru. Prav tako izvirajo iz avtomobilске industrije Coonsove in Gordonove krpe, ki sta jih definirala S. A. Coons in W. Gordon. Coonsove ugotovitve so bile objavljene leta 1967, Gordonove pa leta 1974. Po Coonsu nosi ime tudi nagrada za izredne dosežke na področju računalniške grafike SIGGRAPH (Steven A. Coons Award for Outstanding Contributions to Computer Graphics). Uporabo parametričnih krivulj in ploskev v CAD (Computer Aided Design - oblikovanje s pomočjo računalnika) so poimenovali CAGD (Computer Aided Design - geometrijsko oblikovanje s pomočjo računalnika).

Bézierove krivulje so enolično določene s kontrolnimi točkami (glej npr. [1], [6], [7], [8]). Ta lastnost omogoča enostavno spreminjanje oblike in lažjo računalniško predstavitev. Bézierove krpe (oz. ploskve) so definirane kot tenzorski produkt krivulj (glej npr. [1], [6], [7]), saj je ploskev sled krivulje, ki se giblje vzdolž druge krivulje. Gordon-Coons krpe se od Bézierovih krp razlikujejo po tem, da niso definirane s štirikotno kontrolno mrežo, ampak s štirimi robnimi krivuljami (glej npr. [1], [2], [3], [4] [5]). Krivulje in ploskve danes uporabljajo v idustrijskem oblikovanju, računalniških pisavah in v zabavni industriji (filmi, računalniške igre, risanke).

Zaključna naloga je sestavljena iz treh poglavij:

- V naslednjem poglavju bomo definirali Bézierove krivulje, spoznali de Casteljaeuv algoritem, Bernsteinovo obliko Bézierovih krivulj, opisali njihove glavne lastnosti in lastnosti odvajanja.
- V tretjem poglavju bomo predstavili Bézierove krpe. Podobno kot v drugem poglavju jih bomo najprej definirali in nato opisali glavne lastnosti.
- V zadnjem poglavju se bomo osredotočili na Coonsove in Gordonove krpe. Pogledali si bomo njihovo definicijo ter njihovo uporabo.

V prilogah, dodanih na koncu zaključne naloge, so podane programske kode, s katerimi smo dobili nekatere rezultate v nalogi. Kode so napisane s programskim jezikom Octave, ki je prosto dostopen na <http://www.gnu.org/software/octave/>.

2 Bézierove krivulje

V tem poglavju bomo predstavili osnovne pojme povezane s polinomskimi Bézierovimi krivuljami, ki sta jih neodvisno drug od drugega odkrila francoska matematika de Casteljaeu in Bézier.

Najprej definirajmo parametrične krivulje, saj so Bézierove krivulje poseben primer le-teh.

Definicija 2.1. Preslikavi $\mathbf{p}(t) = (p_1(t), p_2(t), p_3(t)) : I \mapsto \mathbb{R}^3, I \subseteq \mathbb{R}$, pravimo parametrizacija krivulje $p(I)$. Sliki tej preslikave rečemo **parametrična krivulja**, intervalu I pa domena parametra.

Pri definiciji Bézierovih krivulj bomo potrebovali tudi definicijo prostorske linearne interpolacije.

Definicija 2.2. Naj bosta \mathbf{a} in \mathbf{b} dve različni točki v \mathbb{R}^3 . Množico točk $\mathbf{x} \in \mathbb{R}^3$ oblike

$$\mathbf{x} = \mathbf{x}(t) = (1 - t) \mathbf{a} + t \mathbf{b}, \quad t \in \mathbb{R},$$

imenujemo premica skozi točki \mathbf{a} in \mathbf{b} . Za $t = 0$ velja $\mathbf{x} = \mathbf{a}$ in za $t = 1$ velja $\mathbf{x} = \mathbf{b}$. Če $0 < t < 1$, potem točka \mathbf{x} leži na daljici med \mathbf{a} in \mathbf{b} . Če $t \notin [0, 1]$, potem točka \mathbf{x} leži na premici izven daljice \mathbf{a} in \mathbf{b} .

Definicija 2.3. Prostorska linearna interpolacija je afina preslikava, ki preslika realno premico skozi dani točki v tridimenzionalnem prostoru.

2.1 De Casteljauev algoritem

Veliko izračunov v CAGD lahko razbijemo na več manjših korakov - zaporedja linearnih interpolacij. Tako zaporedje opisuje tudi de Casteljauev algoritem, ki je eden izmed najpomembnejših na področju oblikovanja krivulj in ploskev.

De Casteljauev algoritem:

Naj bodo $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{R}^3$ kontrolne točke, ki določajo polinomske Bézierove krivulje \mathbf{b}^n in $t \in \mathbb{R}$ parameter. Kontrolne točke tvorijo kontrolni poligon Bézierove krivulje.

Definirajmo $\mathbf{b}_i^0(t) := \mathbf{b}_i$ in izračunajmo točke

$$\mathbf{b}_i^r(t) := (1 - t)\mathbf{b}_i^{r-1}(t) + t\mathbf{b}_{i+1}^{r-1}(t), \quad i = 0, 1, \dots, n - r, \quad r = 1, 2, \dots, n. \quad (2.1)$$

Točka $\mathbf{b}_0^n(t)$ pri parametru t predstavlja točko na Bézierovi krivulji \mathbf{b}^n . Vmesne točke lahko zapišemo v trikotno shemo, ki ji rečemo **de Casteljaueva shema**:

$$\begin{array}{ccccccc} \mathbf{b}_0^0 & = & \mathbf{b}_0 & & & & \\ \mathbf{b}_1^0 & = & \mathbf{b}_1 & \mathbf{b}_0^1(t) & & & \\ \mathbf{b}_2^0 & = & \mathbf{b}_2 & \mathbf{b}_1^1(t) & \ddots & & \\ & \vdots & & \vdots & & \ddots & \\ & & & & & \mathbf{b}_0^{n-1}(t) & \\ \mathbf{b}_n^0 & = & \mathbf{b}_n & \mathbf{b}_{n-1}^1(t) & \mathbf{b}_1^{n-1}(t) & \mathbf{b}_0^n(t) & \end{array}$$

De Casteljauevo shemo sestavimo tako, da v prvi stolpec zapišemo vse kontrolne točke, v drugi stolpec točke za $r = 1$, v tretji stolpec točke za $r = 2$, itd. Kot že omenjeno, v zadnjem stolpcu dobimo točko $\mathbf{b}_0^n(t)$, ki predstavlja točko na krivulji \mathbf{b}^n za parameter t . Primer kontrolnega polgona, ki ga tvorijo točke izračunane z de Casteljauevim algoritmom, je prikazan na sliki 1.

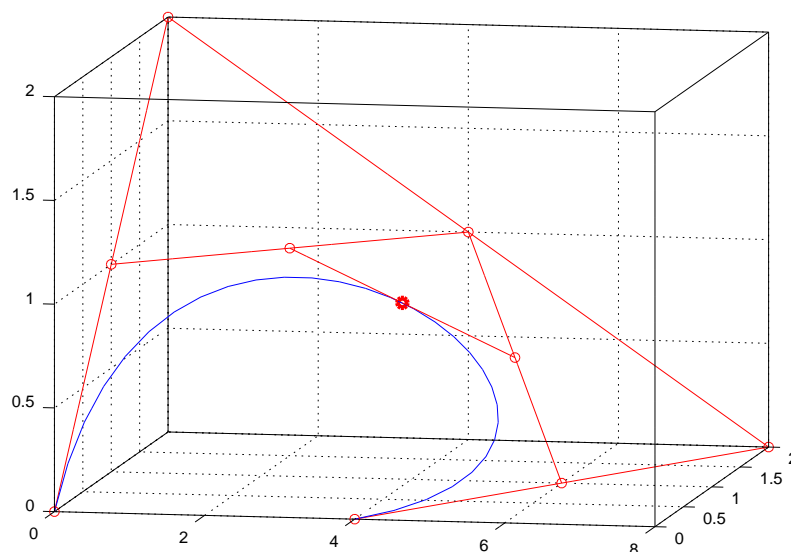
Primer 2.4. Naj bodo $\mathbf{b}_0 = (0, 0, 0)^T$, $\mathbf{b}_1 = (0, 2, 2)^T$, $\mathbf{b}_2 = (8, 2, 0)^T$ in $\mathbf{b}_3 = (4, 0, 0)^T$. Izračunajmo točko na Bézierovi krivulji pri parametru $t = 0.5$. Dobimo de Caustelja-uovo shemo oblike:

$$\begin{array}{ccccccc} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & & & & & & \\ & \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix} & & & & & \\ & & \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} & & & & \\ & & & \begin{bmatrix} 2 \\ 1.5 \\ 1 \end{bmatrix} & & & \\ & & & & \begin{bmatrix} 5 \\ 1.5 \\ 0.5 \end{bmatrix} & & \\ & & & & & \begin{bmatrix} 3.5 \\ 1.5 \\ 0.75 \end{bmatrix} & \end{array}$$

Dobljena krivulja, ko t preteče $[0, 1]$, je prikazana na sliki 1.

2.2 Bernsteinova oblika Bézierove krivulje

Bernsteinova oblika Bézierove krivulje je drug način zapisa te krivulje. Prednost tega zapisa je v tem, da nam poda formulo za Bézierovo krivuljo v zaključeni obliki. Ta



Slika 1: Bézierova krivulja s kontrolnim poligonom iz primera 2.4, skonstruirana z de Casteljaujevim algoritmom, in točka na krivulji pri parametru $t = 0.5$.

lastnost nam olajša nadaljni opis lastnosti krivulj.

Definicija 2.5. Bernsteinovi polinomi so polinomi oblike

$$B_i^n(t) := \binom{n}{i} t^i (1-t)^{n-i}, \quad t \in [0, 1],$$

kjer je binomski koeficient definiran kot

$$\binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!}, & 0 \leq i \leq n \\ 0, & \text{sicer.} \end{cases}$$

Spodaj so opisane nekatere glavne lastnosti Bernsteinovih polinomov, ki nam bodo v pomoč pri dokazovanju lastnosti Bézierovih krivulj.

Izrek 2.6. *Bernsteinovi polinomi zadoščajo naslednji rekurzivni zvezi:*

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

kjer velja $B_0^0(t) \equiv 1$, $B_i^n(t) \equiv 0$, $i \notin \{0, 1, \dots, n\}$.

Dokaz. Vemo, da velja $\binom{n-1}{i} + \binom{n-1}{i-1} = \binom{n}{i}$. Potem velja tudi naslednje:

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

□

Izrek 2.7. *Bernsteinovi polinomi tvorijo particijo enote:*

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

Dokaz. Pri dokazu uporabimo binomski izrek:

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

□

Izrek 2.8. *Bernsteinovi polinomi so simetrični na intervalu $[0,1]$:*

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

Dokaz.

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

□

Izrek 2.9. *Naj bodo $\mathbf{b}_i \in \mathbb{R}^3, i = 0, 1, \dots, n$, kontrolne točke Bézierove krivulje \mathbf{b}^n . Potem velja:*

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

Dokaz. Dovolj je pokazati, da se elementi k -tega stolpca v de Casteljauevi shemi izražajo kot

$$\mathbf{b}_i^k(t) = \sum_{\ell=i}^{i+k} \mathbf{b}_\ell B_{\ell-i}^k(t), \quad i = 0, 1, \dots, n - k. \quad (2.2)$$

Če to pokažemo, potem to velja za vse stolpce v de Casteljauevi shemi, torej tudi za n -ti stolpec. Vstavimo v enačbo (2.2) $k = n$ in $i = 0$ in dobimo:

$$\mathbf{b}^n(t) = \mathbf{b}_0^n(t) = \sum_{\ell=0}^n \mathbf{b}_\ell B_\ell^n(t).$$

Za dokaz enačbe (2.2) uporabimo indukcijo po k . Za $k = 0$ dobimo

$$\mathbf{b}_i^0(t) = \sum_{\ell=i}^i \mathbf{b}_\ell B_{\ell-i}^0(t) = \mathbf{b}_i B_0^0(t) = \mathbf{b}_i.$$

Preverimo še primer, ko $k \rightarrow k + 1$, da velja:

$$\mathbf{b}_i^{k+1}(t) = \sum_{\ell=i}^{i+k+1} \mathbf{b}_\ell B_{\ell-i}^{k+1}(t).$$

Upoštevamo prej dokazane lastnosti Bernsteinovih polinomov in dobimo:

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

□

2.3 Lastnosti polinomskih Bézierovih krivulj

Pokažimo nekaj lastnosti polinomskih Bézierovih krivulj, zaradi katerih so te tako uporabne.

1. Afina invariantnost.

De Casteljaujev algoritem je zaporedje linearnih interpolacij. Potem afina invariantnost za Bézierove krivulje velja, saj lahko poljubno točko Bézierove krivulje dobimo kot končno zaporedje linearnih interpolacij. Naj bo ϕ afina preslikava. Tedaj sta naslednja postopka ekvivalentna:

- izračunamo točko na Bézierovi krivulji \mathbf{b}^n in jih afino preslikamo s preslikavo ϕ v krivuljo $\phi(\mathbf{b}^n)$;
- afino preslikavo uporabimo samo na kontrolnih točkah. Na dobljenih kontrolnih točkah $\phi(\mathbf{b}_0), \phi(\mathbf{b}_1), \dots, \phi(\mathbf{b}_n)$ izračunamo točke na Bézierovi krivulji.

2. Invarianca za afine transformacije parametra.

V definiciji 2.2 smo za definicijsko območje uporabili interval $[0, 1]$, vendar lahko Bézierovo krivuljo definiramo tudi nad poljubnim intervalom $[a, b]$. Za izračun krivulje vpeljemo lokalno parametrizacijo $t = (u - a)/(b - a)$ in uporabimo de Casteljaujev algoritem

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

Velja tudi:

$$\sum_{i=0}^n \mathbf{b}_i B_i^n(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n\left(\frac{u-a}{b-a}\right), \quad t \in [0, 1], \quad u \in [a, b].$$

3. Konveksna ovojnica.

Definicija 2.10. Množica K je **konveksna** \iff vsak $\mathbf{x}, \mathbf{y} \in K$ velja

$$(1-t)\mathbf{x} + t\mathbf{y} \in K, \quad t \in [0, 1].$$

Definicija 2.11. **Konveksna ovojnica** točk $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ je najmanjša konveksna množica, ki vsebuje te točke.

Izrek 2.12. *Bézierova krivulja \mathbf{b}^n leži v konveksni ovojnici svojih kontrolnih točk.*

Dokaz. Vsaka točka $\mathbf{b}_i^r(t)$ iz de Casteljauevega algoritma je konveksna kombinacija točk iz predhodnjih stolpcev, zato leži na daljici med dvema že obstoječima točkama, ki sta v konveksni ovojnici. Torej leži tudi $\mathbf{b}_i^r(t)$ v konveksni ovojnici svojih kontrolnih točk. Podobno velja tudi za zadnjo točko v de Casteljauevi shemi. Ker je to točka na Bézierovi krivulji, leži celotna krivulja \mathbf{b}^n v konveksni ovojnici svojih kontrolnih točk. \square

Opomba: Lastnost velja le za $t \in [0, 1]$.

4. Interpolacija robnih točk.

Bézierova krivulja stopnje n interpolira kontrolni točki \mathbf{b}_0 in \mathbf{b}_n . Enakosti $\mathbf{b}^n(0) = \mathbf{b}_0$ in $\mathbf{b}^n(1) = \mathbf{b}_n$ sledita iz:

$$B_i^n(0) = \delta_{i,0}, \quad B_i^n(1) = \delta_{i,n},$$

kjer je

$$\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Kroneckerjeva delta funkcija.

5. Simetrija.

Opazimo, da lahko točke označimo z $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ ali $\mathbf{b}_n, \mathbf{b}_{n-1}, \dots, \mathbf{b}_0$, saj vrstni red točk vpliva le na smer parametrizacije Bézierove krivulje. To lahko zapišemo s formulo:

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

Formula sledi iz lastnosti Bernsteinovih polinomov:

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

6. Invarianca za baricentrične kombinacije.

Uteženo povprečje dveh Bézierovih krivulj lahko izračunamo z uteženim povprečjem ustreznih točk na krivuljah, ali pa z uteženim povprečjem ustreznih kontrolnih točk in izračunom krivulje na teh kontrolnih točkah. Za $\alpha + \beta = 1$ dobimo:

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

7. Psevdo - lokalni nadzor

Najprej izračunamo stacionarne točke Bernsteinovega polinoma B_i^n . V naslednjem podpoglavju bomo pokazali, da velja:

$$\frac{d}{dt} B_i^n(t) = n \cdot (B_{i-1}^{n-1}(t) - B_i^{n-1}(t)).$$

Če to za trenutek privzamemo, potem lahko izpeljemo naslednje ekvivalence:

$$\begin{aligned} \frac{d}{dt} B_i^n(t) = 0 &\iff n \cdot (B_{i-1}^{n-1}(t) - B_i^{n-1}(t)) \\ &\iff B_{i-1}^{n-1}(t) = B_i^{n-1}(t) \\ &\iff \binom{n-1}{i-1} t^{i-1} (1-t)^{n-i} = \binom{n-1}{i} t^i (1-t)^{n-1-i} \\ &\iff t^{-1} (1-t) = \frac{\binom{n-1}{i}}{\binom{n-1}{i-1}} \\ &\iff \frac{1-t}{t} = \frac{n-i}{i} \\ &\iff t = \left(\frac{n}{i}\right)^{-1} = \frac{i}{n}. \end{aligned}$$

Iz $\frac{d^2}{dt^2} B_i^n(t) \Big|_{t=\frac{i}{n}} < 0$ sledi, da doseže polinom B_i^n lokalni maksimum v točki $t = \frac{i}{n}$ na intervalu $[0, 1]$. Ta lastnost vpliva na oblikovanje Bézierovih krivulj. Premik ene izmed kontrolnih točk \mathbf{b}_i spremeni obliko krivulje najbolj v okolici točke pri parametru $\frac{i}{n}$. Vendar pa sprememba vpliva na celotno krivuljo, zato je lastnost poimenovana psevdo-lokalni nadzor. Za lokalni nadzor celotne krivulje potrebujemo posplošitev Bézierovih krivulj na B-zlepke.

2.4 Lastnosti odvajanja

Odvod Bézierove krivulje izračunamo s pomočjo odvajanja Bernsteinovih polinomov:

$$\frac{d}{dt} \mathbf{b}^n(t) = \frac{d}{dt} \sum_{i=0}^n \mathbf{b}_i B_i^n(t) = \sum_{i=0}^n \mathbf{b}_i \frac{d}{dt} B_i^n(t). \quad (2.3)$$

Izračunajmo torej najprej odvod Bernsteinovega polinoma B_i^n :

$$\begin{aligned}
 \frac{d}{dt}B_i^n(t) &= \frac{d}{dt} \binom{n}{i} t^i (1-t)^{n-i} \\
 &= \binom{n}{i} (i t^{i-1} (1-t)^{n-i} - (n-i) t^i (1-t)^{n-i-1}) \\
 &= \frac{n!}{i!(n-i)!} (i t^{i-1} (1-t)^{n-i} - (n-i) t^i (1-t)^{n-i-1}) \\
 &= \frac{n!}{(i-1)!(n-i)!} t^{i-1} (1-t)^{n-i} - \frac{n!}{i!(n-i-1)!} t^i (1-t)^{n-i-1} \\
 &= n \cdot \frac{(n-1)!}{(i-1)!(n-i)!} t^{i-1} (1-t)^{n-i} - n \cdot \frac{(n-1)!}{i!(n-i-1)!} t^i (1-t)^{n-i-1} \\
 &= n \cdot \left(\binom{n-1}{i-1} t^{i-1} (1-t)^{n-i} - \binom{n-1}{i} t^i (1-t)^{n-i-1} \right) \\
 &= n \cdot (B_{i-1}^{n-1}(t) - B_i^{n-1}(t)). \tag{2.4}
 \end{aligned}$$

Vstavimo sedaj v enačbo (2.3) in izračunamo odvod Bézierove krivulje:

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

Uvedemo definicijo premih končnih diferenc

$$\Delta \mathbf{b}_i := \mathbf{b}_{i+1} - \mathbf{b}_i$$

in dobimo:

$$\frac{d}{dt} \mathbf{b}^n(t) = n \sum_{i=0}^{n-1} \Delta \mathbf{b}_i B_i^{n-1}(t). \tag{2.5}$$

Odvod Bézierove krivulje je torej Bézierova krivulja, katere kontrolni poligon je sestavljen iz diferenc kontrolnih točk začetnega kontrolnega poligona.

Če zgornji postopek večkrat ponovimo, dobimo višje odvode Bézierove krivulje:

$$\frac{d^r}{dt^r} \mathbf{b}^n(t) = \frac{n!}{(n-r)!} \sum_{i=0}^{n-r} \Delta^r \mathbf{b}_i B_i^{n-r}(t),$$

kjer je

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

3 Bézierove krpe

V tem poglavju si bomo ogledali Bézierove krpe. Za lažjo primerjavo zapisov in lastnosti krivulj ter krp, so podpoglavja podobna kot v prejšnjem poglavju.

Podobno kot pri krivuljah, najprej definirjamo parametrične ploskve.

Definicija 3.1. Ploskev \mathbf{x} zapišemo v parametrični obliki kot

$$\mathbf{x} = \mathbf{x}(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix}, \quad u \in [a, b] \subset \mathbb{R}^2, v \in [c, d] \subset \mathbb{R}^2.$$

Definicija nam pove, da je ploskev \mathbf{x} slika pravokotnika $[a, b] \times [c, d]$ v (u, v) ravnini. Temu pravokotniku pravimo domena parametrične ploskve in ima enak pomen za Bézierove ploskve, kot ga ima interval $[a, b]$ pri Bézierovih krivuljah.

3.1 Bilinearna interpolacija

Linearno interpolacijo pri krivuljah iz prejšnjega poglavja prenesemo na ploskve. Naj bodo $\mathbf{b}_{0,0}, \mathbf{b}_{0,1}, \mathbf{b}_{1,0}, \mathbf{b}_{1,1}$ štiri različne točke v \mathbb{R}^3 . Točke $\mathbf{x} \in \mathbb{R}^3$, oblike

$$\mathbf{x}(u, v) = \sum_{i=0}^1 \sum_{j=0}^1 \mathbf{b}_{i,j} B_i^1(u) B_j^1(v), \quad (3.1)$$

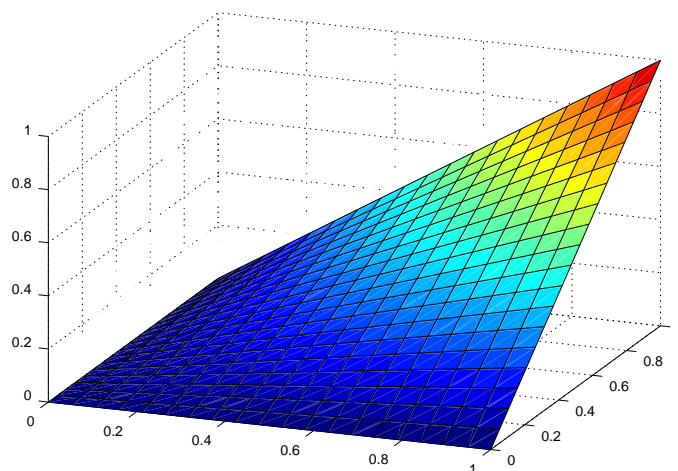
tvorijo **hiperbolični paraboloid** skozi dane začetne točke (glej sliko 2). Vzemimo ploskev $z = xy$. Če ploskev presečemo z ravnino, ki je vzporedna ravnini x, y , dobimo hiperbolo. Če ploskev presečemo z ravnino, ki vsebuje os z , dobimo parabolo.

Enačbo (3.1) lahko zapišemo tudi v matrični obliki:

$$\mathbf{x}(u, v) = \begin{bmatrix} 1 - u & u \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,0} & \mathbf{b}_{0,1} \\ \mathbf{b}_{1,0} & \mathbf{b}_{1,1} \end{bmatrix} \begin{bmatrix} 1 - v \\ v \end{bmatrix}. \quad (3.2)$$

3.2 De Casteljaujev algoritem

Podobno kot smo dobili Bézierove krivulje z zaporedno uporabo linearne interpolacije, dobimo Bézierove ploskve oz. krpe z zaporedno uporabo bilinearne interpolacije. Naj



Slika 2: Primer hiperboličnega paraboloida.

točke $\mathbf{b}_{i,j}, 0 \leq i, j \leq n$, sestavljajo pravokotno mrežo točk in naj bosta $(u, v) \in \mathbb{R}^2$ dana parametra. Definirajmo $\mathbf{b}_{i,j}^{0,0} := \mathbf{b}_{i,j}$. Potem je de Casteljaujev algoritem oblike:

$$\mathbf{b}_{i,j}^{p,r}(u, v) = \begin{bmatrix} 1-u & u \end{bmatrix} \begin{bmatrix} \mathbf{b}_{i,j}^{p-1,r-1} & \mathbf{b}_{i,j+1}^{p-1,r-1} \\ \mathbf{b}_{i+1,j}^{p-1,r-1} & \mathbf{b}_{i+1,j+1}^{p-1,r-1} \end{bmatrix} \begin{bmatrix} 1-v \\ v \end{bmatrix},$$

kjer

$$i = 0, 1, \dots, m-p, \quad j = 0, 1, \dots, n-r \quad p = 1, 2, \dots, m, \quad r = 1, 2, \dots, n.$$

Točka $\mathbf{b}_{0,0}^{m,n}(u, v)$, ki jo dobimo v zadnjem koraku pri vsakem paru parametrov (u, v) , je točka, ki leži na Bézierovi ploskvi $\mathbf{b}^{m,n}$ za dana parametra u in v . Točke $\mathbf{b}_{i,j}$ imenujemo **kontrolne točke**, mrežo teh točk pa **kontrolna mreža**.

Primer 3.2. Naj bo Bézierova ploskev definirana z naslednjo Bézierovo mrežo:

$$\begin{bmatrix} \mathbf{b}_{0,0} \\ \mathbf{b}_{1,0} \\ \mathbf{b}_{2,0} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,1} \\ \mathbf{b}_{1,1} \\ \mathbf{b}_{2,1} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,2} \\ \mathbf{b}_{1,2} \\ \mathbf{b}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \\ 0 \\ 0 \\ 4 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 4 \\ 4 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ 4 \\ 2 \\ 2 \\ 4 \\ 4 \\ 4 \\ 2 \\ 1 \end{bmatrix}.$$

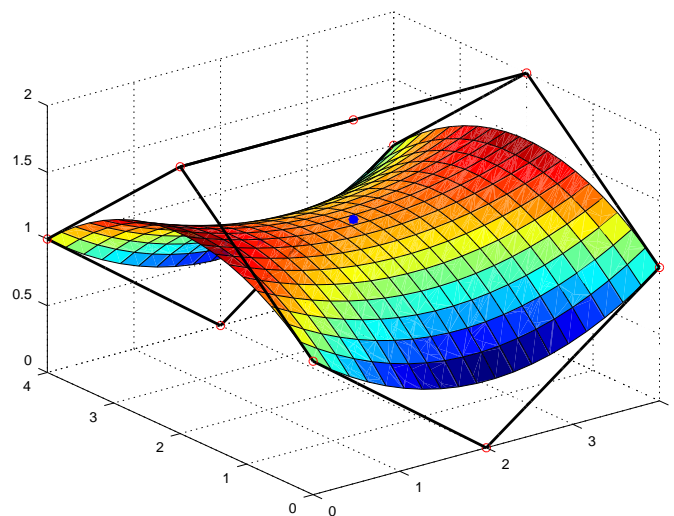
Za parametra $u = 0.5$ in $v = 0.5$ dobimo:

$$\begin{bmatrix} \mathbf{b}_{0,0}^{1,1} \\ \mathbf{b}_{1,0}^{1,1} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,1}^{1,1} \\ \mathbf{b}_{1,1}^{1,1} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1.25 \\ 3 \\ 1 \\ 1.25 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 1.25 \\ 3 \\ 3 \\ 1.25 \end{bmatrix}.$$

V naslednjem koraku dobimo točko:

$$\begin{bmatrix} \mathbf{b}_{0,0}^{2,2} \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 1.125 \end{bmatrix}.$$

Ploskev, ko u in v pretečeta $[0, 1] \times [0, 1]$, je prikazana na sliki 3.



Slika 3: Slika ploskve s kontrolnim poligonom iz primera 3.2 in točka na ploskvi pri parametrih $u = v = 0.5$.

3.3 Bernsteinova oblika Bézierovih krp

Bézierove krpe lahko, tako kot krivulje, zapišemo tudi z Bernsteinovimi polinomi. Zato uvedemo v tem poglavju pojem tenzorskega produkta. Na podlagi slednjega dobimo ploskev tako, da dano začetno krivuljo premikamo vzdolž dveh krivulj. Ploskev bo torej sled začetne krivulje v prostoru. Naj bo krivulja, ki jo premikamo po prostoru, Bézierova krivulja stopnje m z $m + 1$ kontrolnimi točkami. Predpostavimo tudi, da

sta krivulji, po katerih se gibljejo kontrolne točke, Bézierovi krivulji stopnje n z $n + 1$ kontrolnimi točkami. Potem lahko osnovno krivuljo zapišemo kot:

$$\mathbf{b}^m(u) = \sum_{i=0}^m \mathbf{b}_i B_i^m(u), \quad (3.3)$$

kontrolne točke \mathbf{b}_i pa se premikajo po Bézierovi krivulji stopnje n :

$$\mathbf{b}_i(v) = \sum_{j=0}^n \mathbf{b}_{i,j} B_j^n(v). \quad (3.4)$$

Točko $\mathbf{b}^{m,n}(u, v)$ najdemo na ploskvi $\mathbf{b}^{m,n}$ tako, da združimo enačbi (3.3) in (3.4):

$$\mathbf{b}^{m,n}(u, v) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} B_i^m(u) B_j^n(v). \quad (3.5)$$

3.4 Lastnosti Bézierovih krp

Večina lastnosti Bézierovih krp sledi iz lastnosti Bézierovih krivulj, ki smo jih že pokazali.

1. Bézierova krpa interpolira kontrolne točke $\mathbf{b}_{0,0}$, $\mathbf{b}_{0,n}$, $\mathbf{b}_{m,0}$ in $\mathbf{b}_{m,n}$.

Dokaz. Izberemo pare parametrov (u, v) , ki predstavljajo oglišča pravokotne domene ploskve $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$. Uporabimo (3.5):

$$\mathbf{b}^{m,n}(0, 0) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} B_i^m(0) B_j^n(0) = \mathbf{b}_{0,0},$$

saj so vsi Bernsteinovi polinomi enaki nič, razen pri $i = j = 0$. Tako dobimo:

$$\mathbf{b}_{0,0} \cdot \binom{m}{0} 0^0 1^m \cdot \binom{n}{0} 0^0 1^n = \mathbf{b}_{0,0}.$$

Podobno dobimo še ostale točke:

$$\mathbf{b}^{m,n}(0, 1) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} B_i^m(0) B_j^n(1) = \mathbf{b}_{0,n},$$

$$\mathbf{b}^{m,n}(1, 0) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} B_i^m(1) B_j^n(0) = \mathbf{b}_{m,0},$$

$$\mathbf{b}^{m,n}(1, 1) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} B_i^m(1) B_j^n(1) = \mathbf{b}_{m,n}.$$

□

2. Naj bosta $B_i^m(u)$ in $B_j^n(v)$ Bernsteinova polinoma. Potem velja:

$$\sum_{i=0}^m \sum_{j=0}^n B_i^m(u) B_j^n(v) = 1.$$

Dokaz. Sledi iz izreka (2.7):

$$\sum_{i=0}^m \sum_{j=0}^n B_i^m(u) B_j^n(v) = \sum_{i=0}^m B_i^m(u) \sum_{j=0}^n B_j^n(v) = 1 \cdot 1 = 1.$$

□

3. Bézierova krpa leži znotraj konveksne ovojnice svoje kontrolne mreže.

Dokaz. Naj velja $0 \leq u, v \leq 1$, potem je produkt $B_i^m(u) \cdot B_j^n(v)$ nenegativen. Torej bo (3.5) konveksna kombinacija kontrolnih točk in bo zato Bézierova krpa ležala znotraj konveksne ovojnice svoje kontrolne mreže. □

4. Bézierove krpe so invariantne za afine preslikave.

Dokaz. De Casteljaujev algoritem je ponavljanje bilinearnih interpolacij. Upoštevamo, da je bilinearna interpolacija pravzaprav linearna interpolacija v smereh u in v . Iz prejšnjega poglavja vemo, da je linearna interpolacija invariantna za afine preslikave, torej je tudi bilinearna interpolacija afino invariantna. Sledi, da so Bézierove krpe invariantne za afine preslikave. □

5. Robne krivulje Bézierovih krp so Bézierove krivulje.

Dokaz. Robne krivulje Bézierovih ploskev dobimo tako, da izberemo parametre $u = 0$, $v = 0$, $u = 1$ ali $v = 1$. Vzemimo npr. $u = 0$. Potem velja:

$$\mathbf{b}^{m,n}(0, v) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} B_i^m(0) B_j^n(v) = \sum_{j=0}^n \mathbf{b}_{0,j} B_j^n(v),$$

kar je očitno Bézierova krivulja s kontrolnimi točkami $\mathbf{b}_{00}, \mathbf{b}_{01}, \dots, \mathbf{b}_{0n}$. □

3.5 Odvodi

Pri Bézierovih krpah uporabljamo za izračun točk bilinearno interpolacijo, zato moramo pri odvajanju upoštevati odvod v smeri parametra u in v smeri parametra v .

Trditev 3.3. Naj bo $\mathbf{b}^{m,n}$ Bézierova ploskev stopnje (m, n) . Potem velja:

$$\frac{\partial}{\partial u} \mathbf{b}^{m,n}(u, v) = m \cdot \sum_{i=0}^{m-1} \sum_{j=0}^n \Delta^{(1,0)} \mathbf{b}_{i,j} B_i^{m-1}(u) B_j^n(v)$$

$$\frac{\partial}{\partial v} \mathbf{b}^{m,n}(u, v) = n \cdot \sum_{i=0}^m \sum_{j=0}^{n-1} \Delta^{(0,1)} \mathbf{b}_{i,j} B_i^m(u) B_j^{n-1}(v)$$

$$\frac{\partial^2}{\partial u \partial v} \mathbf{b}^{m,n}(u, v) = m \cdot n \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \Delta^{(1,1)} \mathbf{b}_{i,j} B_i^{m-1}(u) B_j^{n-1}(v),$$

kjer so

$$\Delta^{(1,0)} := \mathbf{b}_{i+1,j} - \mathbf{b}_{i,j}, \quad \Delta^{(0,1)} := \mathbf{b}_{i,j+1} - \mathbf{b}_{i,j}, \quad \Delta^{(1,1)} := \mathbf{b}_{i+1,j+1} - \mathbf{b}_{i,j+1} - \mathbf{b}_{i+1,j} + \mathbf{b}_{i,j}$$

Dokaz. Odvajajmo enačbo (3.5) po spremenljivki u in dobimo:

$$\frac{\partial}{\partial u} \mathbf{b}^{m,n}(u, v) = \sum_{j=0}^n \left(\frac{\partial}{\partial u} \sum_{i=0}^m \mathbf{b}_{i,j} B_i^m(u) \right) B_j^n(v).$$

Upoštevamo (2.5) in dobimo:

$$\begin{aligned} \frac{\partial}{\partial u} \mathbf{b}^{m,n}(u, v) &= \sum_{j=0}^n \left(m \cdot \sum_{i=0}^{m-1} (\mathbf{b}_{i+1,j} - \mathbf{b}_{i,j}) B_i^{m-1}(u) \right) B_j^n(v) \\ &= m \cdot \sum_{j=0}^n \sum_{i=0}^{m-1} \Delta^{(1,0)} \mathbf{b}_{i,j} B_i^{m-1}(u) B_j^n(v). \end{aligned}$$

Podobno odvajajmo po parametru v :

$$\begin{aligned} \frac{\partial}{\partial v} \mathbf{b}^{m,n}(u, v) &= \sum_{i=0}^m \left(\frac{\partial}{\partial v} \sum_{j=0}^n \mathbf{b}_{i,j} B_j^n(v) \right) B_i^m(u) \\ &= \sum_{i=0}^m \left(n \cdot \sum_{j=0}^{n-1} (\mathbf{b}_{i,j+1} - \mathbf{b}_{i,j}) B_j^{n-1}(v) \right) B_i^m(u) \\ &= n \cdot \sum_{i=0}^m \sum_{j=0}^{n-1} \Delta^{(0,1)} \mathbf{b}_{i,j} B_i^m(u) B_j^{n-1}(v). \end{aligned}$$

Sedaj lahko izračunamo še mešani odvod.

$$\begin{aligned} \frac{\partial^2}{\partial u \partial v} \mathbf{b}^{m,n}(u, v) &= \frac{\partial}{\partial u} \left(\frac{\partial}{\partial v} \mathbf{b}^{m,n}(u, v) \right) \\ &= \frac{\partial}{\partial u} \left(n \cdot \sum_{i=0}^m \sum_{j=0}^{n-1} \Delta^{(0,1)} \mathbf{b}_{i,j} B_i^m(u) B_j^{n-1}(v) \right) \end{aligned}$$

$$\begin{aligned}
 &= n \cdot \sum_{j=0}^{n-1} B_j^{n-1}(v) \frac{\partial}{\partial u} \left(\sum_{i=0}^m (\mathbf{b}_{i,j+1} - \mathbf{b}_{i,j}) B_i^m(u) \right) \\
 &= n \cdot \sum_{j=0}^{n-1} B_j^{n-1}(v) \frac{\partial}{\partial u} \left(m \cdot \sum_{i=0}^{m-1} ((\mathbf{b}_{i+1,j+1} - \mathbf{b}_{i,j+1}) - (\mathbf{b}_{i+1,j} - \mathbf{b}_{i,j})) B_i^{m-1}(u) \right).
 \end{aligned}$$

□

Podobno kot v prejšnjem poglavju definirajmo še parcialne odvode višjih stopenj. Uporabimo (2.6) in dobimo:

$$\frac{\partial}{\partial u^r} \mathbf{b}^{m,n}(u, v) = \frac{m!}{(m-r)!} \sum_{i=0}^{m-r} \sum_{j=0}^n \Delta^{(r,0)} \mathbf{b}_{i,j} B_i^{m-r}(u) B_j^n(v),$$

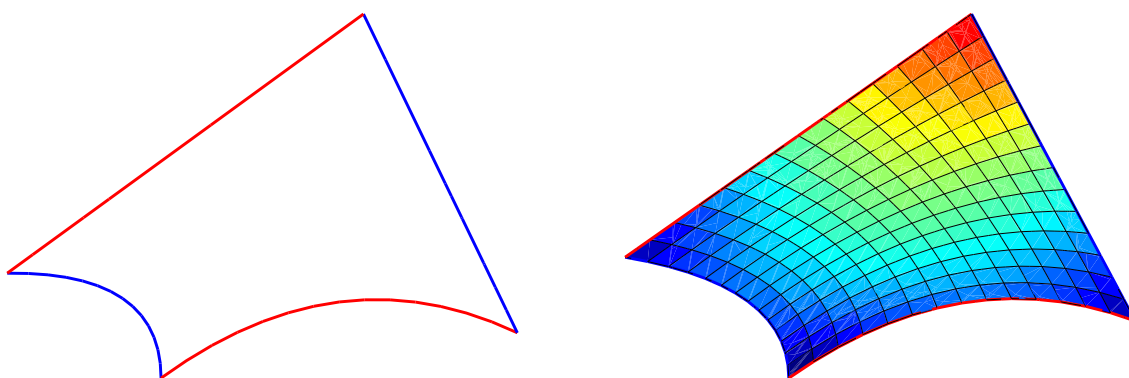
$$\frac{\partial}{\partial v^s} \mathbf{b}^{m,n}(u, v) = \frac{n!}{(n-s)!} \sum_{i=0}^m \sum_{j=0}^{n-s} \Delta^{(0,s)} \mathbf{b}_{i,j} B_i^m(u) B_j^{n-s}(v),$$

kjer je

$$\Delta^{(r,0)} := \Delta^{(r-1,0)} \mathbf{b}_{i+1,j} - \Delta^{(r-1,0)} \mathbf{b}_{i,j}, \quad \Delta^{(0,s)} := \Delta^{(0,s-1)} \mathbf{b}_{i,j+1} - \Delta^{(0,s-1)} \mathbf{b}_{i,j}.$$

4 Gordon-Coons krpe

Poleg Bézierovih krp, ki smo jih opisali v prejšnjem poglavju, poznamo tudi Gordon-Coons krpe. Slednje niso definirane preko kontrolnih točk, ki ležijo na kontrolni mreži, ampak s štirimi krivuljami, ki se sekajo v štirih ogliščih (glej sliko 4).



Slika 4: Štiri krivulje in Coonsova krpa, ki jo definirajo.

4.1 Bilinearno ujemanje

Pojem bilinearno ujemanje izhaja iz konstrukcije Coonsovih krp.

Lema 4.1. *Naj bosta \mathbf{c} in \mathbf{d} polinomski Bézierovi krivulji podani z njunima kontrolnima poligonoma $\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n$ in $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_n$. Potem ploskev definirana kot:*

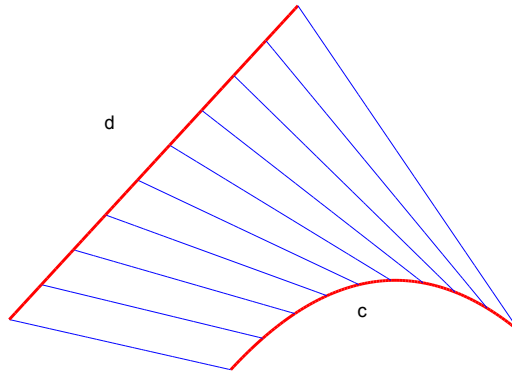
$$\mathbf{r}(u, v) = (1 - v) \mathbf{c}(u) + v \mathbf{d}(u)$$

povezuje krivulji \mathbf{c} in \mathbf{d} .

Dokaz. Ploskev $\mathbf{r}(u, v)$ pri $v = 0$ interpolira $\mathbf{c}(u)$, pri $v = 1$ pa interpolira $\mathbf{d}(u)$. \square

Vzdolž parametra v je ploskev linearna, vzdolž parametra u pa stopnje n . Ploskev \mathbf{r} lahko zapišemo tudi kot:

$$\mathbf{r}(u, v) = \begin{bmatrix} 1 - v, & v \end{bmatrix} \begin{bmatrix} \mathbf{c}_0 & \mathbf{c}_1 & \dots & \mathbf{c}_n \\ \mathbf{d}_0 & \mathbf{d}_1 & \dots & \mathbf{d}_n \end{bmatrix} \begin{bmatrix} B_0^n(u) \\ \dots \\ B_n^n(u) \end{bmatrix}.$$



Slika 5: Ploskev \mathbf{r} definirana s krivuljama \mathbf{c} in \mathbf{d} .

Primer tako definirane ploskve vidimo na sliki 5.

Naj bodo sedaj $\mathbf{c}_1, \mathbf{c}_2$ in $\mathbf{d}_1, \mathbf{d}_2$ štiri polinomske krivulje, definirane s parametroma $u \in [0, 1]$ in $v \in [0, 1]$. Preko teh krivulj definirajmo naslednji ploskvi:

$$\mathbf{r}_c(u, v) = (1 - v) \mathbf{c}_1(u) + v \mathbf{c}_2(u), \quad (4.1)$$

$$\mathbf{r}_d(u, v) = (1 - u) \mathbf{d}_1(v) + u \mathbf{d}_2(v). \quad (4.2)$$

Opazimo, da ploskev \mathbf{r}_c interpolira le krivulji \mathbf{c}_1 in \mathbf{c}_2 , ne pa tudi krivulj \mathbf{d}_1 in \mathbf{d}_2 . Podobno velja za ploskev \mathbf{r}_d . Da bi dobili končno interpolacijsko ploskev, želimo obdržati, kar ploskvi že interpolirata in hkrati odstraniti, kar ne interpolirata. Ta napaka je ravno bilinearna ploskev, definirana s štirimi oglišči, ki jo lahko po (3.2) zapišemo kot:

$$\mathbf{r}_{c,d}(u, v) = \begin{bmatrix} 1 - u, & u \end{bmatrix} \begin{bmatrix} \mathbf{x}(0, 0), & \mathbf{x}(0, 1) \\ \mathbf{x}(1, 0), & \mathbf{x}(1, 1) \end{bmatrix} \begin{bmatrix} 1 - v \\ v \end{bmatrix}. \quad (4.3)$$

Izrek 4.2. Naj bodo $\mathbf{r}_c, \mathbf{r}_d$ in $\mathbf{r}_{c,d}$ kot v (4.1), (4.2) in (4.3). **Coonsovo krpo** \mathbf{x} , ki interpolira robne krivulje $\mathbf{c}_1, \mathbf{c}_2, \mathbf{d}_1$ in \mathbf{d}_2 , dobimo kot:

$$\mathbf{x} = \mathbf{r}_c + \mathbf{r}_d - \mathbf{r}_{c,d}.$$

Dokaz.

$$\begin{aligned} \mathbf{x}(u, 0) &= \mathbf{r}_c(u, 0) + \mathbf{r}_d(u, 0) - \mathbf{r}_{c,d}(u, 0) \\ &= \mathbf{c}_1(u) + (1 - u) \mathbf{d}_1(0) + u \mathbf{d}_2(0) - ((1 - u) \mathbf{x}(0, 0) + u \mathbf{x}(1, 0)), \end{aligned}$$

kjer je $\mathbf{x}(0, 0) = \mathbf{d}_1(0)$ in $\mathbf{x}(1, 0) = \mathbf{d}_2(0)$. Sledi $\mathbf{x}(u, 0) = \mathbf{c}_1(u)$. Podobno dokažemo še $\mathbf{x}(u, 1) = \mathbf{c}_2(u)$, $\mathbf{x}(0, v) = \mathbf{d}_1(v)$ in $\mathbf{x}(v, 1) = \mathbf{d}_2(v)$. \square

Primer konstrukcije Coonsove krpe je viden na sliki 6.

Opomba: Coonsovo krpo lahko zapišemo tudi v matrični obliki:

$$\begin{aligned} \mathbf{x}(u, v) &= \begin{bmatrix} 1-u & u \end{bmatrix} \begin{bmatrix} \mathbf{d}_1(v) \\ \mathbf{d}_2(v) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{c}_1(u) & \mathbf{c}_2(u) \end{bmatrix} \begin{bmatrix} 1-v \\ v \end{bmatrix} \\ &- \begin{bmatrix} 1-u & u \end{bmatrix} \begin{bmatrix} \mathbf{x}(0,0) & \mathbf{x}(0,1) \\ \mathbf{x}(1,0) & \mathbf{x}(1,1) \end{bmatrix} \begin{bmatrix} 1-v \\ v \end{bmatrix}. \end{aligned} \quad (4.4)$$

Funkcije $1-u$, u , $1-v$ in v imenujemo **bazne funkcije**. Če v enačbi (4.4) te funkcije zamenjamo z drugimi pari takih funkcij, npr. $f_1(u)$, $f_2(u)$, $g_1(v)$ in $g_2(v)$, dobimo enačbo posplošene Coonsove krpe:

$$\begin{aligned} \mathbf{x}(u, v) &= \begin{bmatrix} f_1(u) & f_2(u) \end{bmatrix} \begin{bmatrix} \mathbf{d}_1(v) \\ \mathbf{d}_2(v) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{c}_1(u) & \mathbf{c}_2(u) \end{bmatrix} \begin{bmatrix} g_1(v) \\ g_2(v) \end{bmatrix} \\ &- \begin{bmatrix} f_1(u) & f_2(u) \end{bmatrix} \begin{bmatrix} \mathbf{x}(0,0) & \mathbf{x}(0,1) \\ \mathbf{x}(1,0) & \mathbf{x}(1,1) \end{bmatrix} \begin{bmatrix} g_1(v) \\ g_2(v) \end{bmatrix}. \end{aligned}$$

Opomba: Vsak par funkcij f_i in g_i se mora sešteti v ena. Da lahko interpoliramo, mora veljati tudi $f_1(0) = g_1(0) = 1$ in $f_1(1) = g_1(1) = 0$.

Primer 4.3. Naj bodo $\mathbf{c}_1(u) = (u, 0, u-u^2)^T$, $\mathbf{c}_2(u) = (u, 1, u)^T$, $\mathbf{d}_1(v) = (0, v^2, v-v^2)^T$ in $\mathbf{d}_2(v) = (1, v, v)^T$ robne krivulje. Izračunajmo točko na krpi pri parametrih $u = 0.2$ in $v = 0.5$.

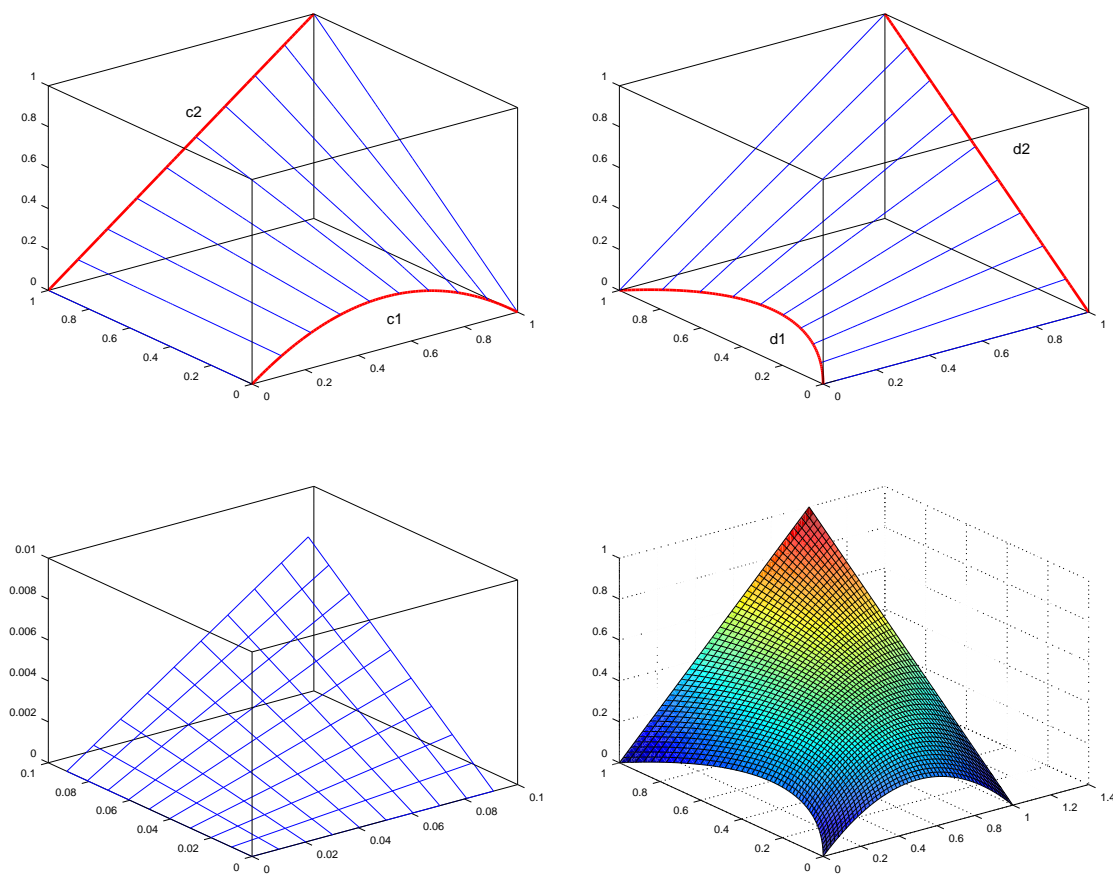
$$\mathbf{r}_c(0.2, 0.5) = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.18 \end{bmatrix}, \quad \mathbf{r}_d(0.2, 0.5) = \begin{bmatrix} 0.2 \\ 0.3 \\ 0.3 \end{bmatrix}, \quad \mathbf{r}_{cd}(0.2, 0.5) = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.1 \end{bmatrix}.$$

Dobimo:

$$\mathbf{x}(0.2, 0.5) = \mathbf{r}_c + \mathbf{r}_d - \mathbf{r}_{cd} = \begin{bmatrix} 0.2 \\ 0.3 \\ 0.38 \end{bmatrix}.$$

Krpa, ko u in v pretečeta $[0, 1] \times [0, 1]$ je prikazana na sliki 6.

Poglejmo si še, kako so povezane Bézierove in Coonsove krpe. Robne krivulje pri Coonsovih krpah so polinomske Bézierove krivulje, zato nam njihovi kontrolni poligoni



Slika 6: Slike prikazuje ploskve \mathbf{r}_c , \mathbf{r}_d , \mathbf{r}_{cd} in Coonsovo krpo \mathbf{x} iz primera 4.3

podajo del kontrolne mreže celotne krpe. Za $m = n = 3$ npr. dobimo shemo:

$$\begin{array}{cccc} \mathbf{b}_{00} & \mathbf{b}_{01} & \mathbf{b}_{02} & \mathbf{b}_{03} \\ & \mathbf{b}_{10} & & \mathbf{b}_{13} \\ & \mathbf{b}_{20} & & \mathbf{b}_{23} \\ & \mathbf{b}_{30} & \mathbf{b}_{31} & \mathbf{b}_{32} & \mathbf{b}_{33} \end{array}$$

Za konstrukcijo kontrolne mreže moramo v tem primeru izračunati še štiri manjkajoče kontrolne točke. Najprej definirajmo točke:

$$\mathbf{b}_{i,j}^u := \left(1 - \frac{i}{m}\right) \mathbf{b}_{0,j} + \frac{i}{m} \mathbf{b}_{m,j}, \quad i = 1, 2, \dots, m-1, \quad j = 1, 2, \dots, n-1$$

in

$$\mathbf{b}_{i,j}^v := \left(1 - \frac{j}{n}\right) \mathbf{b}_{i,0} + \frac{j}{n} \mathbf{b}_{i,n}, \quad i = 1, 2, \dots, n-1, \quad j = 1, 2, \dots, m-1$$

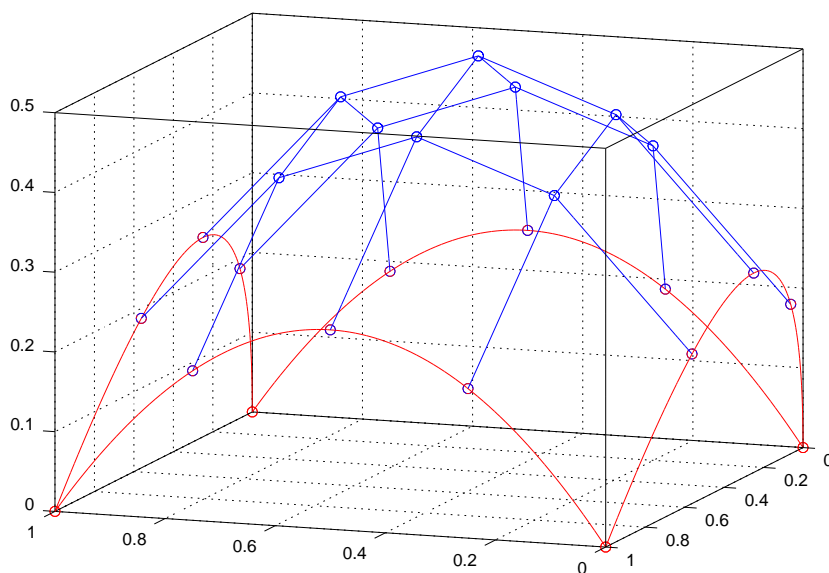
ter točke diskretne krpe, ki interpolirajo oglišča:

$$\mathbf{b}_{i,j}^{u,v} = \begin{bmatrix} 1 - \frac{j}{n} & \frac{j}{n} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,0} & \mathbf{b}_{0,n} \\ \mathbf{b}_{m,0} & \mathbf{b}_{m,n} \end{bmatrix} \begin{bmatrix} 1 - \frac{i}{m} \\ \frac{i}{m} \end{bmatrix}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n.$$

Kontrolne točke $\mathbf{b}_{i,j}$ dobimo podobno kot v (4.4):

$$\begin{aligned} \mathbf{b}_{i,j}^{u,v} &= \mathbf{b}_{i,j}^u + \mathbf{b}_{i,j}^v - \mathbf{b}_{i,j}^{u,v} \\ &= \left(1 - \frac{i}{m}\right) \mathbf{b}_{0,j} + \frac{i}{m} \mathbf{b}_{m,j} \\ &+ \left(1 - \frac{j}{n}\right) \mathbf{b}_{i,0} + \frac{j}{n} \mathbf{b}_{i,n} \\ &- \begin{bmatrix} 1 - \frac{j}{n} & \frac{j}{n} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,0} & \mathbf{b}_{0,n} \\ \mathbf{b}_{m,0} & \mathbf{b}_{m,n} \end{bmatrix} \begin{bmatrix} 1 - \frac{i}{m} \\ \frac{i}{m} \end{bmatrix}. \end{aligned}$$

Ta zapis imenujemo **diskretna Coonsova krpa**.

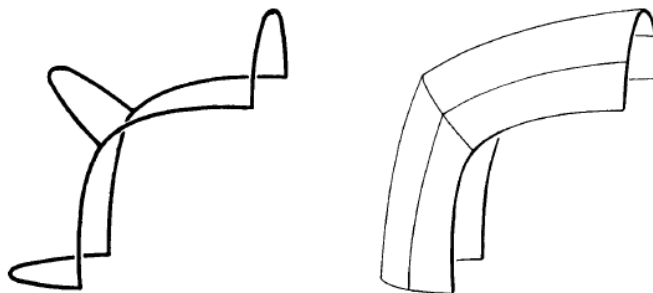


Slika 7: Primer diskretne Coonsove krpe za $m = n = 4$.

4.2 Delno bikubično ujemanje

Z bilinearnim ujemanjem sicer dobimo ploskev, ki je zlepek večih krp, ampak ima taka konstrukcija pomankljivosti. Tako pridobljena ploskev je preveč sploščena in pri nekaterih parametrih u, v ni gladka (glej sliko 8 vzeto iz [1]). Vzrok najdemo v tangentah vzdolž robnih krivulj, ki so hkrati odvisne tudi od podatkov, ki niso vezani na to krivuljo. Kot primer vzemimo robno krivuljo $\mathbf{x}(1, v)$. Vsaka sprememba na tej krivulji bo vplivala tudi na odvode vzdolž robne krivulje $\mathbf{x}(0, v)$.

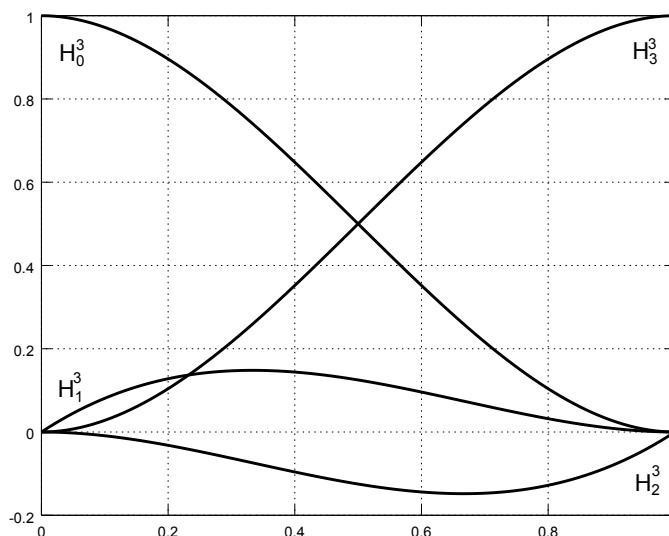
Za neodvisnost odvodov vzdolž ene izmed robnih krivulj od podatkov nasprotne robne krivulje uporabimo kubična Hermiteova polinoma H_0^3 in H_3^3 kot bazni funkciji.



Slika 8: Čeprav so robne krivulje, ki definirajo Coonsovi krpi, gladke, pa ploskev, sestavljena iz teh krp, ni gladka.

Naj bo $f_1 = g_1 = H_0^3 := B_0^3 + B_1^3$ in $f_2 = g_2 = H_3^3 := B_2^3 + B_3^3$. Hitro se prepričamo (glej sliko 9), da velja $\frac{d}{du}(H_0^3(u))|_{u=0} = 0$ in $\frac{d}{du}(H_3^3(u))|_{u=0} = 0$. Potem je odvod vzdolž $u = 0$ enak:

$$\mathbf{x}_u(0, v) = \begin{bmatrix} \mathbf{x}_u(0, 0), & \mathbf{x}_u(0, 1) \end{bmatrix} \begin{bmatrix} H_0^3(v) \\ H_3^3(v) \end{bmatrix}.$$



Slika 9: Kubični Hermiteovi polinomi na intervalu $[0, 1]$.

Sedaj na \mathbf{x}_u vzdolž $u = 0$ vplivata le tangenti $\mathbf{x}_u(0, 0)$ in $\mathbf{x}_u(0, 1)$.

Čeprav smo rešili prejšnjo težavo, se sedaj pojavi nova. Tako definirana ploskev je lahko v okolici ogljišč preveč sploščena. Razlog najdemo v uporabi le dveh od štirih kubičnih Hermiteovih polinomov. Tako H_0^3 kot H_3^3 imata odvod v robnih točkah intervalov enak nič (glej sliko 9), kar se prenese tudi na ploskev. V naslednjem razdelku bomo poiskali rešitev za ta problem.

4.3 Bikubično ujemanje

Za kubično Hermiteovo interpolacijo potrebujemo, poleg že danih podatkov iz mejnih krivulj, tudi prve odvode (glej sliko 10). Na voljo imamo torej naslednje podatke:

$$\mathbf{x}(u, 0), \mathbf{x}(u, 1), \mathbf{x}(0, v), \mathbf{x}(1, v)$$

in

$$\mathbf{x}_v(u, 0), \mathbf{x}_v(u, 1), \mathbf{x}_u(0, v), \mathbf{x}_u(1, v).$$

Podobno kot pri bilinearnem ujemanju (4.1) in (4.2) dobimo:

$$\mathbf{h}_c(u, v) = H_0^3(u)\mathbf{x}(0, v) + H_1^3(u)\mathbf{x}_u(0, v) + H_2^3(u)\mathbf{x}_u(1, v) + H_3^3(u)\mathbf{x}(1, v)$$

in

$$\mathbf{h}_d(u, v) = H_0^3(v)\mathbf{x}(u, 0) + H_1^3(v)\mathbf{x}_v(u, 0) + H_2^3(v)\mathbf{x}_v(u, 1) + H_3^3(v)\mathbf{x}(u, 1),$$

kjer sta $H_1^3(t) := \frac{1}{3}B_1^3(t)$, $H_2^3(t) := -\frac{1}{3}B_2^3(t)$ (glej sliko 9). Bikubični Hermiteov interpolant \mathbf{h}_{cd} izračunamo s tenzorskim produktom:

$$\mathbf{h}_{c,d}(u, v) = \begin{bmatrix} H_0^3(u) & H_1^3(u) & H_2^3(u) & H_3^3(u) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}(0, 0) & \mathbf{x}_v(0, 0) & \mathbf{x}_v(0, 1) & \mathbf{x}(0, 1) \\ \mathbf{x}_u(0, 0) & \mathbf{x}_{uv}(0, 0) & \mathbf{x}_{uv}(0, 1) & \mathbf{x}_u(0, 1) \\ \mathbf{x}_u(1, 0) & \mathbf{x}_{uv}(1, 0) & \mathbf{x}_{uv}(1, 1) & \mathbf{x}_u(1, 1) \\ \mathbf{x}(1, 0) & \mathbf{x}_v(1, 0) & \mathbf{x}_v(1, 1) & \mathbf{x}(1, 1) \end{bmatrix} \cdot \begin{bmatrix} H_0^3(v) \\ H_1^3(v) \\ H_2^3(v) \\ H_3^3(v) \end{bmatrix}. \quad (4.5)$$

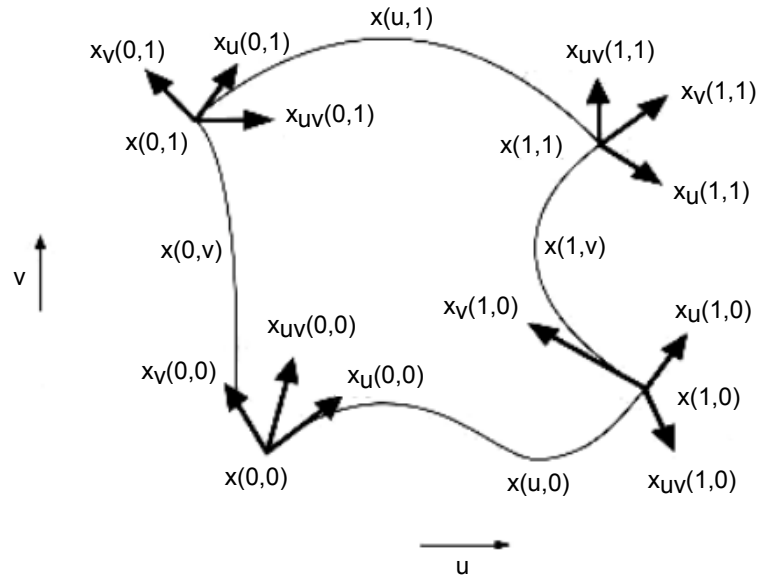
Coonsovo krpo potem zapišemo kot

$$\mathbf{x} = \mathbf{h}_c + \mathbf{h}_d - \mathbf{h}_{c,d}.$$

Opomba: Opazimo, da potrebujemo za izračun $\mathbf{h}_{c,d}$ podatke, ki jih ne želimo podati v začetni opis problema, to so podatki v sredini 4×4 matrike, definirane v (4.5). Ena izmed rešitev je, da definiramo vse mešane druge odvode v ogliščih kot ničelne vektorje.

4.4 Kompatibilnost

Ena izmed očitnih zahtev za konstruiranje Coonsovih krp je ujemanje štirih mejnih krivulj v ogliščih. Ta pogoj imenujemo **kompatibilnostni pogoj**. Če krivulje ne zadoščajo temu pogoju, jih prilagodimo, da se sekajo v ogliščih ploskve. Pri bikubičnem ujemanju Coonsovih krp moramo pri kompatibilnostnem pogoju paziti še na mešane druge odvode v $\mathbf{h}_{c,d}$. Vemo, da lahko menjamo vrstni red odvajanja pri mešanih parcialnih odvodih $\mathbf{x}_{uv} = \mathbf{x}_{vu}$, če je $\mathbf{x}(u, v)$ dvakrat zvezno odvedljiva. Vendar pa te lastnosti



Slika 10: Potrebni podatki za interpolacijo.

tukaj ne moremo uporabiti. Poglejmo primer pri $\mathbf{x}(0,0)$. Mešani drugi odvod v točki $\mathbf{x}(0,0)$ izračunamo kot

$$\mathbf{x}_{vu}(0,0) = \lim_{u \rightarrow 0} \frac{\partial}{\partial u} \mathbf{x}_v(u,0) \quad (4.6)$$

ali

$$\mathbf{x}_{uv}(0,0) = \lim_{v \rightarrow 0} \frac{\partial}{\partial v} \mathbf{x}_u(0,v). \quad (4.7)$$

Če sta $\mathbf{x}_{uv}(0,0)$ in $\mathbf{x}_{vu}(0,0)$ enaka, potem lahko to vnesemo v matriko (4.5) in je Coonsova krpa dobro definirana. Vendar mešana druga odvoda nista nujno enaka. Če vnesemo podatke samo v eno izmed enačb (4.6) ali (4.7), bomo le delno interpolirali dane podatke. Tudi če vse mešane druge odvode definiramo kot ničelne vektorje, problema ne rešimo popolnoma.

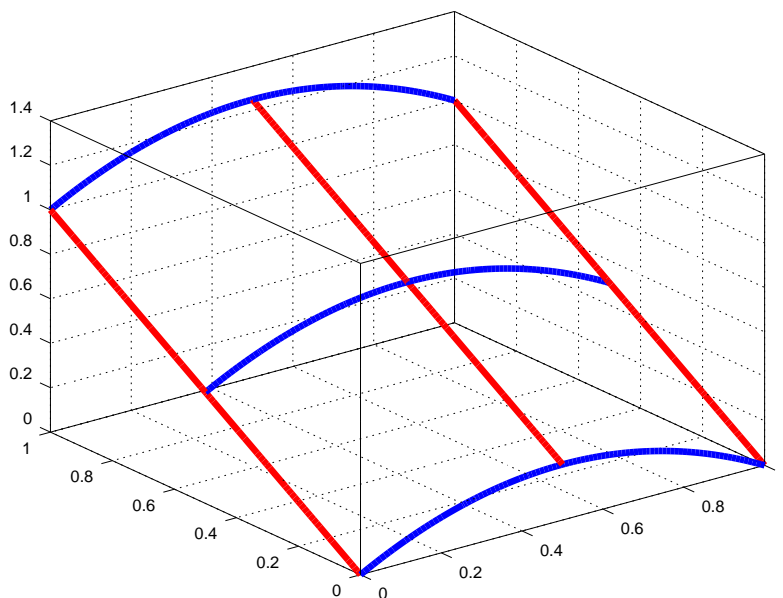
Za rešitev imamo dve možnosti: bodisi prilagodimo podatke bodisi, ko podatkov ne moremo spremeniti, uporabimo metodo Gregoryevih kvadratov. Slednja zamenja konstantne pogoje mešanih drugih odvodov v matriki s spremenljivimi. Spremenljive pogoje izračunamo kot:

$$\begin{aligned} \mathbf{x}_{uv}(0,0) &= \frac{u \frac{\partial}{\partial v} \mathbf{x}_u(0,0) + v \frac{\partial}{\partial u} \mathbf{x}_v(0,0)}{u+v}, \\ \mathbf{x}_{uv}(0,1) &= \frac{-u \frac{\partial}{\partial v} \mathbf{x}_u(0,1) + (v-1) \frac{\partial}{\partial u} \mathbf{x}_v(0,1)}{-u+v-1}, \\ \mathbf{x}_{uv}(1,0) &= \frac{(1-u) \frac{\partial}{\partial v} \mathbf{x}_u(1,0) + v \frac{\partial}{\partial u} \mathbf{x}_v(1,0)}{1-u+v}, \\ \mathbf{x}_{uv}(1,1) &= \frac{(u-1) \frac{\partial}{\partial v} \mathbf{x}_u(1,1) + (v-1) \frac{\partial}{\partial u} \mathbf{x}_v(1,1)}{u-1+v-1}. \end{aligned}$$

4.5 Gordonove krpe

Gordonove krpe so posplošitev Coonsovih krp. Včasih samo štiri robne krivulje ne zadoščajo za konstrukcijo ploskve, saj imamo lahko podano celo mrežo krivulj (glej sliko 11), ki jih moramo interpolirati. Krivulje zapišemo kot $\mathbf{g}(u_i, v)$, $i = 0, 1, \dots, m$, in $\mathbf{g}(u, v_j)$, $j = 0, 1, \dots, n$. Pokazali bomo konstrukcijo ploskve \mathbf{g} , ki interpolira vse krivulje v tej mreži. Ideja je podobna kot pri konstrukciji Coonsovih krp: najprej poiščemo ploskev \mathbf{g}_1 , ki interpolira krivulje $\mathbf{g}(u_i, v)$, nato ploskev \mathbf{g}_2 , ki interpolira krivulje $\mathbf{g}(u, v_j)$. Na koncu dobljeno seštejemo in odštejemo ploskev \mathbf{g}_{12} .

Poiščimo najprej enačbo ploskve \mathbf{g}_1 . V primeru, ko imamo samo dve krivulji $\mathbf{g}(u_0, v)$



Slika 11: Mreža krivulj, ki jih želimo interpolirati.

in $\mathbf{g}(u_1, v)$, dobimo:

$$\mathbf{g}_1(u, v) = L_0^1(u)\mathbf{g}(u_0, v) + L_1^1(u)\mathbf{g}(u_1, v), \quad (4.8)$$

kjer sta linearna L_0^1 in L_1^1 Lagrangeeva polinoma. Če je krivulj več, lahko (4.8) posplošimo v:

$$\mathbf{g}_1(u, v) = \sum_{i=0}^m \mathbf{g}(u_i, v)L_i^m(u), \quad (4.9)$$

kjer so

$$L_i^m(t) = \frac{\prod_{j=0, j \neq i}^m (t - t_j)}{\prod_{j=0, j \neq i}^m (t_i - t_j)}.$$

Podobno dobimo še ploskev \mathbf{g}_2 :

$$\mathbf{g}_2(u, v) = \sum_{j=0}^n \mathbf{g}(u, v_j)L_j^n(v). \quad (4.10)$$

Podobno kot v prejšnjem poglavju dobimo tudi zadnjo ploskev $\mathbf{g}_{1,2}$ kot:

$$\mathbf{g}_{1,2}(u, v) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{g}(u_i, v_j) L_i^m(u) L_j^n(v). \quad (4.11)$$

Izrek 4.4. Naj bodo $\mathbf{g}_1, \mathbf{g}_2$ in $\mathbf{g}_{1,2}$ definirane kot v (4.9), (4.10) in (4.11). Potem Gordonova krpa \mathbf{g} oblike:

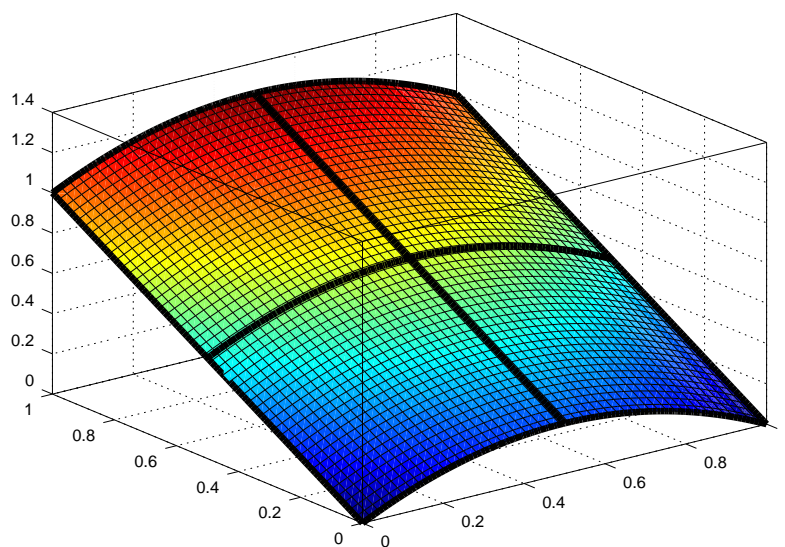
$$\mathbf{g} = \mathbf{g}_1 + \mathbf{g}_2 - \mathbf{g}_{1,2}$$

interpolira dano družino krivulj $\mathbf{g}(u_i, v)$, $i = 0, 1, \dots, m$, $\mathbf{g}(u, v_j)$, $j = 0, 1, \dots, n$.

Dokaz.

$$\begin{aligned} \mathbf{g}(u_k, v) &= \mathbf{g}_1(u_k, v) + \mathbf{g}_2(u_k, v) - \mathbf{g}_{1,2}(u_k, v) \\ &= \sum_{i=0}^m \mathbf{g}(u_i, v) L_i^m(u_k) + \sum_{j=0}^n \mathbf{g}(u_k, v_j) L_j^n(v) - \sum_{i=0}^m \sum_{j=0}^n \mathbf{g}(u_i, v_j) L_i^m(u_k) L_j^n(v) \\ &= \sum_{i=0}^m \mathbf{g}(u_i, v) L_i^m(u_k) + \sum_{j=0}^n \mathbf{g}(u_k, v_j) L_j^n(v) - \sum_{j=0}^n L_j^n(v) \sum_{i=0}^m \mathbf{g}(u_i, v_j) L_i^m(u_k) \\ &= \mathbf{g}(u_k, v) + \sum_{j=0}^n \mathbf{g}(u_k, v_j) L_j^n(v) - \sum_{j=0}^n \mathbf{g}(u_k, v_j) L_j^n(v) \\ &= \mathbf{g}(u_k, v). \end{aligned}$$

Pri tem smo upoštevali $L_i^m(u_k) = \delta_{i,k}$, kjer je $\delta_{i,k}$ Kroneckerjeva delta funkcija. Podobno dokažemo tudi za krivulje $\mathbf{g}(u, v_j)$, $j = 0, 1, \dots, n$. \square



Slika 12: Gordonova krpa, ki interpolira mrežo krivulj na sliki 11.

5 Zaključek

V zaključni nalogi smo spoznali matematično ozadje Bézierovih krivulj in krp ter Gordon-Coonsove krpe, ki se danes uporabljajo na različnih področjih. Bézierove krivulje so poseben primer parametričnih krivulj, ki jih definiramo s kontrolnimi točkami. Podobno velja tudi za Bézierove krpe, ki jih določimo s kontrolno mrežo. Coonsove krpe smo definirali s štirimi robnimi krivuljami, ki naj jih te krpe interpolirajo. Ugotovili smo, da pri interpolaciji naletimo na več težav: krpe so sploščene in sploščena. Da bi odpravili napake, smo pogledali lastnosti parcialnih odvodov pri Coonsovih krpah. Na koncu smo definirali še Gordonove krpe, ki so posplošitev Coonsovih krp. Razlikujejo se po tem, da imamo namesto robnih krivulj podano celo mrežo krivulj. Več informacij najdemo v knjigi G. Farina *Curves and surfaces for CAGD*. Odkritja Béziera, de Casteljaua, Coonsa in Gordona imajo tudi še v današnjih časih pomembno vlogo pri razvoju računalniškega oblikovanja.

6 Literatura

- [1] G. FARIN, *Curves and surfaces for CAGD, Fifth ed.*, Morgan Kaufmann publishers, San Francisco, 2006. (*Citirano na straneh II, III, 1 in 22.*)
- [2] G. FARIN in D. HANSFORD, Discrete Coons patches, *Computer Aided Geometric Design* 16 (1999), 691–700. (*Citirano na strani 1.*)
- [3] I.D. FAUX in M.J. PRATT, *Computational Geometry for Design and Manufacture*, Ellis Horwood Ltd, 1980. (*Citirano na strani 1.*)
- [4] D.L. FLINN, *Blended and Ruled Surfaces*, MA 323 Geometric Modelling Course Notes: Day 31. (*Citirano na strani 1.*)
- [5] P.S. HECKBERT, Bilinear Coons Patch Image Wrapping, v: P.S. Heckbert (ur.), *Graphic Gems IV*, Academic Press Professional, Inc., 1994, 438–446. (*Citirano na strani 1.*)
- [6] G. JAKLIČ, *Krivulje in ploskve v računalniško podprtem geometrijskem oblikovanju, skripta*, Fakulteta za matematiko in fiziko, Ljubljana, 2011. (*Citirano na strani 1.*)
- [7] M. TUTA, *Bézierove krivulje in ploskve: od teoretičnega ozadja do računalniških aplikacij*, Fakulteta za matematiko, naravoslovje in informacijske tehnologije, Koper, 2012. (*Citirano na strani 1.*)
- [8] V. VOLAŠ, *Racionalne Bézierove krivulje*, Univerza na Primorskem, Fakulteta za matematiko, naravoslovje in informacijske tehnologije, 2013. (*Citirano na strani 1.*)

Priloge

A Priloga

Algoritem 1: de Casteljauev algoritem

Vhod: kontrolne točke in parameter N , ki nam pove, kako na gosto računamo točke na krivulji

Izhod: graf krivulje in njen kontrolni poligon

```
1 function deCasteljau(kontrolneT,N)
2 m = size(kontrolneT)(1);
3 n = size(kontrolneT)(2);
4 T = zeros(3,N);
5 s = linspace(0,1,N);
6 for k = 1:N
7     t = s(k);
8     for j = 1:m
9         for i = 1:(n-1)
10            kontrolneT(i,:) = (1-t)*kontrolneT(i,:)+t*kontrolneT(i+1,:);
11        end;
12    end;
13    T(:,[k]) = kontrolneT(:,[1]);
14 end;
15 plot3(T(1,:),T(2,:),T(3,:));
16 hold on;
17 plot3(kontrolneT(1,:),kontrolneT(2,:),kontrolneT(3,:),'ro');
18 hold on;
19 plot3(kontrolneT(1,:),kontrolneT(2,:),kontrolneT(3,:),'r');
20 hold off;
21 end;
```

B Priloga

Algoritem 2: ploskevB

Vhod: kontrolne točke in parameter N , ki nam pove, kako na gosto računamo točke na krpi

Izhod: nariše Bézierovo krpo

```
1 function ploskevB(kontrolneT,N)
2 m = size(kontrolneT)(1)-1;
3 n = (size(kontrolneT)(2)/3)-1;
4 t = linspace(0,1,N);
5 for i = 1:N
6     for j = 1:N
7         u = t(i);
8         v = t(j);
9         P(i,j,:) = bilinearne(kontrolneT,u,v,N);
10    end;
11 end;
12 surf(P(:,:,1),P(:,:,2),P(:,:,3));
13 end;
```

Algoritem 3: bilinearna

Vhod: kontrolne točke, parametra u, v , pri katerih računamo točke na ploskvi in parameter N , ki nam pove, kako na gosto računamo točke na ploskvi

Izhod: izračuna točke na Bézierovi krpi

```
1 function L = bilinearna(B,u,v,N)
2 T = zeros(N,N,3);
3 m = size(B)(1)-1;
4 n = (size(B)(2)/3)-1;
5 B1 = tenzor(B);
6 for k=1:m
7     for i = 1:(m-k+1)
8         for j = 1:(n-k+1)
9             T(i,j,:) = (1-u)*(1-v)*B1(i,j,:)+u*(1-v)*B1(i+1,j,:)+(1-u)*v*B1(i,j+1,:)+u*v*B1(i+1,j+1,:);
10        end;
11    end;
12    B1=T;
13 end;
14 L = T(1,1,:);
15 end;
```

Algoritem 4: tenzor

Vhod: kontrolne točke

Izhod: matriko B pretvori v tenzor

```
1 function A=tenzor(B)
2 m=size(B)(1)-1;
3 n=(size(B)(2)/3) -1;
4 for k=1: m+1
5     for l=1: n+1
6         A(k,l,)= B(k,3*l-2:3*l);
7     end;
8 end;
```

C Priloga

Algoritem 5: Coonsova krpa

Vhod: štiri mejne krivulje in njihova oglišča

Izhod: nariše Coonsovo krpo in mejne krivulje

```
1 function krpaCoons(c1,c2,d1,d2,P1,P2,P3,P4)
2 u = linspace(0,1,N);
3 v = linspace(0,1,N);
4 R = zeros(N,N);
5 for i = 1:N
6     k = 1;
7     for j = 1:N
8         rc = (1-v(j))*c1(:,i)+v(j)*c2(:,i);
9         rd = (1-u(i))*d1(:,j)+u(i)*d2(:,j);
10        rcd = (1-u(i))*(1-v(j))*P1+u(i)*(1-v(j))*P2+(1-u(i))*v(j)*P3+u(i)*v(j)*P4;
11        R(i,k:(k+2)) = rc+rd-rcd;
12        k = k+3;
13    end;
14 end;
15 P = tenzor(R);
16 surf(P(:,:,1),P(:,:,2),P(:,:,3));
17 hold on;
18 plot3(c1(1,:),c1(2,:),c1(3,:), 'r', 'LineWidth', 2.5)
19 hold on
20 plot3(c2(1,:),c2(2,:),c2(3,:), 'r', 'LineWidth', 2.5)
21 hold on
22 plot3(d1(1,:),d1(2,:),d1(3,:), 'g', 'LineWidth', 2.5)
23 hold on
24 plot3(d2(1,:),d2(2,:),d2(3,:), 'g', 'LineWidth', 2.5)
25 hold on
26 end;
```

Vhod: primer zapisa krivulj za algoritem 5

```
1 u = linspace(0,1,N);
2 v = linspace(0,1,N);
3 % c1 = [u;0;u-u^2];
4 c1x = u;
5 c1y = zeros(1,N);
6 c1z = zeros(1,N);
7 for i = 1:N
8     c1z(i) = u(i)-(u(i))^2;
9 end;
10 c1 = [c1x;c1y;c1z];
11 % c2 = [u,1,u];
12 c2x = u;
13 c2y = ones(1,N);
14 c2z = u;
15 c2 = [c2x;c2y;c2z];
16 % d1 = [0;v^2;v-v^2];
17 d1x = zeros(1,N);
18 d1y = zeros(1,N);
19 for i = 1:N
20     d1y(i) = (v(i))^2;
21 end;
22 d1z = zeros(1,N);
23 for i = 1:N
24     d1z(i) = v(i)-(v(i))^2;
25 end;
26 d1 = [d1x;d1y;d1z];
27 % d2 = [1;v;v];
28 d2x = ones(1,N);
29 d2y = v;
30 d2z = v;
31 d2 = [d2x;d2y;d2z];
32 P1 = [0;0;0];
33 P2 = [1;0;0];
34 P3 = [0;1;0];
35 P3 = [1;1;1];
```

Algoritem 6: vrednostCoons

Vhod: parametra u in v , štiri mejne krivulje ter njihova oglišča

Izhod: izračuna točko na Coonsovi krpi pri parametrih u in v

```
1 function vrednostCoons(u,v,c1,c2,d1,d2,P1,P2,P3,P4)
2 rc = (1-v)*c1+v*c2;
3 rd = (1-u)*d1+u*d2;
4 rcd = (1-u)*(1-v)*P1+u*(1-v)*P2+(1-u)*v*P3+u*v*P4;
5 T = rc+rd-rcd
6 end;
```

Vhod: primer zapisa krivulj za algoritem 6

```
1 c1 = [u;0;u-u^2];
2 c2 = [u;1;u];
3 d1 = [0;v^2;v-v^2];
4 d2 = [1;v;v];
5 P1 = [0;0;0];
6 P2 = [1;0;0];
7 P3 = [0;1;0];
8 P4 = [1;1;1];
```

D Priloga

Algoritem 7: Gordonova krpa

Vhod: krivulje, presečišča krivulj, vektor x in parameter N , ki nam pove, kako na gosto računamo točke na krpi

Izhod: nariše Gordonovo krpo

```
1 function krpaGordon(g_u,g_v,g_uv,x,N)
2 t = linspace(0,1,N);
3 R = zeros(N,N);
4 for i = 1:N
5     u = t(i);
6     k = 1;
7     for j = 1:N
8         v = t(j);
9         m = size(g_u)(2);
10        n = size(g_v)(2);
11        g1 = g2 = g12 = 0;
12        for l = 1:m
13            g1 = g_u(:,l)*Lagrange(x,l,v)+g1;
14            g2 = g_v(:,l)*Lagrange(x,l,u)+g2;
15            c = 0;
16            for s = 1:n
17                g12 = g_uv(l,s+c:s+c+2)*Lagrange(x,l,u)*Lagrange(x,s,v)+g12;
18                c = c+2;
19            end;
20        end;
21        R(i,k:(k+2)) = g1+g2-g12;
22        k = k+3;
23    end;
24 end;
25 P=tensor(R);
26 surf(P(:,:,1),P(:,:,2),P(:,:,3));
27 end;
```

Vhod: primer zapisa podatkov za algoritem 7

```
1 g00=[0 0 0];
2 g01=[0.5 0 0.25];
3 g02=[1 0 0];
4 g10=[0 0.5 0.5];
5 g11=[0.5 0.5 0.75];
6 g12=[1 0.5 0.5];
7 g20=[0 1 1];
8 g21=[0.5 1 1.25];
9 g22=[1 1 1];
10 g_uv=[g00 g10 g20;g01 g11 g21;g02 g12 g22];
11 c0=[u;0;u-u^2];
12 c1=[u;0.5;0.5+u-u^2];
13 c2=[u;1;1+u-u^2];
14 g_u=[c0 c1 c2];
15 d0=[0;v;v];
16 d1=[0.5;v;0.25+v];
17 d2=[1;v;v];
18 g_v=[d0 d1 d2];
19 x=[0 0.5 1];
```

Algoritem 8: Lagrangeev polinom

Vhod: v vektor x so podane vrednosti polinoma, m je stopnja Lagrangeevega polinoma, parameter u, pri katerem računamo vrednost polinoma

Izhod: vrednost Lagrangeevega polinoma

```
1 function v = Lagrange(x,m,u)
2 n = length(x);
3 v = zeros(size(u));
4 for j = [1:m-1 m+1:n]
5     v = (u-x(j))./(x(m)-x(j)).*v;
6 end;
```
