

UNIVERZA NA PRIMORSKEM

FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

MATEMATIČNE ZNANOSTI - 2. stopnja

Marko Tuta

**Bézierove krivulje in ploskve: od
teoretičnega ozadja do računalniških
aplikacij**

Magistrsko delo

Koper, julij 2012

Mentor: doc. dr. Vito Vitrih

Ključna dokumentacijska informacija

Ime in PRIIMEK: Marko TUTA

Naslov magistrskega dela: Bézierove krivulje in ploskve: od teoretičnega ozadja do računalniških aplikacij

Kraj: Koper

Leto: 2012

Število listov: 102 Število slik: 65

Število prilog: 5 Število strani prilog: 9

Število referenc: 22

Mentor: doc. dr. Vito Vitrih

UDK: 519.65(043.2)

Ključne besede: Bézierove krivulje, Bézierove ploskve, Bézierovi trikotniki, polinom, kontrolne točke, Octave, Blender

Math. Subj. Class. (2010): 65D17, 65D18, 65D07, 68U07, 68U05

Izvlaček:

Na začetku šestdesetih let prejšnjega stoletja sta francoska matematika P. Bézier in P. de Casteljau neodvisno defnirala nov tip krivulj in ploskev, ki so bile s svojimi lastnostmi primerne za računalniško oblikovanje in načrtovanje. Defnirala sta jih s pomočjo t.i. kontrolnih točk in ne več s pomočjo točk, ki so ležale na njih. V magistrskem delu se bomo osredotočili na vprašanje, zakaj so Bézierove krivulje in ploskve primerne za računalniško modeliranje in zakaj so dosegle toliko uspeha. Opisali bomo matematično ozadje, ki definira Bézierove krivulje in ploskve ter opiše njihove lastnosti. V osrednjih poglavjih si podglavja sledijo v podobnem vrstnem redu z namenom poudariti razlike in podobnosti med Bézierovimi krivuljami in ploskvami. Poleg teoretičnega opisa bomo z njimi računalniško modelirali konkretne predmete in objekte s pomočjo računalniškega programa Blender. Prav tako bomo v magistrsko delo vključili tudi implementacijo nekaterih algoritmov, kar bo narejeno s programskim jezikom Octave.

Key words documentation

Name and SURNAME: Marko TUTA

Title of Master's degree: Bézier curves and surfaces: from the theoretical background to computer applications

Place: Koper

Year: 2012

Number of pages: 102 Number of figures: 65

Number of appendices: 5 Number of appendix pages: 9

Number of references: 22

Supervisor: Assist. Prof. Vito Vitrih

UDC: 519.65(043.2)

Key words: Bézier curves, Bézier patches, Bézier triangles, polynomial, control points, Octave, Blender

Math. Subj. Class. (2010): 65D17, 65D18, 65D07, 68U07, 68U05

Abstract:

During the early 60's French mathematicians P. Bézier and P. de Casteljau discovered independently the same type of curves and surfaces which were suitable for computer design. Their definitions of new curves and surfaces were based on the concept of control points: curves and surfaces were no longer defined with points which were lying on them. In the master thesis we will focus on the fact, why Bézier curves and surfaces are suitable for computer and geometric design and why they achieved such success. We will discuss the mathematical background which defines Bézier curves and surfaces and their most interesting properties. We want also to explain similarities and differences between Bézier curves and surfaces. For this purpose we divided each chapter in similar sections. On the other hand we will deal with computer modelling of several objects with the modelling software Blender. We will include also some examples of Octave's programming codes which implement some algorithms described in the master's thesis.

Kazalo

| | | |
|----------|---|-----------|
| 1 | Uvod | 1 |
| 2 | Osnovni pojmi | 3 |
| 2.1 | Točke in vektorji | 3 |
| 2.2 | Afine preslikave | 4 |
| 2.3 | Parametrične krivulje | 6 |
| 2.4 | Linearna interpolacija | 7 |
| 2.5 | Baricentrične koordinate v ravnini | 9 |
| 3 | Bézierove krivulje | 11 |
| 3.1 | de Casteljauov algoritem | 11 |
| 3.2 | Bernsteinova oblika Bézierove krivulje | 13 |
| 3.3 | Pomembne lastnosti Bézierovih krivulj | 15 |
| 3.4 | Višanje stopnje | 16 |
| 3.5 | Odvodi | 18 |
| 3.6 | Zveznost in gladkost Bézierovih zlepkov | 21 |
| 3.6.1 | C^k zveznost | 21 |
| 3.6.2 | Geometrijska zveznost G^k | 22 |
| 3.7 | Racionalne Bézierove krivulje | 24 |
| 3.8 | Matrična oblika Bézierove krivulje | 25 |
| 3.9 | Računalniško oblikovanje z Bézierovimi krivuljami | 27 |
| 4 | Bézierove ploskve | 31 |
| 4.1 | Bilinearna interpolacija | 32 |
| 4.2 | de Casteljauov algoritem | 32 |
| 4.3 | Bernsteinova oblika Bézierovih ploskev <i>ali</i> tenzorski produkt | 33 |
| 4.4 | Pomembne lastnosti Bézierovih ploskev | 34 |
| 4.5 | Višanje stopnje | 36 |
| 4.6 | Odvodi | 38 |
| 4.7 | Normalni vektorji | 41 |
| 4.8 | C^r zveznost | 42 |
| 4.9 | Racionalne Bézierove ploskve | 43 |
| 4.10 | Matrična oblika Bézierove ploskve | 45 |
| 4.11 | Računalniško oblikovanje z Bézierovimi ploskvami | 47 |
| 5 | Bézierovi trikotniki | 51 |
| 5.1 | de Casteljauov algoritem | 51 |
| 5.2 | Bernsteinovi polinomi | 54 |
| 5.3 | Pomembne lastnosti Bézierovih trikotnikov | 57 |

| | | |
|----------|---|-----------|
| 5.4 | Višanje stopnje | 58 |
| 5.5 | Odvodi | 59 |
| 5.6 | C^r zveznost | 62 |
| 6 | Računalniške aplikacije | 67 |
| 6.1 | Računalniški program Blender | 67 |
| 6.1.1 | Grafični uporabniški vmesnik programa | 67 |
| 6.1.2 | Glavni ukazi in lastnosti | 68 |
| 6.1.3 | Barvanje in dodajanje efektov | 72 |
| 6.2 | Modeliranje avtomobila | 73 |
| 6.2.1 | Karoserija | 73 |
| 6.2.2 | Pnevmatike | 78 |
| 6.2.3 | Končni model | 81 |
| 7 | Zaključek | 83 |
| | Literatura | 84 |
| | Priloga A | 85 |
| | Priloga B | 88 |
| | Priloga C | 89 |
| | Priloga Č | 91 |
| | Priloga D | 92 |

Slike

| | | |
|------|---|----|
| 2.1 | Obarvani del je konveksna ovojnica narisane poligona. | 4 |
| 2.2 | Primer striga. | 6 |
| 2.3 | Vpliv baricentričnih koordinat u, v, w v trikotniku. | 10 |
| 3.1 | Bézierova krivulja \mathbf{b}^4 in njen kontrolni poligon. | 13 |
| 3.2 | Primerjava med originalnim poligonom in novim poligonom pri višanju stopnje krivulje. | 18 |
| 3.3 | Odvod Bézierove krivulje. | 19 |
| 3.4 | Primer Bézierovega zlepk, ki je \mathcal{C}^1 zvezen, in prikaz izbranih vozlov. | 23 |
| 3.5 | Racionalne Bézierove krivulje pri različnih izbirah parametra ω_1 : če je $\omega_1 > 1$, se krivulja približa točki \mathbf{b}_1 (zelena barva), če pa je $\omega_1 < 1$, se krivulja oddalji od točke \mathbf{b}_1 (modra barva). Če je $\omega_1 = 1$, Bézierova krivulja (rdeča barva) ostane nepremenjena v okolici točke \mathbf{b}_1 | 26 |
| 3.6 | Začetek oblikovanja prereza steklenice. | 28 |
| 3.7 | Popravek uteži ustrezne kontrolne točke. | 29 |
| 3.8 | Končni prerez steklenice, ki ga potem dopolnimo. | 29 |
| 3.9 | Steklenica po zavrtenju prereza. | 30 |
| 3.10 | “Gladka” steklenica in njen končni barvni videz. | 30 |
| 4.1 | Primer parametrične ploskve in njena domena. Barve na ploskvi predstavljajo višino ploskve oz. z -komponento točk na ploskvi. | 31 |
| 4.2 | Primer hiperboličnega paraboloida. | 32 |
| 4.3 | Bézierova ploskev s kontrolno mrežo, kjer rdeči kvadrati predstavljajo točko na ploskvi, ki smo jo izračunali pri parametrih $(u, v) = (0.3, 0.5)$ | 34 |
| 4.4 | Bézierova ploskev z robno in izoparametrično krivuljo. | 37 |
| 4.5 | Bézierova ploskev z novo kontrolno mrežo po višanju stopnje. | 39 |
| 4.6 | Prikaz dveh Bézierovih ploskev stopnje $(2, 3)$, ki skupaj tvorita \mathcal{C}^1 zlepek. | 44 |
| 4.7 | Kontrolni mreži dveh Bézierovih ploskev, ki skupaj tvorita \mathcal{C}^1 zlepek. | 44 |
| 4.8 | Domena zlepk. | 44 |
| 4.9 | Primerjava med polinomsko Bézierovo ploskvijo in racionalno Bézierovo ploskvijo, kjer je $\omega_{1,1} = 5$ | 45 |
| 4.10 | Bézierova ploskev stopnje $(3, 3)$ | 47 |
| 4.11 | Racionalna Bézierova ploskev, ki ima vse uteži enake ena. | 48 |
| 4.12 | Višanje uteži vsem štirim označenim točkam. | 48 |
| 4.13 | Nove točke bomo uporabili, da bomo oblikovali ročaje posode. | 48 |
| 4.14 | Posoda ima sedaj že oblikovano dno in ročaje. | 49 |
| 4.15 | Bézierova ploskev stopnje $(3, 3)$, ki predstavlja mizo. | 49 |
| 4.16 | Prikaz posode na mizi, potem ko dodamo debeline, barve, prozornost in luč. | 50 |
| 5.1 | Primer Bézierovega trikotnika in njegove trikotne domene. | 51 |

| | | |
|------|---|----|
| 5.2 | Bézierov trikotnik s kontrolno mrežo. | 54 |
| 5.3 | Bézierov trikotnik s prvotno kontrolno mrežo (modra barva) in s kontrolno mrežo po višanju stopnje (rdeča barva). | 60 |
| 5.4 | Prikaz sestavljene domene. | 63 |
| 5.5 | Domena za zlepek dveh Bézierovih trikotnikov. | 64 |
| 5.6 | Zlepek dveh Bézierovih trikotnikov, ki je C^1 zvezen. | 65 |
| 5.7 | Kontrolni mreži obeh Bézierovih trikotnikov. | 66 |
| 6.1 | Začetna scena Blenderja. | 67 |
| 6.2 | Glavno okno razdelimo na štiri podokna. | 68 |
| 6.3 | Primer uporabe ukaza <i>Rotate</i> za rotacijo objekta. | 69 |
| 6.4 | V pogledu <i>Edit Mode</i> lahko objektu spremenimo strukturo. | 69 |
| 6.5 | Krivulje, ploskve in druge objekte dodajamo na sceno z gumbom <i>Add</i> | 70 |
| 6.6 | Raznolikost orodja <i>Modifier</i> | 70 |
| 6.7 | Zrcaljenje objekta s pomočjo ukaza <i>Modifier Mirror</i> | 71 |
| 6.8 | Ukaz, s katerim spremenimo krivulje ali ploskve v objekte tipa <i>Mesh</i> | 71 |
| 6.9 | Bézierova krivulja pred in po spremembi v objekt tipa <i>Mesh</i> | 72 |
| 6.10 | Okni <i>Materials</i> in <i>Texture</i> | 72 |
| 6.11 | Primer kocke z danimi lastnostmi. | 73 |
| 6.12 | Slike, ki prikazujejo vse perspektive avtomobila, vstavimo v Blender. | 73 |
| 6.13 | Bézierova ploskev s kontrolnimi točkami na isti višini. | 74 |
| 6.14 | Primer uporabe ukaza <i>Extrude</i> | 74 |
| 6.15 | Z <i>RingCut</i> ploskvi poljubno dodamo nove točke. | 75 |
| 6.16 | Z ukazom <i>Face</i> zapolnimo nastale prazne prostore med ploskvami. | 75 |
| 6.17 | <i>Modifier Mirror</i> nam prihrani veliko dela. | 76 |
| 6.18 | Paziti moramo, da pravilno postavimo ploskve v vseh oknih Blenderja. | 76 |
| 6.19 | Sprednji del karoserije. | 77 |
| 6.20 | Začetek oblikovanja zadnjega dela karoserije. | 77 |
| 6.21 | Končni model karoserije. | 78 |
| 6.22 | Začetek oblikovanja avtomobilske gume. | 78 |
| 6.23 | Profil avtomobilske gume dopolnimo. | 79 |
| 6.24 | Prikaz krožnice okoli katere bomo ovili dobljen del gume. | 79 |
| 6.25 | Z <i>Modifier Curve</i> smo profil gume postavili v obliko kroga. | 80 |
| 6.26 | Avtomobilska guma, ki jo bomo kasneje dodali h karoseriji. | 80 |
| 6.27 | Končni model avtomobila. | 81 |
| 6.28 | Modeliranje črk z Bézierovimi krivuljami. | 81 |
| 6.29 | Barve in drugi efekti nas privedejo do realističnega prikaza avtomobila. | 82 |

Zahvala

Iskreno se zahvaljujem doc. dr. Vitu Vitrihu, ki me je usmerjal pri izdelavi magistrskega dela. Njegovi nasveti glede vsebine in oblike nastajajoče naloge so bili vedno primerni in učinkoviti in so pripomogli, da je delo nastalo v najboljši obliki.

Zahvaljujem se tudi svoji družini, ker mi je bila med študijem moralno in finančno vedno ob strani in je tako pripomogla k temu, da sem dosegel vse zastavljene cilje na univerzi.

Zahvaljujem pa se predvsem Eleni, ker me je s svojo ljubeznijo toliko navdihnila, da sem lahko v pomembnih trenutkih izbral pravo pot.

Poglavje 1

Uvod

V petdesetih letih prejšnjega stoletja so se v industrijskem svetu soočali z naslednjo problematiko: na eni strani so imeli na papirju načrte za izdelovanje npr. avtomobila ali letala; na drugi strani pa so imeli na razpolago stroje, ki bi lahko te objekte izdelovali. Vprašanje je bilo, kako načrte na papirju “sporočiti” prvim računalnikom, ki bi potem vodili stroje pri izdelavi avtomobila ali letala. Na začetku šestdesetih let sta francoska matematika P. Bézier in P. de Casteljau prišla do odgovora. Neodvisno sta dosegla isti rezultat po različnih poteh: definirala sta nov tip krivulj in kasneje ploskev, ki so bile s svojimi lastnostmi primerne za računalniško oblikovanje in načrtovanje. Matematiki so nato začeli razširjati ideje obeh Francozov. Leta 1974 sta R. Barnhill in R. Riesenfeld prvič uporabila izraz *CAGD (Computer Aided Geometric Design)*, ko sta organizirala konferenco v ameriški državi Utah na to temo. Prva knjiga je izšla leta 1979 z naslovom *Computational Geometry for Design and Manufacture*; leta 1984 pa sta R. Barnhill in W. Boehm začela izdajati revijo *Computer Aided Geometric Design*. Krivulje in ploskve, ki sta jih odkrila francoska matematika, jih danes veliko uporabljajo pri računalniškem modeliranju in načrtovanju. V magistrskem delu se bomo osredotočili na vprašanje, zakaj so Bézierove krivulje in ploskve primerne za računalniško modeliranje in zakaj so dosegle toliko uspeha. Na to vprašanje bomo odgovorili s teoretičnega in praktičnega zornega kota. Opisali bomo matematično ozadje, ki definira Bézierove krivulje in ploskve in opiše njihove lastnosti. Na drugi strani bomo z njimi računalniško modelirali konkretne predmete in druge objekte ter se prepričali v njihovo uporabnost. V ta namen je osrednji del magistrskega dela strukturiran v pet poglavij:

- V drugem poglavju bomo spoznali glavne matematične pojme in koncepte, ki jih bomo potrebovali v naslednjih poglavjih.
- V tretjem poglavju se bomo osredotočili na Bézierove krivulje: kako so definirane, katere so njihove glavne lastnosti in prednosti ter kako jih lahko spreminjamo. V zadnjem podpoglavju pa bomo podali praktičen primer uporabe Bézierovih krivulj s pomočjo računalniškega programa.
- V četrtem poglavju bomo obravnavali Bézierove ploskve. Kot smo storili v prejšnjem poglavju, jih bomo najprej definirali, nato pa našteji in opisali nekaj njihovih glavnih lastnosti. V zadnjem podpoglavju bomo s praktičnim primerom pokazali, kako jih lahko uporabljamo pri računalniškem modeliranju.
- V petem poglavju bo govora o Bézierovih trikotnikih, ki so še vedno primer parametričnih ploskev, le da so od Bézierovih ploskev nekoliko drugače definirane.

- Zadnje poglavje bo posvečeno računalniškemu programu *Blender*, ki omogoča oblikovanje raznoraznih objektov s pomočjo Bézierovih krivulj in ploskev ter drugih orodij. Razdeljeno bo na dve podpoglavji: v prvem bomo podrobno opisali program in njegove glavne funkcije, ukaze in metode; v drugem pa bomo oblikovali avtomobil in pokazali, da so Bézierove krivulje in ploskve uporabne in koristne predvsem pri oblikovanju kompleksnih objektov.

V tretjem, četrtem in petem poglavju si podpoglavja sledijo v podobnem vrstnem redu z namenom, da bi bile bolj razvidne razlike in podobnosti med Bézierovimi krivuljami, ploskvami in trikotniki.

Kot že rečeno, smo za praktični del magistrskega dela uporabili računalniški program Blender (verzija 2.5 Beta), ki je dostopen na spletni strani www.blender.org.

V prilogah *A, B, C, D, E*, ki smo jih dodali na konec magistrskega dela, pa so navedene programske kode, s katerimi smo implementirali nekatere rezultate, ki smo jih matematično opisali v drugem, tretjem in četrtem poglavju. Kode so napisane s programskim jezikom Octave (verzija 3.2.2), ki je dostopen na spletni strani www.gnu.org/software/octave.

Poglavje 2

Osnovni pojmi

V tem poglavju bomo obravnavali osnovne pojme in koncepte, ki nam bodo služili pri razumevanju Bézierovih krivulj in ploskev. Objekte in metode bomo obravnavali v prostoru, v katerem ne bomo definirali določenega koordinatnega sistema, zato bodo definicije objektov neodvisne od izbire le-tega. Zanimal nas bo objekt in njegove lastnosti, ne pa, kako bo vplival na izbrani koordinatni sistem. Primer takega objekta so parametrične krivulje, ki jih bomo obravnavali v podpoglavju 2.3.

2.1 Točke in vektorji

Definicija 1. *Točka* je element tridimenzionalnega evklidskega prostora \mathbb{E}^3 . Označimo jo z malimi krepkimi črkami: $\mathbf{a}, \mathbf{b}, \mathbf{c}$, itd.

Definicija 2. *Vektorji* so elementi linearnega ali vektorskega prostora \mathbb{R}^3 in jih označimo z malimi krepkimi črkami.

Čeprav so tako točke kot vektorji definirani kot trojice realnih števil, moramo biti pazljivi: za poljubni točki \mathbf{a} in \mathbf{b} obstaja *enoličen* vektor \mathbf{v} , ki poteka od točke \mathbf{a} do točke \mathbf{b} :

$$\mathbf{v} = \mathbf{b} - \mathbf{a}, \quad \mathbf{a}, \mathbf{b} \in \mathbb{E}^3, \mathbf{v} \in \mathbb{R}^3.$$

Po drugi strani pa za poljubni vektor \mathbf{v} obstaja *neskončno mnogo* parov točk \mathbf{a} in \mathbf{b} , tako da velja $\mathbf{v} = \mathbf{b} - \mathbf{a}$. Da se v to prepričamo, izberimo poljubni točki \mathbf{a} in \mathbf{b} , da velja $\mathbf{v} = \mathbf{b} - \mathbf{a}$, ter poljubni vektor \mathbf{z} . Potem velja $\mathbf{v} = (\mathbf{b} + \mathbf{z}) - (\mathbf{a} + \mathbf{z}) = \mathbf{b} - \mathbf{a}$.

Opombi:

- Vzporedni premik ne vpliva na vektorje, vpliva pa na točke, saj

$$\mathbf{a} + \mathbf{z} \neq \mathbf{a}, \quad \mathbf{a} \in \mathbb{E}^3, \mathbf{z} \in \mathbb{R}^3;$$

- Točke lahko odštevamo med seboj in dobimo (po definiciji) vektorje, ne moremo pa jih seštevati med seboj (različni koordinatni sistemi bi namreč privedli do različnih rezultatov).

Operacijo seštevanja lahko vseeno apliciramo na točke s t.i. *baricentrično kombinacijo*.

Definicija 3. *Baricentrična kombinacija* je utežena vsota točk $\mathbf{b}_j, j = 0, 1, \dots, n$, kjer se uteži $\alpha_j, j = 0, 1, \dots, n$, seštejejo v ena:

$$\mathbf{b} = \sum_{j=0}^n \alpha_j \mathbf{b}_j; \quad \sum_{j=0}^n \alpha_j = 1, \quad \mathbf{b}_j \in \mathbb{E}^3, \quad \alpha_j \in \mathbb{R}. \quad (2.1)$$

Baricentrična kombinacija, ki iz danih točk generira nove, ni vsota samih točk, ampak je vsota točke z vektorjem, saj, če preoblikujemo (2.1), dobimo:

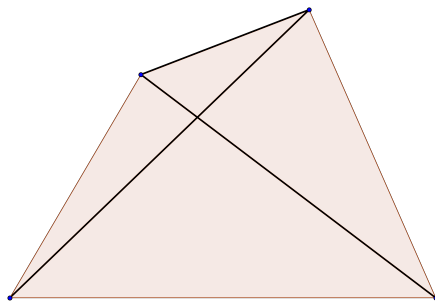
$$\mathbf{b} = \sum_{j=0}^n \alpha_j \mathbf{b}_j = \mathbf{b}_0 + \sum_{j=1}^n \alpha_j (\mathbf{b}_j - \mathbf{b}_0).$$

Če imamo pri baricentričnih kombinacijah še dodatni pogoj, da so koeficienti α_j nenegativni, jih imenujemo **konveksne kombinacije**.

Ker smo tu srečali pojem konveksnosti, definirajmo še naslednja pojma, ki ju bomo večkrat uporabljali pri lastnostih Bézierovih krivulj in ploskev.

Definicija 4. Množica $A \subseteq \mathbb{R}^n$ je **konveksna**, če za poljubni točki $\mathbf{x}, \mathbf{y} \in A$ in vsak $\lambda \in [0, 1]$, $\lambda \in \mathbb{R}$, velja $\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} \in A$.

Definicija 5. **Konveksna ovojnica** množice A je najmanjša konveksna množica, ki vsebuje A oziroma je množica vseh konveksnih kombinacij točk iz A .



Slika 2.1: Obarvani del je konveksna ovojnica narisane poligona.

Podroben opis konveksnosti množic je v [21].

2.2 Afine preslikave

Afine preslikave so pomembno orodje v računalniško podprtem geometrijskem načrtovanju. Uporabljamo jih, če hočemo objekte premikati ali spreminjati njihove velikosti. Definirali jih bomo s pomočjo baricentričnih kombinacij.

Definicija 6. Preslikava $\phi : \mathbb{E}^3 \rightarrow \mathbb{E}^3$ je **afina**, če ohranja baricentrične kombinacije.

Natančneje: če je ϕ afina preslikava in velja

$$\mathbf{b} = \sum_{j=0}^n \alpha_j \mathbf{b}_j, \quad \sum_{j=0}^n \alpha_j = 1, \quad \mathbf{b}_j \in \mathbb{E}^3, \quad \alpha_j \in \mathbb{R},$$

potem velja tudi

$$\phi(\mathbf{b}) = \sum_{j=0}^n \alpha_j \phi(\mathbf{b}_j), \quad \sum_{j=0}^n \alpha_j = 1, \quad \phi(\mathbf{b}_j) \in \mathbb{E}^3, \quad \phi(\alpha_j) \in \mathbb{R}.$$

Trditev 1. Naj bo \mathbf{x} vektor s tremi koordinatami, ki predstavlja točko v danem koordinatnem sistemu. Potem je preslikava

$$\phi(\mathbf{x}) = A\mathbf{x} + \mathbf{v},$$

kjer $A \in \mathbb{R}^{3 \times 3}$ in $\mathbf{v} \in \mathbb{R}^3$, afina.

Dokaz. Najprej moramo vektor \mathbf{x} preoblikovati v baricentrično kombinacijo točk \mathbf{x}_j :

$$\mathbf{x} = \sum_{j=0}^n \alpha_j \mathbf{x}_j.$$

Upoštevajoč enakost $\sum_{j=0}^n \alpha_j = 1$, dobimo

$$\begin{aligned} \phi(\mathbf{x}) &= \phi\left(\sum_{j=0}^n \alpha_j \mathbf{x}_j\right) = A\left(\sum_{j=0}^n \alpha_j \mathbf{x}_j\right) + \mathbf{v} = \sum_{j=0}^n \alpha_j A\mathbf{x}_j + \sum_{j=0}^n \alpha_j \mathbf{v} \\ &= \sum_{j=0}^n \alpha_j (A\mathbf{x}_j + \mathbf{v}) = \sum_{j=0}^n \alpha_j \phi(\mathbf{x}_j). \end{aligned}$$

□

Navedimo sedaj nekaj primerov afinih preslikav pri različnih izbirah matrike A in vektorja \mathbf{v} :

- Če je \mathbf{v} ničelni vektor, t.j. $\mathbf{v} = \mathbf{0}$, in je A identična matrika, t.j. $A = I$, dobimo afino preslikavo

$$\phi(\mathbf{x}) = \mathbf{x}$$

in jo imenujemo **identiteta**;

- Če je $A = I$ in je \mathbf{v} poljuben vektor:

$$\phi(\mathbf{x}) = \mathbf{x} + \mathbf{v},$$

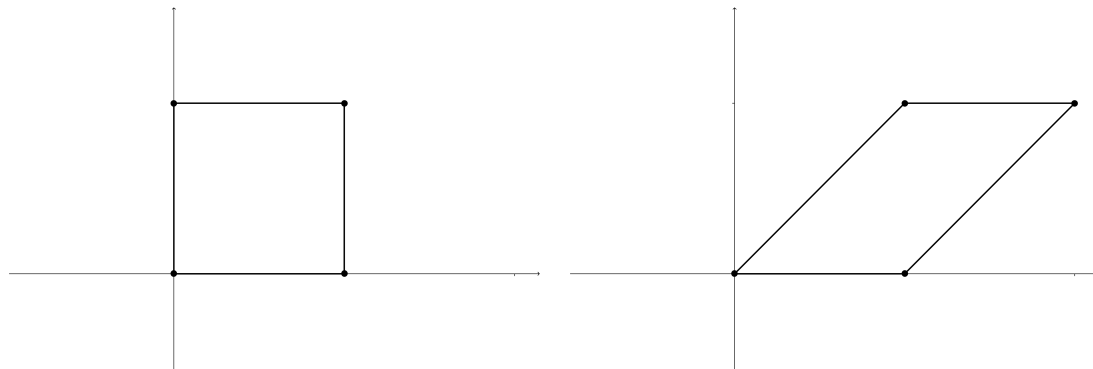
to afino preslikavo imenujemo **translacija**;

- Če je $\mathbf{v} = \mathbf{0}$ in je A diagonalna matrika, imenujemo preslikavo **skaliranje**. Diagonalni elementi matrike A nam povedo, koliko bodo skalirani posamezni elementi vektorja \mathbf{x} ;
- **Rotacijo** dobimo, če je $\mathbf{v} = \mathbf{0}$ in je A ortogonalna matrika;
- **Strig** je primer afine preslikave, ko je $\mathbf{v} = \mathbf{0}$ in je A trikotna matrika, ki ima identitete kot diagonalne bloke.

Zgled 2.1

Naj bo A matrika

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$



Slika 2.2: Primer striga.

Slika 2.2 prikazuje, kako strig pri dani matriki A vpliva na točke $(0, 0)$, $(1, 0)$, $(0, 1)$ in $(1, 1)$ in jih preslika v točke $(0, 0)$, $(1, 0)$, $(1, 1)$ in $(2, 1)$.

2.3 Parametrične krivulje

V nalogi bomo obravnavali krivulje in ploskve, ki so primerne za geometrijsko modeliranje. Potrebujemo take objekte, da jih lahko enostavno rišemo in jih tudi poljubno preoblikujemo. Poleg tega bi radi take objekte, da bodo neodvisni od izbire koordinatnega sistema, kot smo že poudarili v prvem podpoglavju. Krivulje, ki jih bomo tu opisali, imajo vse navedene lastnosti.

Definicija 7. Parametrična krivulja je krivulja, ki jo predstavimo s funkcijami parametra, ki se spreminja vzdolž poljubnega intervala.

Krivuljo $\mathbf{x} : [a, b] \rightarrow \mathbb{E}^3$ predstavimo s tremi funkcijami parametra t :

$$\mathbf{x} = \mathbf{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}, \quad t \in [a, b] \subset \mathbb{R}.$$

Kartezične koordinate x, y in z so odvedljive funkcije v t . V izogib težavam se bomo ukvarjali le s krivuljami \mathbf{x} , katerih odvod \mathbf{x}' je v vsaki točki različen od nič:

$$\mathbf{x}'(t) = \begin{bmatrix} x'(t) \\ y'(t) \\ z'(t) \end{bmatrix} \neq \mathbf{0}, \quad \forall t \in [a, b] \subset \mathbb{R}.$$

Tako parametrizacijo krivulje imenujemo **regularna parametrizacija**.

Pomembna pojma pri parametričnih krivuljah sta *predznačena fleksijska ukrivljenost* in *torzijska ukrivljenost*.

Definicija 8. Naj bo \mathbf{x} regularna ravninska krivulja. **Predznačena fleksijska ukrivljenost** $\kappa(t)$ v točki $\mathbf{x}(t)$ je definirana kot

$$\kappa(t) = \frac{\mathbf{x}'(t) \times \mathbf{x}''(t)}{\|\mathbf{x}'(t)\|^3},$$

kjer je $\|\cdot\|$ Evklidska norma in \times planarni vektorski produkt, t.j. vzamemo le tretjo komponento vektorja $\mathbf{x}'(t) \times \mathbf{x}''(t)$.

Ravninske krivulje imajo samo fleksijsko ukrivljenost, ker je torzijska ukrivljenost enaka nič. Pri prostorskih krivuljah pa imamo definirani obe ukrivljenosti.

Definicija 9. Naj bo \mathbf{x} regularna prostorska krivulja. **Fleksijska ukrivljenost** $\kappa(t)$ v točki $\mathbf{x}(t)$ je definirana kot

$$\kappa(t) = \frac{\|\mathbf{x}'(t) \times \mathbf{x}''(t)\|}{\|\mathbf{x}'(t)\|^3},$$

torzijska ukrivljenost $\tau(t)$ v točki $\mathbf{x}(t)$ pa je definirana kot

$$\tau(t) = \frac{\det[\mathbf{x}'(t), \mathbf{x}''(t), \mathbf{x}'''(t)]}{\|\mathbf{x}'(t) \times \mathbf{x}''(t)\|^2}.$$

Pri tem je \times vektorski produkt, $\|\cdot\|$ Evklidska norma in \det determinanta matrike, ki ima kot stolpce elemente $\mathbf{x}'(t)$, $\mathbf{x}''(t)$ in $\mathbf{x}'''(t)$.

2.4 Linearna interpolacija

Koncept linearne interpolacije bomo povezali s konceptom baricentričnih koordinat in ga kasneje uporabili pri definiciji Bézierove krivulje in Bézierove ploskve.

Definicija 10. Naj bosta \mathbf{a} in \mathbf{b} poljubni točki v \mathbb{E}^3 . **Premica** skozi točki \mathbf{a} in \mathbf{b} je množica točk $\mathbf{x} \in \mathbb{E}^3$, ki so odvisne od parametra t , in so oblike:

$$\mathbf{x} = \mathbf{x}(t) = (1 - t)\mathbf{a} + t\mathbf{b}, \quad t \in \mathbb{R}. \quad (2.2)$$

Opazimo, da dobimo za $t = 0$ točko $\mathbf{x} = \mathbf{a}$, in za $t = 1$ točko $\mathbf{x} = \mathbf{b}$. Ko pa vzamemo $0 < t < 1$, točka \mathbf{x} leži na daljici med \mathbf{a} in \mathbf{b} . Če parameter t zavzame katerokoli drugo vrednost, bo točka \mathbf{x} ležala na premici izven daljice med \mathbf{a} in \mathbf{b} .

Poglejmo sedaj, če nam uspe izraziti točko \mathbf{x} kot baricentrično kombinacijo točk \mathbf{a} in \mathbf{b} v \mathbb{E}^3 , in točko t kot baricentrično kombinacijo točk 0 in 1 v \mathbb{E}^1 . Kot odgovor na prvo vprašanje imamo enakost (2.2), točko t pa lahko izrazimo kot

$$t = (1 - t) \cdot 0 + t \cdot 1.$$

Zanimivo je, da imamo v obeh primerih iste uteži, za katere seveda velja: $(1 - t) + t = 1$. Po definiciji afinih preslikav so točke \mathbf{x} , \mathbf{a} , \mathbf{b} v \mathbb{E}^3 afina preslikava točk $t, 0, 1$ v \mathbb{E}^1 . To pa nas privede do definicije linearne interpolacije.

Definicija 11. **Linearna interpolacija** je afina preslikava, ki preslika realno premico v poljubno premico v \mathbb{E}^3 .

Koncept linearne interpolacije je močno povezan s pojmom *baricentričnih koordinat*. Naj bodo \mathbf{x} , \mathbf{a} in \mathbf{b} tri kolinearne točke v \mathbb{E}^3 in naj bo točka \mathbf{x} izražena z ostalima dvema točkama:

$$\mathbf{x} = \alpha\mathbf{a} + \beta\mathbf{b}, \quad \alpha + \beta = 1.$$

Potem sta α in β *baricentrični koordinati* točke \mathbf{x} glede na \mathbf{a} in \mathbf{b} . Vidimo, da je točka \mathbf{x} baricentrična kombinacija \mathbf{a} in \mathbf{b} .

Povezava med baricentričnimi koordinatami in linearno interpolacijo je dokaj očitna:

$$\alpha = 1 - t, \quad \beta = t.$$

Če je $0 \leq t \leq 1$, potem so baricentrične koordinate nenegativne. Sicer so lahko vrednosti α ali β negativne.

Definirajmo sedaj baricentrične koordinate s pomočjo predznačene razdalje med dvema točkama. Označimo jo z *razd.* Vzemimo spet tri kolinearne točke \mathbf{a} , \mathbf{x} in \mathbf{b} . Baricentrične koordinate točke \mathbf{x} so potem definirane kot:

$$\alpha = \frac{\text{razd}(\mathbf{x}, \mathbf{b})}{\text{razd}(\mathbf{a}, \mathbf{b})}, \quad \beta = \frac{\text{razd}(\mathbf{a}, \mathbf{x})}{\text{razd}(\mathbf{a}, \mathbf{b})}.$$

Ni se težko prepričati, da res velja $\alpha + \beta = 1$.

Koncept baricentričnih koordinat lahko povežemo s konceptom *razmerja*.

Definicija 12. *Razmerje med tremi kolinearnimi točkami \mathbf{a} , \mathbf{x} in \mathbf{b} , je definirano kot:*

$$\text{razmerje}(\mathbf{a}, \mathbf{x}, \mathbf{b}) = \frac{\text{razd}(\mathbf{a}, \mathbf{x})}{\text{razd}(\mathbf{x}, \mathbf{b})}.$$

Iz tega sledi:

$$\text{razmerje}(\mathbf{a}, \mathbf{x}, \mathbf{b}) = \frac{\text{razd}(\mathbf{a}, \mathbf{x})}{\text{razd}(\mathbf{x}, \mathbf{b})} = \frac{\text{razd}(\mathbf{a}, \mathbf{x})}{\text{razd}(\mathbf{a}, \mathbf{b})} \cdot \frac{\text{razd}(\mathbf{a}, \mathbf{b})}{\text{razd}(\mathbf{x}, \mathbf{b})} = \frac{\beta}{\alpha}. \quad (2.3)$$

Pomembna lastnost baricentričnih koordinat poljubne točke je, da jih afine preslikave ohranjajo. To pomeni, da bo ulomek $\frac{\beta}{\alpha}$ ostal nespremenjen. Torej, če je ϕ afina preslikava in velja (2.3), potem velja tudi:

$$\text{razmerje}(\phi(\mathbf{a}), \phi(\mathbf{x}), \phi(\mathbf{b})) = \frac{\beta}{\alpha}. \quad (2.4)$$

Enakost (2.4) nam pove, da afine preslikave ohranjajo razmerja. Pri definiciji linearne interpolacije smo poljubno daljico v \mathbb{E}^3 dobili, če smo afino preslikali interval $[0, 1]$. Ni pa nujno, da je daljica slika tega intervala. Če vzamemo $t \in [0, 1]$ in $u \in [a, b]$ ter definiramo $t = (u - a)/(b - a)$, točko na daljici dobimo kot

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

kar postane

$$\mathbf{x}(u) = \frac{b - u}{b - a}\mathbf{a} + \frac{u - a}{b - a}\mathbf{b}.$$

Uteži so sprejemljive, saj velja $\frac{b-u}{b-a} + \frac{u-a}{b-a} = 1$. Torej smo poljubno daljico v \mathbb{E}^3 dobili kot sliko poljubnega intervala $[a, b]$.

Ker so a, u, b in $0, t, 1$ v istem razmerju kot $\mathbf{a}, \mathbf{x}, \mathbf{b}$, smo prišli do ugotovitve, da afine transformacije ohranjajo linearno interpolacijo.

2.5 Baricentrične koordinate v ravnini

Baricentrične koordinate niso definirane samo na premicah, kot smo jih spoznali v prejšnjem razdelku, ampak jih lahko definiramo tudi v ravnini, kot jih je uvedel F. Moebius leta 1827. Za več informacij glej [14].

Definicija 13. Naj bo dan trikotnik s tremi oglišči $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{E}^2$. Vzemimo poljubno četrto točko $\mathbf{p} \in \mathbb{E}^2$. Če zapišemo

$$\mathbf{p} = u\mathbf{a} + v\mathbf{b} + w\mathbf{c}, \quad u + v + w = 1, \quad (2.5)$$

potem uteži u, v in w imenujemo **baricentrične koordinate** točke \mathbf{p} glede na točke $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

Četrto točko lahko torej izrazimo kot baricentrično kombinacijo ostalih treh.

Označimo predznačeno ploščino trikotnika z oglišči $\mathbf{a}, \mathbf{b}, \mathbf{c}$ s $pl(\mathbf{a}, \mathbf{b}, \mathbf{c})$.

Trditev 2. Če poznamo koordinate točk $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{p} \in \mathbb{E}^2$, potem lahko izračunamo baricentrične koordinate u, v in w točke \mathbf{p} glede na preostale tri na sledeči način:

$$u = \frac{pl(\mathbf{p}, \mathbf{b}, \mathbf{c})}{pl(\mathbf{a}, \mathbf{b}, \mathbf{c})}, \quad v = \frac{pl(\mathbf{a}, \mathbf{p}, \mathbf{c})}{pl(\mathbf{a}, \mathbf{b}, \mathbf{c})}, \quad w = \frac{pl(\mathbf{a}, \mathbf{b}, \mathbf{p})}{pl(\mathbf{a}, \mathbf{b}, \mathbf{c})}.$$

Dokaz. Vemo, da lahko ploščino trikotnika v koordinatnem sistemu izračunamo s pomočjo determinant. Naj bodo $\mathbf{a}, \mathbf{b}, \mathbf{c}$ oglišča trikotnika in naj bodo $(x_a, y_a), (x_b, y_b), (x_c, y_c)$ njihove x, y - koordinate. Potem je ploščina trikotnika enaka:

$$pl(a, b, c) = \frac{1}{2} \begin{vmatrix} x_a & x_b & x_c \\ y_a & y_b & y_c \\ 1 & 1 & 1 \end{vmatrix} \quad (2.6)$$

Za dokončanje dokaza bomo potrebovali še *Cramerjevo pravilo*.

Cramerjevo pravilo uporabljamo za reševanje sistemov n linearnih enačb z n neznankami. Zapišemo sistem v matrični obliki:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

oziroma $Ax = b$, kjer je A matrika koeficientov v sistemu enačb, x vektor iskanih spremenljivk in b vektor, ki vsebuje vrednosti na desni strani enačajev v sistemu. Pravilo pravi, da dobimo rešitve x_i na sledeči način:

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad i = 1, 2, \dots, n,$$

kjer je A_i matrika, ki jo dobimo, če i -ti stolpec matrike A nadomestimo s stolpcem b .

Če sistem

$$\begin{aligned} u\mathbf{a} + v\mathbf{b} + w\mathbf{c} &= \mathbf{p} \\ u + v + w &= 1 \end{aligned}$$

preoblikujemo v matrično obliko, dobimo

$$\begin{pmatrix} x_a & x_b & x_c \\ y_a & y_b & y_c \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}.$$

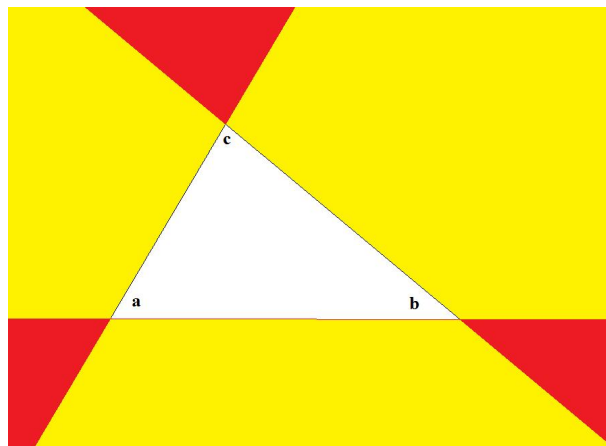
Uporabimo Cramerjevo pravilo in enakost (2.6), da dobimo neznanke u :

$$u = \frac{\begin{vmatrix} x_p & x_b & x_c \\ y_p & y_b & y_c \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_a & x_b & x_c \\ y_a & y_b & y_c \\ 1 & 1 & 1 \end{vmatrix}} = \frac{2 \cdot pl(\mathbf{p}, \mathbf{b}, \mathbf{c})}{2 \cdot pl(\mathbf{a}, \mathbf{b}, \mathbf{c})} = \frac{pl(\mathbf{p}, \mathbf{b}, \mathbf{c})}{pl(\mathbf{a}, \mathbf{b}, \mathbf{c})}.$$

Podobno dobimo še neznanke v in w .

□

Opomba: Če so vse tri baricentrične koordinate u, v, w pozitivne, potem bo točka \mathbf{p} ležala znotraj trikotnika z oglišči $\mathbf{a}, \mathbf{b}, \mathbf{c}$. Če pa je katera izmed koordinat negativna, potem se bo točka \mathbf{p} nahajala izven trikotnika.



Slika 2.3: Vpliv baricentričnih koordinat u, v, w v trikotniku.

Na sliki 2.3 beli pobarvan trikotnik predstavlja točke, ki jih dobimo, če so vse tri koordinate u, v, w pozitivne. Če imamo samo eno koordinato negativno in ostali dve pozitivni, dobimo točke v rumenem delu slike. Točke v rdečem delu pa dobimo, če sta dve koordinati negativni, preostala pa pozitivna. Ker velja (2.5), ne morejo biti vse tri koordinate hkrati negativne.

Poglavje 3

Bézierove krivulje

Matematiki so v prejšnjem stoletju iskali takšne krivulje, da bi jih lahko učinkovito spreminjali in z njimi načrtovali raznorazne objekte. Leta 1959 je francoska avtomobilska hiša Citroën zaposlila mladega matematika, da bi rešil nekaj teoretskih problemov. Ko je P. de Casteljau raziskoval krivulje, je odkril popolnoma nov pristop risanja le-teh. Definiral jih ni več s pomočjo točk, ki so ležale na njej, ampak s pomočjo t.i. *kontrolnega poligona*. Če je želel preoblikovati krivuljo, ni več spremenil točke na krivulji, ampak točke na kontrolnem poligonu. Razvil je učinkovit, hiter, numerično stabilen in geometrijsko nazoren algoritem, ki izračuna točke na krivulji. Prvič so algoritem predstavili leta 1971, čeprav ga je de Casteljau razvil že prej, ampak Citroën ni nikoli objavil njegovih rezultatov.

De Casteljau ni bil edini, ki se je ukvarjal z risanjem in oblikovanjem učinkovitih krivulj. V avtomobilski hiši Renault je P. Bézier definiral enake krivulje, kot jih je odkril de Casteljau. Rezultat je bil enak, čeprav je Bézier uporabil druge matematične koncepte. Bézierovo delo je bilo objavljeno in dobro sprejeto. Kasneje so krivulje dobile ime po Bézierju, čeprav so matematiki navdušeno sprejeli tako Bézierovo kot de Casteljauovo delo.

3.1 de Casteljauov algoritem

Bézierove krivulje lahko predstavimo na različne načine. Tu bomo predstavili algoritem, ki si ga je zamislil P. de Casteljau, ko je bil zaposlen v avtomobilski hiši Citroën.

de Casteljauov algoritem

Naj bodo $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{E}^3$ dane točke in naj bo $t \in \mathbb{R}$ parameter. Točke $\mathbf{b}_i, i = 0, 1, \dots, n$, imenujemo *kontrolne točke* ali *Bézierove točke* in skupaj tvorijo *kontrolni poligon* ali *Bézierov poligon* krivulje. Nato izračunamo točke:

$$\mathbf{b}_i^r(t) = (1 - t) \mathbf{b}_i^{r-1}(t) + t \mathbf{b}_{i+1}^{r-1}(t), \quad (3.1)$$

kjer je

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

Velja $\mathbf{b}_i^0(t) := \mathbf{b}_i$. To pomeni, da, če hočemo izračunati točke $\mathbf{b}_i^1(t)$, jih bomo dobili s pomočjo točk $\mathbf{b}_i^0(t)$, ki so ravno kontrolne točke. Točko na Bézierovi krivulji \mathbf{b}^n pri danem parametru t , t.j. $\mathbf{b}_0^n(t)$, bomo dobili tako, da bomo izračunali vse vmesne točke pri istem

parametru, dokler ne bomo izračunali še zaželjene točke.

Zato, da bolje predstavimo algoritem, lahko dobljene točke pri parametru t zapišemo v obliko "trikotnika" in dobimo t.i. *de Casteljauovo shemo*. Podajmo shemo za $n = 4$:

$$\begin{array}{cccccc} \mathbf{b}_0^0 & = & \mathbf{b}_0 & & & \\ \mathbf{b}_1^0 & = & \mathbf{b}_1 & \mathbf{b}_0^1 & & \\ \mathbf{b}_2^0 & = & \mathbf{b}_2 & \mathbf{b}_1^1 & \mathbf{b}_0^2 & \\ \mathbf{b}_3^0 & = & \mathbf{b}_3 & \mathbf{b}_2^1 & \mathbf{b}_1^2 & \mathbf{b}_0^3 & \\ \mathbf{b}_4^0 & = & \mathbf{b}_4 & \mathbf{b}_3^1 & \mathbf{b}_2^2 & \mathbf{b}_1^3 & \mathbf{b}_0^4 \end{array}$$

Vidimo, da de Casteljauovo shemo sestavimo tako, da v prvi stolpec zapišemo vse kontrolne točke, nato v drugi stolpec vstavimo točke za $r = 1$, v tretji točke za $r = 2$, itd. Kot smo že povedali, je zadnja točka \mathbf{b}_0^4 točka na Bézierovi krivulji za dan parameter t .

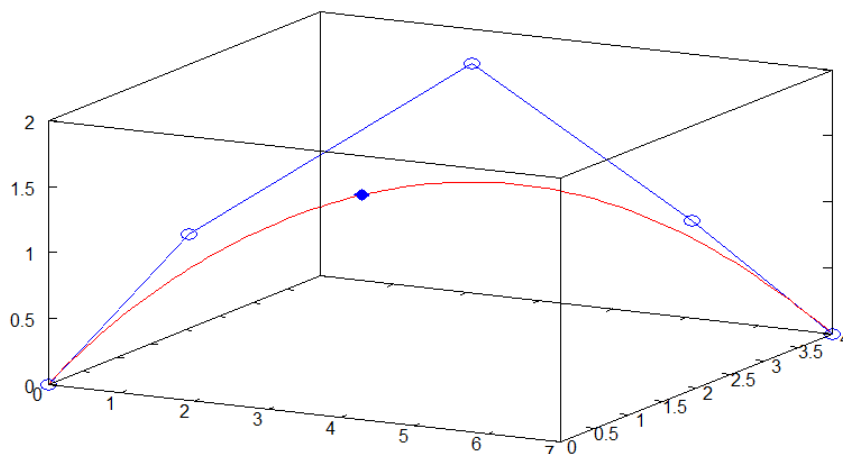
Zgled 3.1

Naj bodo $\mathbf{b}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{b}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, $\mathbf{b}_2 = \begin{bmatrix} 3 \\ 3 \\ 2 \end{bmatrix}$, $\mathbf{b}_3 = \begin{bmatrix} 6 \\ 3 \\ 1 \end{bmatrix}$, $\mathbf{b}_4 = \begin{bmatrix} 7 \\ 4 \\ 0 \end{bmatrix}$ kontrolne točke Bézierove krivulje \mathbf{b}^4 . Izračunajmo točko na Bézierovi krivulji pri parametru $t = 0.4$. De Causteljauova shema se glasi:

$$\begin{array}{cccccc} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & & & & & \\ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0.4 \\ 0.4 \\ 0.4 \end{bmatrix} & & & & \\ \begin{bmatrix} 3 \\ 3 \\ 2 \end{bmatrix} & \begin{bmatrix} 1.8 \\ 1.8 \\ 1.4 \end{bmatrix} & \begin{bmatrix} 0.96 \\ 0.96 \\ 0.96 \end{bmatrix} & & & \\ \begin{bmatrix} 6 \\ 3 \\ 1 \end{bmatrix} & \begin{bmatrix} 4.2 \\ 3 \\ 1.6 \end{bmatrix} & \begin{bmatrix} 2.76 \\ 2.28 \\ 1.48 \end{bmatrix} & \begin{bmatrix} 1.68 \\ 1.49 \\ 1.07 \end{bmatrix} & & \\ \begin{bmatrix} 7 \\ 4 \\ 0 \end{bmatrix} & \begin{bmatrix} 6.4 \\ 3.4 \\ 0.6 \end{bmatrix} & \begin{bmatrix} 5.08 \\ 3.16 \\ 1.2 \end{bmatrix} & \begin{bmatrix} 3.69 \\ 2.63 \\ 1.37 \end{bmatrix} & \begin{bmatrix} 2.48 \\ 1.95 \\ 1.19 \end{bmatrix} & . \end{array}$$

Zadnja obarvana točka v shemi je dobljena točka na Bézierovi krivulji. Na sliki 3.1 je z modro barvo narisani kontrolni poligon s kontrolnimi točkami $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4$; v rdeči barvi pa je narisana Bézierova krivulja. Točka na Bézierovi krivulji, ki smo jo v shemi dobili za $t = 0.4$, je na sliki označena z modrim kvadratom.

Za podrobne informacije glede implementacije de Casteljauovega algoritma glej prilogo A.

Slika 3.1: Bézierova krivulja \mathbf{b}^4 in njen kontrolni poligon.

3.2 Bernsteinova oblika Bézierove krivulje

Bézierove krivulje lahko definiramo tudi na drugačen način, brez uporabe de Casteljauovega algoritma. Najprej definirajmo Bernsteinove polinome, kasneje pa bomo zapisali, kako jih lahko uporabimo, da eksplicitno izračunamo Bézierovo krivuljo. Tako definirano Bézierovo krivuljo je opisal A.R. Forrester, ko se je začel zanimati za Bézierovo delo. De Casteljau je že pred njim prišel do istega rezultata, čeprav, kot smo že povedali, ni objavil svojih rezultatov.

Definicija 14. *Bernsteinovi polinomi so polinomi oblike*

$$B_i^n(t) := \binom{n}{i} t^i (1-t)^{n-i}, \quad \binom{n}{i} := \begin{cases} \frac{n!}{i!(n-i)!} & 0 \leq i \leq n, \\ 0 & \text{sicer.} \end{cases}$$

Bézierove krivulje lahko potem zapišemo kot ([5]):

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

V naslednjih podpoglavjih bomo pomembne operacije z Bézierovimi krivuljami opisali s pomočjo Bernsteinovih polinomov, zato bom spodaj navedel nekaj njihovih glavnih lastnosti in jih dokazal. Prva dva dokaza sta vzeta iz [5].

1. Bernsteinove polinome lahko izračunamo rekurzivno:

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

Dokaz. Vemo, da velja:

- $B_0^0(t) = 1$,
- $B_i^n(t) = 0$, če $i \notin \{0, \dots, n\}$ in
- $\binom{n-1}{i} + \binom{n-1}{i-1} = \binom{n}{i}$.

Potem lahko $B_i^n(t)$ zapišemo kot:

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

□

2. Bernsteinovi polinomi tvorijo particijo enote:

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

Dokaz. Uporabimo enakost $1 = t + (1-t)$ in binomski izrek:

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

□

3. Odvod Bernsteinovega polinoma se glasi

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

Dokaz.

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

□

4. Bernsteinov polinom $B_i^n(t)$ ima natanko en maksimum, in sicer pri $t = \frac{i}{n}$.

Dokaz. Poiščimo najprej stacionarne točke:

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

Izračunajmo še drugi odvod in vstavimo $t = \frac{i}{n}$:

$$\begin{aligned} \frac{d^2}{dt^2} B_i^n \left(\frac{i}{n} \right) &= \frac{d}{dt} \left(n \cdot \left(B_{i-1}^{n-1} \left(\frac{i}{n} \right) - B_i^{n-1} \left(\frac{i}{n} \right) \right) \right) \\ &= n \cdot (n-1) \cdot \left(B_{i-2}^{n-2} \left(\frac{i}{n} \right) - 2 \cdot B_{i-1}^{n-2} \left(\frac{i}{n} \right) + B_i^{n-2} \left(\frac{i}{n} \right) \right). \end{aligned}$$

Če bi rezultat še poenostavili in analizirali, bi videli, da bi bila končna vrednost negativna, kar pomeni, da imamo pri $t = \frac{i}{n}$ lokalni maksimum.

□

3.3 Pomembne lastnosti Bézierovih krivulj

V tem podpoglavju bomo spoznali nekaj lastnosti Bézierovih krivulj in navedli nekaj primerov, kdaj in kako so lahko uporabne.

1. Bézierove krivulje so invariantne za affine preslikave.

Dejstvo sledi iz tega, da so tudi baricentrične koordinate invariantne za affine preslikave in isto velja za linearno interpolacijo. Ker je de Casteljauov algoritem zaporedje linearnih interpolacij, invariantnost drži tudi za krivulje.

Naj bo ϕ afina preslikava. Imamo torej dve možnosti:

- izračunamo točko na Bézierovi krivulji \mathbf{b}^n in jo afino preslikamo: $\phi(\mathbf{b}^n)$;
- afino preslikamo kontrolni poligon, torej izračunamo $\phi(\mathbf{b}_0), \phi(\mathbf{b}_1), \dots, \phi(\mathbf{b}_n)$, potem pa izračunamo točko na tako dobljeni Bézierovi krivulji.

Zakaj je pravzaprav pomembna ta lastnost? Pomislimo, da affine preslikave uporabljamo, ko hočemo objekt skalirati, rotirati ipd. Če bi uporabili prvo metodo, potem bi morali izračunati vse točke na Bézierovi krivulji in nato vse točke preslikati. Če pa bi uporabili drugo metodo, bi bilo treba le preslikati kontrolne točke in potem izračunati točke na novi krivulji. V drugem primeru bi imeli bistveno manj računskih operacij.

2. Bézierova krivulja leži v konveksni ovojnici svojega kontrolnega poligona.

Ko rišemo Bézierovo krivuljo, izberemo parameter $t \in [0, 1]$. V tem primeru krivulja \mathbf{b}^n leži v konveksni ovojnici svojega kontrolnega poligona, saj dobimo točke $\mathbf{b}_i^r(t)$ iz (3.1) kot konveksno kombinacijo prejšnjih. To pomeni, da bodo nove točke iz (3.1) ležale na daljicah med dvema že obstoječima točkama in bo torej končna točka, tj. točka na Bézierovi krivulji, tudi ležala v konveksni ovojnici kontrolnega poligona. To velja za vse točke na krivulji in torej za celotno krivuljo. Če $t \notin [0, 1]$, potem ta lastnost ne velja več.

S to lastnostjo si lahko pomagamo, če želimo preveriti, ali se bosta dve telesi pri gibanju dotaknili. Predstavimo tire gibanja z Bézierovimi krivuljami. Okrog vsake množice kontrolnih točk zarišemo kvader. Imenujemo ga *minimax škatla*. Minimax škatla vsebuje kontrolni poligon in torej tudi krivuljo. Če se škatli ne dotikata, kar lahko zelo hitro preverimo, potem se gotovo tudi Bézierovi krivulji ne dotikata. Če pa se škatli dotikata, se lahko krivulji dotikata, ali pa ne.

3. Interval parametra lahko poljubno spremenimo.

Običajno za parameter t uporabljamo interval $[0, 1]$, ker nam to poenostavi računanje. Če pa bi želeli, bi lahko imeli parameter u iz intervala $[a, b]$:

$$t = \frac{u - a}{b - a}, \quad a \leq u \leq b,$$

in bi se vmesni korak de Casteljauovega algoritma glasil:

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

V podpoglavju 2.4 smo pokazali, da je preslikanje intervala $[0, 1]$ na poljubni interval $[a, b]$ afina preslikava. Torej lahko interval parametra poljubno izberemo.

4. Učinkovito risanje z Bézierovimi krivuljami.

Bézier in de Casteljau sta raziskovala to vrsto parametričnih krivulj in ploskev, ker sta se zavedala, da jih lahko učinkovito spreminjamo, da dosežemo zaželeno obliko. Ko želimo doseči določeno obliko Bézierove krivulje, lahko startamo s približkom, in potem spremenimo kontrolni poligon tako, da bo definiral vedno isto krivuljo, vendar bo ta bolj fleksibilna in jo bomo lahko učinkoviteje spreminjali. Več o tem v naslednjem podpoglavju.

3.4 Višanje stopnje

Recimo, da večkrat spreminjamo lego točk kontrolnega poligona, ker hočemo doseči željeno obliko Bézierove krivulje. Ker pa nam ne uspe, bi želeli imeti bolj fleksibilno krivuljo. Problem lahko rešimo tako, da uvedemo novo kontrolno točko, ne da bi spremenili oblike naše krivulje. Na ta način navidezno zvišamo stopnjo krivulje, ki bo imela večjo fleksibilnost.

Vsekakor je višanje stopnje pomembno orodje tudi pri drugih problemih. Ko imamo opravka z algoritmi, ki generirajo ploskve s pomočjo danih Bézierovih krivulj, potem je pomembno, da so vse krivulje iste stopnje. V tem primeru lahko višanje stopnje priskoči na pomoč.

Postopek, s katerim pridemo do uvedbe nove kontrolne točke, je opisan v [16], in si ga podrobneje oglejmo.

Uporabili bomo Bézierove krivulje definirane z Bernsteinovi polinomi. Naj bo Bézierova krivulja podana z $n+1$ kontrolnimi točkami $\mathbf{b}_i, i = 0, 1, \dots, n$. Naj bodo $\mathbf{b}_i^1, i = 0, 1, \dots, n+1$, točke novega iskanega kontrolnega poligona.

Kot smo že povedali, hočemo, da sta Bézierovi krivulji obeh poligonov enaki:

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

oziroma

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

Radi bi imeli na obeh straneh element $t^i(1-t)^{n+1-i}$, ker bi lahko potem primerjali koeficiente na obeh straneh enačbe. Zato, da to dosežemo, moramo levo stran enačbe pomnožiti z $1 = t + (1-t)$. Dobimo

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

Izpostavimo $\mathbf{b}_i \binom{n}{i}$:

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

Levo stran poenostavimo in pogledamo koeficiente pri $t^i(1-t)^{n+1-i}$ na obeh straneh:

$$\mathbf{b}_i \binom{n}{i} + \mathbf{b}_{i-1} \binom{n}{i-1} = \mathbf{b}_i^1 \binom{n+1}{i}.$$

Obe strani delimo z $\binom{n+1}{i}$ in, ker velja:

$$\frac{\binom{n}{i}}{\binom{n+1}{i}} = 1 - \frac{i}{n+1}, \quad \frac{\binom{n}{i-1}}{\binom{n+1}{i}} = \frac{i}{n+1},$$

zaključimo, da je

$$\mathbf{b}_i^1 = \frac{i}{n+1} \mathbf{b}_{i-1} + \left(1 - \frac{i}{n+1}\right) \mathbf{b}_i, \quad i = 0, 1, \dots, n+1. \quad (3.2)$$

Vidimo, da parametra $\frac{i}{n+1}$ in $\left(1 - \frac{i}{n+1}\right)$ pripadata intervalu $[0, 1]$. Po definiciji linearne interpolacije pomeni, da bodo nove točke \mathbf{b}_i^1 ležale na daljicah med \mathbf{b}_{i-1} in \mathbf{b}_i , torej na stranicah originalnega kontrolnega poligona.

Iz (3.2) izračunamo, da se bosta prva in zadnja točka originalnega in novega kontrolnega poligona ujemali:

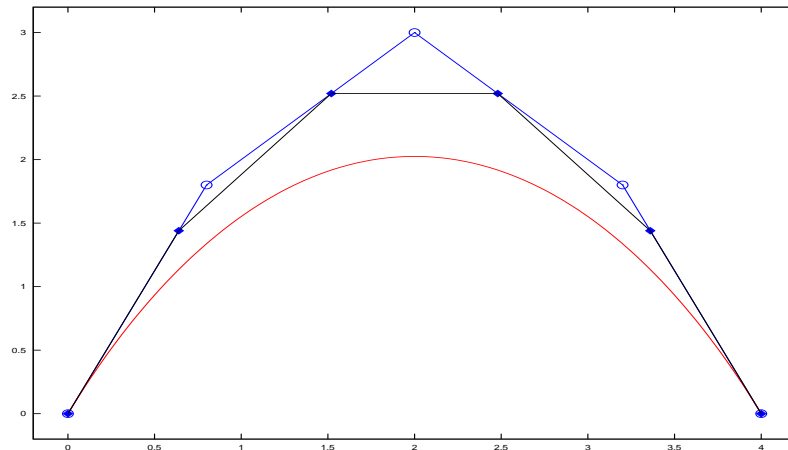
$$\mathbf{b}_0^1 = \mathbf{b}_0, \quad \mathbf{b}_{n+1}^1 = \mathbf{b}_n.$$

Iz teh ugotovitev smo se prepričali, da bo nov kontrolni poligon ležal znotraj konveksne ovojnice originalnega poligona.

Zgled 3.2

Na sliki 3.2 so originalne Bézierove točke $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4$ predstavljene s krogci. Ko zvišamo stopnjo krivulje za ena, dobimo točke $\mathbf{b}_0^1, \mathbf{b}_1^1, \mathbf{b}_2^1, \mathbf{b}_3^1, \mathbf{b}_4^1, \mathbf{b}_5^1$ novega kontrolnega poligona, ki so predstavljene s kvadratici. Vidimo, da je Bézierova krivulja pri obeh poligonih enaka.

A. R. Forrest je prvi domneval in kasneje je G. Farin dokazal, da kontrolni poligoni z višanjem stopnje konvergirajo k Bézierovi krivulji. Ta lastnost ni zelo uporabna, ker je red konvergence zelo počasen. Vendar tega ne bomo dokazovali, ker ni glavni cilj naloge. Več informacij dobite v [6] in [10].



Slika 3.2: Primerjava med originalnim polinomom in novim polinomom pri višanju stopnje krivulje.

3.5 Odvodi

Odvode obravnavamo zato, ker igrajo pomembno vlogo pri študiju zveznosti in zlepkih Bézierovih krivulj. Pomagajo nam definirati, kdaj so krivulje “gladke”.

Odvod Bézierove krivulje izračunamo s pomočjo Bernsteinovih polinomov:

$$\frac{d}{dt}\mathbf{b}^n(t) = \frac{d}{dt} \sum_{i=0}^n \mathbf{b}_i B_i^n(t). \quad (3.3)$$

Ker odvajamo po spremenljivki t , točke \mathbf{b}_i ne vplivajo na odvod. To pomeni, da se naša naloga reducira na odvajanje Bernsteinovih polinomov. V prejšnjem podpoglavju smo dokazali, da je odvod Bernsteinovega polinoma enak

$$\frac{d}{dt}B_i^n(t) = n \cdot (B_{i-1}^{n-1}(t) - B_i^{n-1}(t)). \quad (3.4)$$

Če (3.4) vstavimo v (3.3), dobimo:

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

Vsoto razlike zapišemo kot razliko vsot in pri tem pazimo na indekse:

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

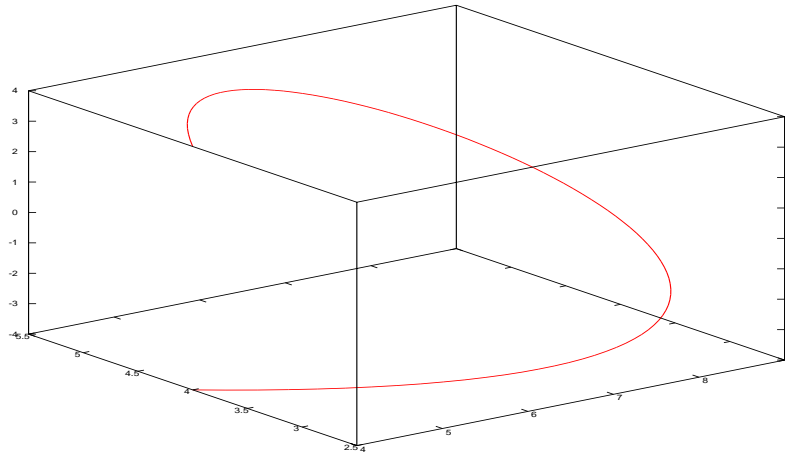
Če uvedemo preme končne difference $\Delta \mathbf{b}_i := \mathbf{b}_{i+1} - \mathbf{b}_i$, se odvod Bézierove krivulje glasi:

$$\frac{d}{dt} \mathbf{b}^n(t) = n \cdot \sum_{i=0}^{n-1} \Delta \mathbf{b}_i B_i^{n-1}(t), \quad \Delta \mathbf{b}_i \in \mathbb{R}^3. \quad (3.5)$$

Opomba: Diference točk $\Delta \mathbf{b}_i$ tvorijo kontrolni poligon, ki bo definiral krivuljo $\frac{d}{dt} \mathbf{b}^n(t)$. Vendar moramo biti pazljivi, saj $\Delta \mathbf{b}_i$ niso točke, ampak so po definiciji vektorji (glej podglavje 2.1). Zato izberemo poljubno točko \mathbf{c} in sestavimo kontrolni poligon s točkami $\mathbf{c} + \Delta \mathbf{b}_0, \mathbf{c} + \Delta \mathbf{b}_1, \dots, \mathbf{c} + \Delta \mathbf{b}_{n-1}$. Običajno izberemo točko $\mathbf{c} = \mathbf{0}$, saj imamo potem poenostavljeno računanje.

Zgled 3.3

Naj bodo $\mathbf{b}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{b}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, $\mathbf{b}_2 = \begin{bmatrix} 3 \\ 3 \\ 2 \end{bmatrix}$, $\mathbf{b}_3 = \begin{bmatrix} 6 \\ 3 \\ 1 \end{bmatrix}$, $\mathbf{b}_4 = \begin{bmatrix} 7 \\ 4 \\ 0 \end{bmatrix}$ kontrolne točke iz zgleda 3.1. Slika 3.3 predstavlja prvi odvod dane Bézierove krivulje.



Slika 3.3: Odvod Bézierove krivulje.

Poiščimo sedaj še višje odvode Bézierove krivulje.

Začnimo z drugim odvodom:

$$\begin{aligned} \frac{d^2}{dt^2} \mathbf{b}^n(t) &= \frac{d}{dt} \left(\frac{d}{dt} \mathbf{b}^n(t) \right) = \frac{d}{dt} \left(n \cdot \sum_{i=0}^{n-1} \Delta \mathbf{b}_i B_i^{n-1}(t) \right) \\ &= n \cdot (n-1) \cdot ((\mathbf{b}_1 - \mathbf{b}_0) \cdot B_{-1}^{n-2}(t) - (\mathbf{b}_1 - \mathbf{b}_0) \cdot B_0^{n-2}(t) + \\ &\quad \cdots + (\mathbf{b}_n - \mathbf{b}_{n-1}) \cdot B_{n-2}^{n-2}(t) - (\mathbf{b}_n - \mathbf{b}_{n-1}) \cdot B_{n-1}^{n-2}(t)) \\ &= n \cdot (n-1) \cdot (B_0^{n-2}(t)(\mathbf{b}_2 - 2\mathbf{b}_1 + \mathbf{b}_0) + \cdots + B_{n-2}^{n-2}(t)(\mathbf{b}_n - 2\mathbf{b}_{n-1} + \mathbf{b}_{n-2})) \\ &= n \cdot (n-1) \cdot \left(\sum_{i=0}^{n-2} B_i^{n-2}(t)(\mathbf{b}_{i+2} - 2\mathbf{b}_{i+1} + \mathbf{b}_i) \right). \end{aligned}$$

Tretji odvod bi po podobnih računih bil enak

$$\frac{d^3}{dt^3} \mathbf{b}^n(t) = n \cdot (n-1) \cdot (n-2) \cdot \left(\sum_{i=0}^{n-3} B_i^{n-3}(t) (\mathbf{b}_{i+3} - 3\mathbf{b}_{i+2} + 3\mathbf{b}_{i+1} - \mathbf{b}_i) \right).$$

Definirajmo sedaj premo končno diferenco

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

Potem je r -ti odvod Bézierove krivulje definiran kot

$$\frac{d^r}{dt^r} \mathbf{b}^n(t) = \frac{n!}{(n-r)!} \sum_{i=0}^{n-r} \Delta^r \mathbf{b}_i \cdot B_i^{n-r}(t). \quad (3.6)$$

Formalni dokaz (3.6) bomo izpustili, je pa dostopen v [5].

Poglejmo si, kaj se dogaja v robnih točkah:

$$\begin{aligned} \frac{d^r}{dt^r} \mathbf{b}^n(0) &= \frac{n!}{(n-r)!} \Delta^r \mathbf{b}_0, \\ \frac{d^r}{dt^r} \mathbf{b}^n(1) &= \frac{n!}{(n-r)!} \Delta^r \mathbf{b}_{n-r}. \end{aligned}$$

V robni točki pri $t = 0$ je r -ti odvod Bézierove krivulje odvisen le od kontrolnih točk $\mathbf{b}_0, \dots, \mathbf{b}_r$, oziroma pri $t = 1$ le od točk $\mathbf{b}_n, \dots, \mathbf{b}_{n-r}$.

Pri $r = 0$ dobimo lastnost interpolacije v robnih točkah:

$$\frac{d^0}{dt^0} \mathbf{b}^n(0) = \mathbf{b}_0, \quad \frac{d^0}{dt^0} \mathbf{b}^n(1) = \mathbf{b}_n.$$

Pri $r = 1$ pa dobimo:

$$\frac{d}{dt} \mathbf{b}^n(0) = n \cdot (\mathbf{b}_1 - \mathbf{b}_0), \quad (3.7)$$

$$\frac{d}{dt} \mathbf{b}^n(1) = n \cdot (\mathbf{b}_n - \mathbf{b}_{n-1}), \quad (3.8)$$

kar pomeni, da je tangentni vektor na Bézierovo krivuljo v točki \mathbf{b}_0 v smeri vektorja $(\mathbf{b}_1 - \mathbf{b}_0)$, medtem ko je tangentni vektor na Bézierovo krivuljo v točki \mathbf{b}_n v smeri vektorja $(\mathbf{b}_n - \mathbf{b}_{n-1})$. Vloga tangent v robnih točkah je zelo pomembna pri zveznosti in pri konstrukciji zlepkov Bézierovih krivulj.

3.6 Zveznost in gladkost Bézierovih zlepkov

Večkrat naletimo na sledeči problem: imamo končno mnogo točk in jih moramo interpolirati oziroma aproksimirati. V prvem primeru moramo narisati krivuljo, ki bo potekala skozi dane točke. V drugem primeru pa moramo začrtati tako krivuljo, da se bo čimbolj "približala" danim točkam. Problem je, kako rešiti to nalogo, ko imamo danih veliko število točk. Če bi želeli začrtati eno samo krivuljo visoke stopnje, potem bi naleteli na veliko težav. Najbolje je, da interpoliramo oziroma aproksimiramo manjši del podatkov in potem dobljene krivulje manjše stopnje zlepimo skupaj. Dobljeni rezultat imenujemo *zlepek*. Posamezne krivulje, ki sestavljajo zlepek, pa bomo imenovali *odseki*. Ukvarjali se bomo z *Bézierovimi zlepkami*, to so zlepki, ki imajo kot odseke Bézierove krivulje.

Kar se tiče zveznosti, nimamo težav pri odsekih, saj so Bézierove krivulje neskončnokrat zvezno odvedljive krivulje (ker so polinomi). Torej se bo potrebno osredotočiti le na t.i. *stične točke*, to so stičišča posameznih odsekov. Najprej formalno definirajmo pojem zlepka, nato pa bomo preučili pogoje, ki morajo veljati, da bo celotna krivulja zvezna in gladka.

Definicija 15. *Naj bo*

$$u_0 < u_1 < \dots < u_{n-1} < u_n$$

*naraščajoče zaporedje delilnih točk, ki jih bomo imenovali **vozli**. Nad vsakim podintervalom $[u_i, u_{i+1}]$ je definirana Bézierova krivulja $\mathbf{s}_i, i = 0, 1, \dots, n-1$, stopnje n . Tako dobljen zlepek imenujemo **Bézierov zlepek** in ga označimo s \mathbf{s} .*

Globalni parameter u preteče interval $[u_0, u_n]$. Za vsako vrednost parametra u leži točka $\mathbf{s}(u)$ na krivulji \mathbf{s} . Ker pa smo celotno krivuljo sestavili iz manjših delov, je smiselno, da uvedemo **lokalni parameter** t :

$$t = \frac{u - u_i}{\Delta u_i}, \quad u \in [u_i, u_{i+1}], \quad i = 0, 1, \dots, n-1.$$

Lokalni parameter preteče na vsakem odseku interval $[0, 1]$. Videli smo že, da je preslikanje intervalov afina operacija. Torej velja:

$$\mathbf{s}|_{[u_i, u_{i+1}]}(u) = \mathbf{s}_i(t).$$

Poglejmo, kaj se zgodi, če odvajamo krivuljo v točki pri parametru u . Pri tem uporabimo navadna pravila za odvajanje:

$$\frac{d}{du}\mathbf{s}(u) = \frac{d}{dt}\mathbf{s}_i(t) \frac{dt}{du} = \frac{1}{\Delta u_i} \frac{d}{dt}\mathbf{s}_i(t). \quad (3.9)$$

Sedaj pa pogledajmo, kateri pogoji morajo veljati, da bo naša celotna krivulja \mathcal{C}^k zvezna.

3.6.1 \mathcal{C}^k zveznost

Naj bosta \mathbf{s}_0 in \mathbf{s}_1 odseka Bézierovega zlepka. Naj bo prvi definiran na intervalu $[u_0, u_1]$, drugi pa na $[u_1, u_2]$. Zlepek \mathbf{s} bo znotraj intervalov \mathcal{C}^k zvezen povsod, saj sta Bézierovi krivulji polinoma. Kaj pa v stični točki $\mathbf{s}(u_1)$?

- Če hočemo, da bo zlepek \mathcal{C}^0 zvezen, potem mora veljati

$$\mathbf{s}_0(1) = \mathbf{s}_1(0).$$

Zadnja kontrolna točka prve krivulje mora biti torej enaka prvi kontrolni točki druge krivulje.

Označimo kontrolne točke prvega kontrolnega poligona z $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$. Kontrolne točke drugega poligona pa z $\mathbf{b}_n, \mathbf{b}_{n+1}, \dots, \mathbf{b}_{2n}$.

- Če hočemo, da bo zlepek \mathcal{C}^1 zvezen, potem morata biti prva odvoda v stični točki enaka. Uporabimo (3.7), (3.8) oziroma (3.9). Ker vemo, da je kontrolna točka \mathbf{b}_0 drugega poligona v resnici točka \mathbf{b}_n , dobimo pogoj:

$$\frac{1}{\Delta u_0} \Delta \mathbf{b}_{n-1} = \frac{1}{\Delta u_1} \Delta \mathbf{b}_n,$$

oziroma, če malo preoblikujemo:

$$\mathbf{b}_{n+1} = \mathbf{b}_n + \frac{\Delta u_1}{\Delta u_0} \Delta \mathbf{b}_{n-1}.$$

Kontrolne točke $\mathbf{b}_{n-1}, \mathbf{b}_n, \mathbf{b}_{n+1}$ morajo biti kolinearne in v razmerju $\Delta u_0 : \Delta u_1$. Tu vidimo, kako pomembno vlogo odigra izbira vozlov.

- Če hočemo, da bo zlepek \mathcal{C}^2 zvezen, potem morata v stičnih točkah biti enaka tudi druga odvoda. Uporabimo isti pristop kot v prejšnji točki in dobimo:

$$\frac{1}{(\Delta u_0)^2} \Delta^2 \mathbf{b}_{n-2} = \frac{1}{(\Delta u_1)^2} \Delta^2 \mathbf{b}_n.$$

Zgled 3.4

Na sliki 3.4 imamo primer Bézierovega zlepka, ki je sestavljen iz dveh kubičnih Bézierovih krivulj, ki sta na sliki označeni z rdečo oz. zeleno barvo. Izbrali smo vozle $u_0 = 0, u_1 = \frac{1}{3}$ in $u_2 = 1$. Vidimo, da so točke $\mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4$ kolinearne in v razmerju $\Delta u_0 : \Delta u_1 = \frac{1}{3} : \frac{2}{3}$.

V praksi redko srečamo \mathcal{C}^k zveznosti za $k = 3, 4, \dots$, tako da bomo izpustili splošno formulo za višji red zveznosti.

3.6.2 Geometrijska zveznost G^k

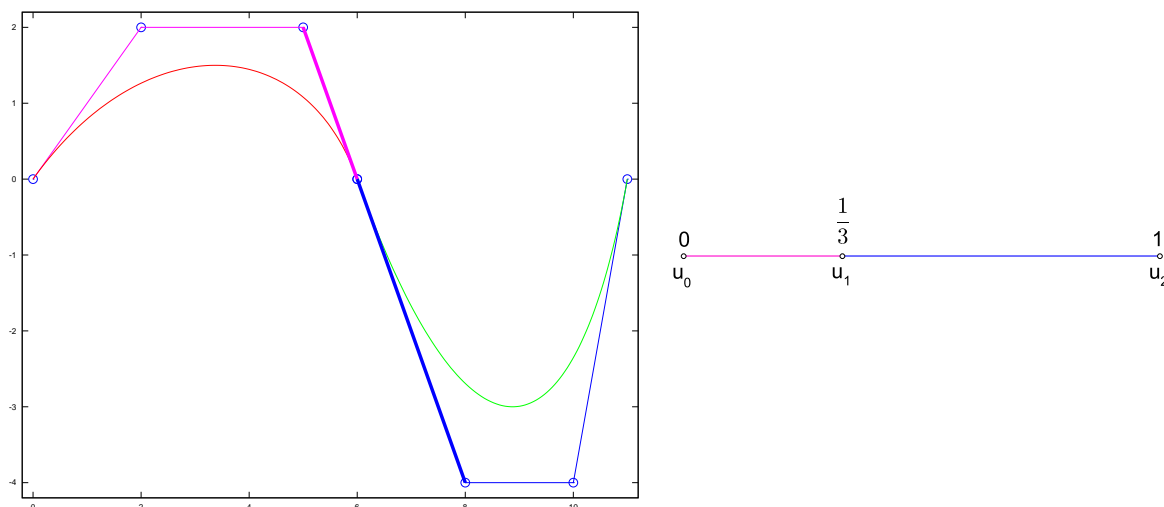
Pogledali si bomo drug pomemben pojem: geometrijska zveznost G^k . Podobno kot pri \mathcal{C} zveznosti bomo preučili le G^0, G^1 in G^2 zveznosti. Najprej povejmo, da je G^0 kar enaka že obravnavani \mathcal{C}^0 zveznosti. Definirajmo sedaj formalno geometrijsko zveznost G^k :

Definicija 16. Naj bosta \mathbf{s}_0 in \mathbf{s}_1 vsaj k -krat zvezno odvedljiva odseka zlepka \mathbf{s} . Naj bo prvi definiran na intervalu $[u_0, u_1]$, drugi pa na intervalu $[u_1, u_2]$. Predpostavimo, da je zlepek zvezen v točki $\mathbf{s}(u_1)$:

$$\mathbf{s}_0(u_1) = \mathbf{s}_1(u_1).$$

Zlepek \mathbf{s} je **geometrijsko zvezen reda k** , torej G^k zvezen, če obstaja regularna bijektivna reparametrizacija $\rho : [v_0, v_1] \rightarrow [u_0, u_1]$, da bo $\rho'(v_1) > 0$ in

$$\frac{d^\ell}{du^\ell} \mathbf{s}_1(u_1) = \frac{d^\ell}{dv^\ell} \mathbf{s}_0(\rho(v_1)), \quad \ell = 1, 2, \dots, k. \quad (3.10)$$

Slika 3.4: Primer Bézierovega zleпка, ki je C^1 zvezen, in prikaz izbranih vozlov.

Uporabimo (3.10) in dobimo pogoj za G^1 zveznost:

$$\frac{d}{du}\mathbf{s}_1(u_1) = \frac{d}{du}\mathbf{s}_0(u_1) \cdot \rho'(v_1).$$

Kar pomeni, da sta enotska tangentna vektorja v $\mathbf{s}(u_1)$ enaka. Uvedemo konstanto $\alpha \in \mathbb{R}$ in pogoj se spremeni v:

$$\frac{d}{du}\mathbf{s}_1(u_1) = \frac{d}{du}\mathbf{s}_0(u_1) \cdot \alpha, \quad \alpha > 0. \quad (3.11)$$

Torej bo zlepek v $\mathbf{s}(u_1)$ G^1 zvezen, če bo obstajal tak $\alpha > 0$, da bo veljala enakost (3.11). Uporabimo sedaj spet relacijo (3.10) za pogoj za geometrijsko zveznost G^2 :

$$\frac{d^2}{du^2}\mathbf{s}_1(u_1) = \frac{d^2}{du^2}\mathbf{s}_0(u_1) \cdot \rho'(v_1) + \frac{d}{du}\mathbf{s}_0(u_1) \cdot \rho''(v_1). \quad (3.12)$$

Vpeljimo spet nove konstante v (3.12):

$$\frac{d^2}{du^2}\mathbf{s}_1(u_1) = \frac{d^2}{du^2}\mathbf{s}_0(u_1) \cdot \alpha^2 + \frac{d}{du}\mathbf{s}_0(u_1) \cdot \beta. \quad (3.13)$$

Ne pozabimo, da preden se lotimo G^2 zveznosti danega zleпка, moramo predpostaviti, da je zlepek G^1 zvezen.

Poglejmo sedaj G^k zveznost bolj podrobno za ravninske Bézierove zlepek:

Naj bosta \mathbf{s}_0 in \mathbf{s}_1 odseka Bézierovega zleпка. Naj bo prvi definiran na intervalu $[u_0, u_1]$, drugi pa na $[u_1, u_2]$. Označimo kontrolne točke prvega kontrolnega poligona z $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$. Kontrolne točke drugega poligona pa z $\mathbf{b}_n, \mathbf{b}_{n+1}, \dots, \mathbf{b}_{2n}$. Predpostavimo, da je zlepek zvezen. Kaj lahko povemo o G^1 zveznosti?

Uporabimo (3.11) in odvode Bézierovih krivulj:

$$n \cdot \frac{1}{\Delta u_1} \Delta \mathbf{b}_n = n \cdot \frac{1}{\Delta u_0} \Delta \mathbf{b}_{n-1} \cdot \alpha, \quad \alpha > 0.$$

Če to preoblikujemo, dobimo

$$\Delta \mathbf{b}_n = \Delta \mathbf{b}_{n-1} \cdot \alpha \frac{\Delta u_1}{\Delta u_0},$$

oziroma

$$\mathbf{b}_n = \mathbf{b}_{n-1} \cdot \frac{\alpha \frac{\Delta u_1}{\Delta u_0}}{\alpha \frac{\Delta u_1}{\Delta u_0} + 1} + \mathbf{b}_{n+1} \cdot \frac{1}{\alpha \frac{\Delta u_1}{\Delta u_0} + 1}.$$

Geometrijsko to pomeni, da se *smer* tangente spreminja zvezno.

Z definicijo fleksijske ukrivljenosti, ki smo jo definirali v podpoglavju 2.3, bomo pokazali pogoje za G^2 zveznost. Označimo s $\kappa_0(u_1)$ ukrivljenost odseka \mathbf{s}_0 v točki $\mathbf{s}(u_1)$ in podobno s $\kappa_1(u_1)$ ukrivljenost odseka \mathbf{s}_1 v točki $\mathbf{s}(u_1)$. Po definiciji velja

$$\kappa_0(u_1) = \frac{\mathbf{s}'_0(u_1) \times \mathbf{s}''_0(u_1)}{\|\mathbf{s}'_0(u_1)\|^3}.$$

Upoštevajoč (3.11) in (3.13) preoblikujemo dano ukrivljenost v:

$$\kappa_0(u_1) = \frac{\mathbf{s}'_0(u_1) \times \mathbf{s}''_0(u_1)}{\|\mathbf{s}'_0(u_1)\|^3} = \frac{\frac{\mathbf{s}'_1(u_1)}{\alpha} \times \frac{\mathbf{s}''_1(u_1) - \mathbf{s}'_0(u_1) \cdot \beta}{\alpha^2}}{\left\| \frac{\mathbf{s}'_1(u_1)}{\alpha} \right\|^3}. \quad (3.14)$$

Podobno izračunamo še $\kappa_1(u_1)$:

$$\kappa_1(u_1) = \frac{\mathbf{s}'_1(u_1) \times \mathbf{s}''_1(u_1)}{\|\mathbf{s}'_1(u_1)\|^3} = \frac{(\mathbf{s}'_0(u_1) \cdot \alpha) \times (\mathbf{s}'_0(u_1) \cdot \alpha^2 + \mathbf{s}'_0(u_1) \cdot \beta)}{\|\mathbf{s}'_0(u_1) \cdot \alpha\|^3}. \quad (3.15)$$

Če upoštevamo dejstvo, da je smer tangente v točki $\mathbf{s}(u_1)$ enaka tako za prvi kot drugi odsek, torej:

$$\mathbf{s}'_1(u_1) = \alpha \mathbf{s}'_0(u_1),$$

in če malo poenostavimo izraza (3.14) in (3.15), dobimo:

$$\kappa_0(u_1) = \kappa_1(u_1).$$

Torej: ravninski Bézierov zlepek je G^2 zvezen, če se smer tangente in fleksijska ukrivljenost v dani stični točki zvezno spreminjata. Vidimo tudi, da pri G zveznosti parametrizacija oz. izbira vozlov ne igra nobene vloge.

3.7 Racionalne Bézierove krivulje

Videli smo, da lahko Bézierovim krivuljam zvišamo stopnjo in se tako bolj približamo željeni obliki. V tem primeru smo spreminjali število kontrolnih točk. Obstaja pa možnost, da ohranjamo vedno isti kontrolni poligon, krivuljo pa lahko vseeno preoblikujemo. Pri tem nam pomagajo t.i. *racionalne Bézierove krivulje*. Pogledali si bomo, kako jih lahko definiramo in kako jih uporabljamo. Racionalne krivulje so bolj podrobno obravnavane v [5] in [20].

Definicija 17. *Racionalna Bézierova krivulja stopnje n je definirana kot*

$$\mathbf{x}(t) = \frac{\sum_{i=0}^n \omega_i \mathbf{b}_i B_i^n(t)}{\sum_{i=0}^n \omega_i B_i^n(t)}, \quad \mathbf{x}(t), \mathbf{b}_i \in \mathbb{E}^3, \omega_i \in \mathbb{R}. \quad (3.16)$$

Racionalne Bézierove krivulje so definirane na podoben način kot polinomske Bézierove krivulje, le da tu nastopajo koeficienti ω_i , ki jih imenujemo *uteži*. Vsaka utež ω_i je poljubno realno število, ki vpliva na kontrolno točko \mathbf{b}_i :

- če je $\omega_i > 1$, potem se Bézierova krivulja približa pripadajoči kontrolni točki \mathbf{b}_i ;
- če je $\omega_i < 1$, potem se Bézierova krivulja oddalji od pripadajoče kontrolne točke \mathbf{b}_i ;
- če je $\omega_i = 1$, Bézierova krivulja ostane nespremenjena v okolici kontrolne točke \mathbf{b}_i . Opazimo, da, če velja $\omega_i = 1, \forall i$, potem se (3.16) spremeni v Bernsteinovo obliko polinomske Bézierove krivulje, saj že vemo, da se Bernsteinovi polinomi seštejejo v ena.

Opombe:

- Če so vse uteži pozitivne, potem imajo racionalne Bézierove krivulje podobne lastnosti kot navadne Bézierove krivulje, kot npr. afina invariantnost in konveksna ovojnica. Če pa je katera od uteži negativna, potem moramo biti previdni, kar se tiče lastnosti takih krivulj.
- Racionalne Bézierove krivulje so enake polinomskim, ne samo, ko so vse uteži enake ena, ampak tudi, ko so vse med seboj enake: $\omega_0 = \omega_1 = \dots = \omega_n$.
- Ne sme veljati $\omega_i = 0, \forall i$, saj bi potem imeli deljenje z nič v (3.16).

Zgled 3.5

Na sliki 3.5 si lahko pogledamo primere racionalnih Bézierovih krivulj pri različnih izbirah parametra ω_1 , ki torej vpliva na krivuljo v okolici kontrolne točke \mathbf{b}_1 . Rdeča krivulja je racionalna Bézierova krivulja, kjer so vsi parametri ω_i enaki ena, kar je ravno polinomska Bézierova krivulja. Če določimo $\omega_1 = 4$, dobimo zeleno krivuljo. Če pa določimo $\omega_1 = 0.3$, dobimo modro krivuljo.

3.8 Matrična oblika Bézierove krivulje

V prejšnjih podpoglavjih smo se pretežno ukvarjali z Bézierovo krivuljo stopnje n oblike:

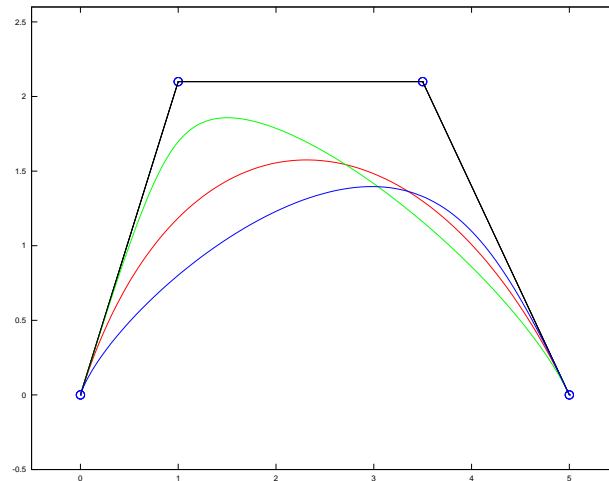
$$\mathbf{b}^n(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n(t). \quad (3.17)$$

Zapišimo (3.17) v matrično obliko. Naj bo $\mathbf{t} \in \mathbb{R}^{n+1}$ vektor potenc parametra t :

$$\mathbf{t} = [t^n, t^{n-1}, \dots, t^1, t^0]^T. \quad (3.18)$$

Naj bo A kvadratna matrika velikosti $(n+1) \times (n+1)$, ki ima naslednje elemente:

$$A_{j,i} = (-1)^{n-j-i} \binom{n-i}{j} \binom{n}{i}, \quad i, j = 0, 1, \dots, n+1.$$



Slika 3.5: Racionalne Bézierove krivulje pri različnih izbirah parametra ω_1 : če je $\omega_1 > 1$, se krivulja približa točki \mathbf{b}_1 (zelena barva), če pa je $\omega_1 < 1$, se krivulja oddalji od točke \mathbf{b}_1 (modra barva). Če je $\omega_1 = 1$, Bézierova krivulja (rdeča barva) ostane nepremenjena v okolici točke \mathbf{b}_1 .

Naj bo $\mathbf{b} \in \mathbb{R}^{n+1}$ vektor kontrolnih točk \mathbf{b}_i , $i = 0, 1, \dots, n$:

$$\mathbf{b} = [\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n]^T \quad (3.19)$$

Potem lahko Bézierovo krivuljo stopnje n zapišemo kot:

$$\mathbf{b}^n(t) = \mathbf{t}^T \cdot A \cdot \mathbf{b}.$$

Dokaz.

$$\begin{aligned} \mathbf{b}^n(t) &= \sum_{i=0}^n \mathbf{b}_i B_i^n(t) \\ &= \mathbf{b}_0 \binom{n}{0} t^0 (1-t)^n + \mathbf{b}_1 \binom{n}{1} t^1 (1-t)^{n-1} + \dots + \mathbf{b}_n \binom{n}{n} t^n (1-t)^0 \end{aligned} \quad (3.20)$$

Uporabimo binomski izrek, da zapišemo elemente $(1-t)^n, (1-t)^{n-1}, \dots, (1-t), (1-t)^0$ z vsotami in jih vključimo v (3.20). Dobimo:

$$\begin{aligned} \mathbf{b}^n(t) &= \mathbf{b}_0 \left[\binom{n}{0} t^0 \left(\sum_{j=0}^n \binom{n}{j} (-1)^{n-j} t^{n-j} \right) \right] + \\ &\quad \mathbf{b}_1 \left[\binom{n}{1} t^1 \left(\sum_{j=0}^{n-1} \binom{n-1}{j} (-1)^{n-j-1} t^{n-j-1} \right) \right] + \\ &\quad \dots + \mathbf{b}_n \left[\binom{n}{n} t^n \left(\sum_{j=0}^0 \binom{0}{j} (-1)^{n-j-n} t^{n-j-n} \right) \right] \\ &= \sum_{i=0}^n \mathbf{b}_i \left[\sum_{j=0}^{n-i} (-1)^{n-j-i} \binom{n-i}{j} \binom{n}{i} t^{n-j} \right]. \end{aligned} \quad (3.21)$$

Predstavimo vrednosti t^{n-j} v (3.21) v obliki vektorja 3.18.

Koeficiente $(-1)^{n-j-i} \binom{n-i}{j} \binom{n}{i}$ pa zapišemo v matriko A velikosti $(n+1) \times (n+1)$. Paziti moramo le, da bo indeks $j = 0, 1, \dots, n$ predstavljal vrstice matrike A , indeks $i = 0, 1, \dots, n$ pa stolpce:

$$A = \begin{matrix} & i = 0 & i = 1 & i = 2 & \dots & i = n \\ \begin{matrix} j = 0 \\ j = 1 \\ \vdots \\ \vdots \\ j = n \end{matrix} & \begin{pmatrix} (-1)^n \binom{n}{0} \binom{n}{0} \\ (-1)^{n-1} \binom{n}{1} \binom{n}{0} \\ \vdots \\ (-1)^1 \binom{n}{n-1} \binom{n}{0} \\ (-1)^0 \binom{n}{n} \binom{n}{0} \end{pmatrix} & \begin{pmatrix} (-1)^{n-1} \binom{n-1}{0} \binom{n}{1} \\ (-1)^{n-2} \binom{n-1}{1} \binom{n}{1} \\ \vdots \\ (-1)^0 \binom{n-1}{n-1} \binom{n}{1} \\ 0 \end{pmatrix} & \begin{pmatrix} (-1)^{n-2} \binom{n-2}{0} \binom{n}{2} \\ (-1)^{n-3} \binom{n-2}{1} \binom{n}{2} \\ \vdots \\ 0 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} (-1)^0 \binom{0}{0} \binom{n}{n} \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \end{matrix}$$

Matrika A je trikotna. Pod kodiagonalno imamo same ničle, ker je pri tistih indeksih i, j koeficient $\binom{n-i}{j}$ enak nič. Kar smo dobili pa moramo še pomnožiti z vektorjem 3.19, da bomo dobili izraz (3.21):

$$\mathbf{b}^n(t) = \mathbf{t}^T \cdot A \cdot \mathbf{b}.$$

□

Za implementacijo tega postopka računanja, glej prilogo B.

Zgled 3.6

Vzemimo spet kontrolne točke in parameter $t = 0.4$ iz zgleda 2.1. Zapišimo matriko A in vektorja \mathbf{t} in \mathbf{b} :

$$\mathbf{t} = [0.0256, 0.064, 0.16, 0.4, 1]^T, \quad \mathbf{b} = [\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4]^T,$$

$$A = \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -12 & 6 & 0 & 0 \\ -4 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Z opisanim postopkom izračunajmo točko na Bézierovi krivulji stopnje 4 za dani t :

$$\mathbf{b}^4(0.4) = \mathbf{t}^T \cdot A \cdot \mathbf{b} = \begin{bmatrix} 2.48 \\ 1.95 \\ 1.19 \end{bmatrix}.$$

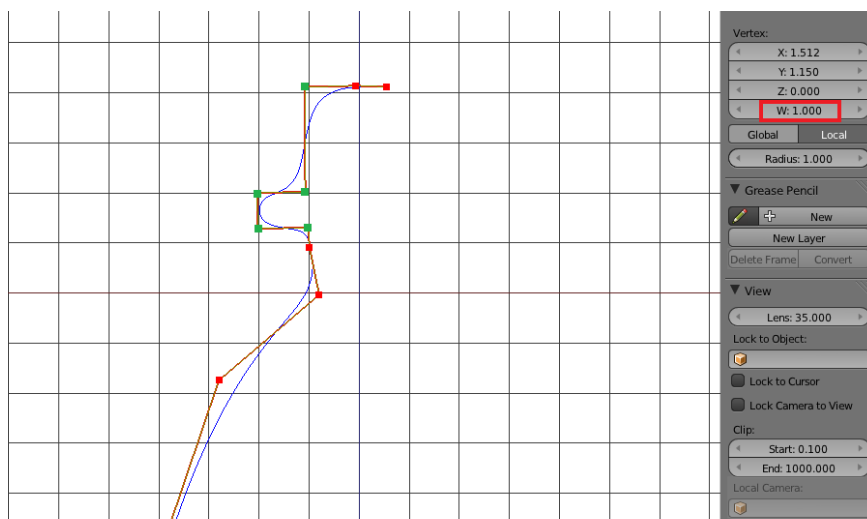
Ko te produkte izračunamo, dobimo iste vrednosti kot v zgledu 3.1, ko smo jih izračunali preko de Casteljauove sheme.

3.9 Računalniško oblikovanje z Bézierovimi krivuljami

Po teoretični obravnavi si bomo v zadnjem podpoglavju pogledali, kako lahko oblikujemo predmete s pomočjo Bézierovih krivulj. Pri tem bomo uporabili računalniški program Blender. Kot smo že povedali v uvodu magistrskega dela, bomo tu samo prikazali, kako so lahko krivulje uporabne pri računalniškem modeliranju. Podrobnemu opisu delovanja

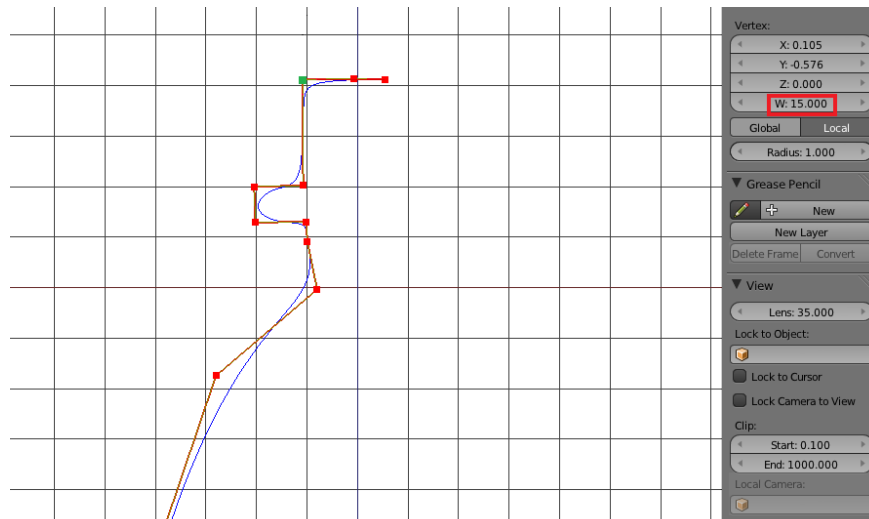
Blenderja in njegovih ukazov se bomo posvetili v zadnjem poglavju. Uporabili bomo koncepte, ki smo jih srečali v prejšnjih podpoglavjih, da bomo oblikovali *steklenico*. Začeli bomo tako, da bomo načrtali samo prerez steklenice, ki ga bomo zavrteli okoli svoje osi in dobili vrtenino v obliki steklenice. Prerez lahko načrtamo na dva načina:

- Z ustreznim Blenderjevim ukazom skonstruiramo racionalno kubično Bézierovo krivuljo. Označimo zadnjo kontrolno točko in z ukazom *Extrude* dodamo novo kontrolno točko, ki jo nato postavimo na poljubno mesto. Ukaz ponovimo na novo nastali točki. To zaporedoma ponavljamo, dokler ne pridemo do željene oblike. Na vsakem koraku povečamo stopnjo racionalne krivulje (ki ni zlepek). Oblike torej preoblikujemo s prostim očesom.
- Podobno kot prej začnemo s kubično racionalno Bézierovo krivuljo in z dodajanjem novih kontrolnih točk oblikujemo le del prereza. Nato dodamo novo Bézierovo krivuljo, s katero preoblikujemo drugi del prereza. Sedaj moramo oba dobljena racionalna kosa zlepiti, da dobimo Bézierov zlepek. Kar se tiče C^0 in G^0 zveznosti ne bomo imeli velikih težav, ker je dovolj zagotoviti, da se zadnja kontrolna točka prvega odseka ujema s prvo kontrolno točko drugega odseka. V Blenderju določimo koordinate neke točke tako, da označimo točko in spremenimo ustrezna polja, ki se nahajajo na desni strani pod imenom *Vertex* (glej sliko 3.6). Zagotovimo lahko tudi G^1 zveznost tako, da označimo tri kontrolne točke, kjer je srednja izmed teh stična točka zleпка, in jim določimo koordinate, da bodo kolinearne. Kolinearnost lahko določimo tudi s prostim očesom, saj bo rezultat zadovoljiv. Določitev višjih zveznosti, ki smo jih opisali v teoretičnem delu, pa v Blenderju ni tako zelo preprosta naloga.



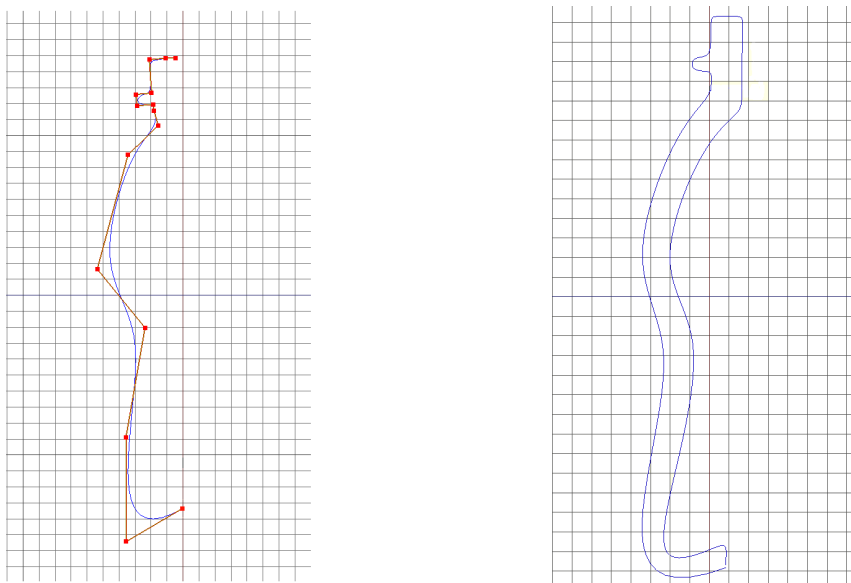
Slika 3.6: Začetek oblikovanja prereza steklenice.

Izbral sem prvi način določanja prereza steklenice. Na sliki 3.6 je prikazan začetek našega oblikovanja. Začeli smo, kot rečeno, s štirimi kontrolnimi točkami (zeleno obarvane točke) in potem dodali nove. Računalniški program nariše racionalno Bézierovo krivuljo z utežmi $\omega_i = 1, \forall i$. Na sliki 3.6 vidimo, da imajo začetne štiri točke uteži enake 1 in torej predstavljajo navadno polinomsko Bézierovo krivuljo. Ker bi radi imeli zamašek z bolj ostrimi koti, označimo ustrezno kontrolno točko in ji zvišajmo utež. Operacija je prikazana na sliki 3.7 (zeleno obarvana točka). Kako utežem določiti prave vrednosti je stvar uporabnika. Postaviti se moramo na ustrezno točko in poskusiti razne možnosti, dokler nismo zadovoljni z rezultatom. Pomagamo si tudi tako, da premikamo kontrolne točke.



Slika 3.7: Popravek uteži ustrezne kontrolne točke.

Nadaljujemo z dodajanjem novih točk in popravljanjem racionalnega kosa, ki bo predstavljal prerez steklenice. Rezultat je prikazan na sliki 3.8 (levo).



Slika 3.8: Končni prerez steklenice, ki ga potem dopolnimo.

Če bi sedaj zavrteli nastali prerez okoli svoje osi z ukazom *Spin*, bi dobili vrtenino s “polno” prostornino, kot če steklenica ne bi imela praznega prostora v sebi. Da bi pri dodajanju barv in prozornosti prišli do bolj realističnega rezultata, dopolnimo prerez do prereza na sliki 3.8 (desno). Na ta način bomo lahko zavrteli objekt in dobili steklenico, ki bo imela na eni strani “prazno, votlo” prostornino, na drugi strani pa bomo ustvarili debelino stekla, kar bo prispevalo do še bolj realističnega prikaza steklenice.



Slika 3.9: Steklenica po zavrtanju prereza.

Z Blenderjem lahko potem zavrtimo prerez in dobimo steklenico na sliki 3.9. Vidimo, da nastali objekt ni ravno “gladka” steklenica. Problem rešimo tako, da uporabimo ukaz *Smooth*, ki “zgladi” objekte. Potem ko apliciramo ukaz na našo steklenico, dobimo rezultat na sliki 3.10 (levo), in dobimo lepo oblikovano steklenico, ki jo lahko kasneje polepšamo z barvami, s sencami, s prozornostjo in podobnimi lastnostmi oz. lahko vanjo dodamo poljubno tekočino. Na sliki 3.10 (desno) je prikazana steklenica, potem ko ji dodamo svetlo modro barvo in prozornost, zamašek pa pobarvamo z rdečo barvo. Če si natančno ogledamo končni rezultat, opazimo, da je debelina stekla označena s temnejšo barvo, kar je posledica svetlobe, ki “potuje” po objektu. Tako dobimo torej določen efekt za notranjost steklenice, drugačen efekt pa za debelino steklenice.



Slika 3.10: “Gladka” steklenica in njen končni barvni videz.

Poglavje 4

Bézierove ploskve

V tem poglavju bomo obravnavali podobne pojme in koncepte, ki smo jih srečali že v prejšnjem poglavju. Tu bo linearno interpolacijo zamenjala *bilinearna interpolacija*, s katero bomo definirali *Bézierove ploskve*. Spoznali bomo, kako jih lahko definiramo, sestavimo in zlepimo. Zato, da bomo lažje primerjali Bézierove krivulje z Bézierovimi ploskvami, bomo obravnavali podpoglavja v istem vrstnem redu kot v drugem poglavju.

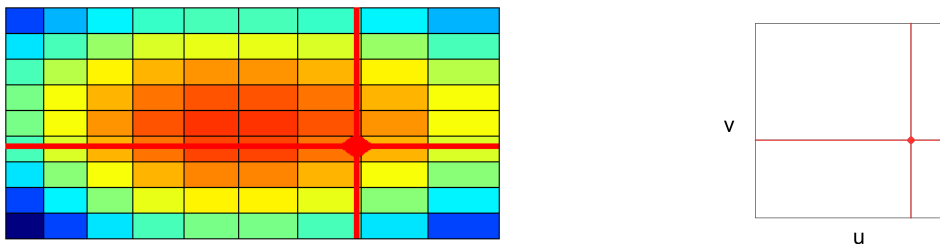
Najprej pa definirajmo *parametrične ploskve*, ki so, podobno kot parametrične krivulje, zelo uporabne v računalniško podprtem načrtovanju.

Definicija 18. Ploskev \mathbf{x} v parametrični obliki zapišemo 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 pove, da se vsaka točka pravokotnika preslika na ploskev oz. da je ploskev \mathbf{x} slika pravokotnika $[a, b] \times [c, d]$ v (u, v) -ravnini. Pravokotnik v (u, v) -ravnini, ki ga imenujemo tudi *domena* parametrične ploskve, igra isto vlogo kot interval $[a, b]$ pri Bézierovih krivuljah. Domena ploskve je lahko tudi trikotne oblike, kot bomo videli v naslednjem poglavju.

Slika 4.1 nam prikazuje primer parametrične ploskve, če se postavimo na z -os in jo pogledamo od zgoraj navzdol. Točko na ploskvi, ki je označena z rdečim kvadratom, smo dobili pri parametrih $u = 0.8$ in $v = 0.4$. Ob ploskvi je prikazana tudi njena domena.



Slika 4.1: Primer parametrične ploskve in njena domena. Barve na ploskvi predstavljajo višino ploskve oz. z -komponento točk na ploskvi.

4.1 Bilinearna interpolacija

Koncept linearne interpolacije pri krivuljah bomo prenesli 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{E}^3 . Točke $\mathbf{x} \in \mathbb{E}^3$, oblike

$$\mathbf{x}(u, v) = \sum_{i=0}^1 \sum_{j=0}^1 \mathbf{b}_{i,j} B_i^1(u) B_j^1(v), \quad (4.1)$$

tvorijo *hiperbolični paraboloid* skozi dane začetne točke.

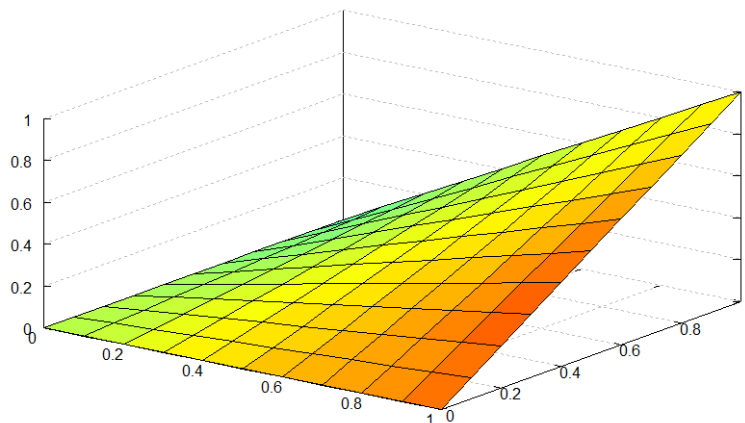
Ploskev \mathbf{x} imenujemo *bilinearni interpolant*, saj je (4.1) linearni izraz v obeh parametrih in interpolira dane točke. Enačbo (4.1) lahko zapišemo v matrični obliki:

$$\mathbf{x}(u, v) = [1 - u, \quad u] \begin{bmatrix} \mathbf{b}_{0,0} & \mathbf{b}_{0,1} \\ \mathbf{b}_{1,0} & \mathbf{b}_{1,1} \end{bmatrix} \begin{bmatrix} 1 - v \\ v \end{bmatrix}. \quad (4.2)$$

V prilogi C je dostopna implementacija zapisa (4.2).

Zgled 4.1

Naj bodo dane naslednje štiri točke: $\mathbf{b}_{0,0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{b}_{0,1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{b}_{1,0} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, $\mathbf{b}_{1,1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$. Na spodnji sliki je narisan hiperbolični paraboloid, ki ga dobimo tako, da v (4.1) vstavimo dane štiri točke.



Slika 4.2: Primer hiperboličnega paraboloida.

Zgled 4.1 nam ponuja odgovor na vprašanje, odkod ime hiperbolični paraboloid. Če ploskev iz zgleda presekamo z ravnino, ki je vzporedna z x, y -ravnino, potem je dobljena krivulja hiperbola; če pa ploskev presekamo z ravnino, ki je vzporedna z z -osjo, potem je dobljena krivulja parabola.

4.2 de Casteljauov algoritem

Bézierove krivulje smo dobili z večkratno uporabo linearne interpolacije. Podobno bomo Bézierove ploskve dobili z zaporedno uporabo bilinearne interpolacije. Opišimo de Casteljauov algoritem za Bézierove ploskve, ki bo imel podobno obliko kot (4.2).

Naj 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. De Casteljauov algoritem je oblike:

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

pri čemer

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

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

Zgled 4.2

Vzemimo Bézierovo ploskev, definirano z naslednjo Bézierovo mrežo:

$$\begin{bmatrix} \mathbf{b}_{0,0} \\ \mathbf{b}_{1,0} \\ \mathbf{b}_{2,0} \end{bmatrix} \quad \begin{bmatrix} \mathbf{b}_{0,1} \\ \mathbf{b}_{1,1} \\ \mathbf{b}_{2,1} \end{bmatrix} \quad \begin{bmatrix} \mathbf{b}_{0,2} \\ \mathbf{b}_{1,2} \\ \mathbf{b}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 1 \\ 3 \\ 0 \\ -1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 2 \\ 3 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 2 \\ -1 \\ 1 \\ 2 \\ 0 \\ 3 \\ 2 \\ -1 \end{bmatrix}.$$

Po prvem koraku de Casteljauovega algoritma za $u = 0.3$ in $v = 0.5$ dobimo:

$$\begin{bmatrix} \mathbf{b}_{0,0}^{1,1} \\ \mathbf{b}_{1,0}^{1,1} \end{bmatrix} \quad \begin{bmatrix} \mathbf{b}_{0,1}^{1,1} \\ \mathbf{b}_{1,1}^{1,1} \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.5 \\ 0.45 \end{bmatrix} \quad \begin{bmatrix} 0.3 \\ 1.5 \\ 0.3 \\ 1.6 \\ 1.5 \\ 0.7 \end{bmatrix}.$$

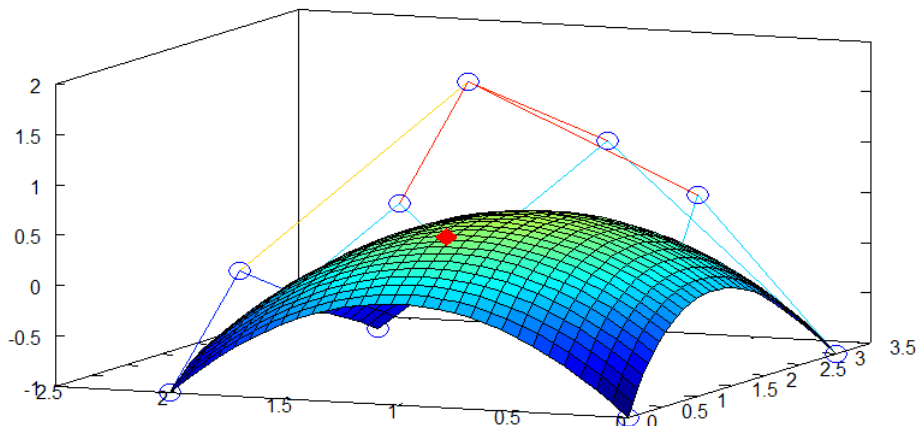
Po drugem koraku pa pridemo do točke

$$\begin{bmatrix} \mathbf{b}_{0,0}^{2,2} \end{bmatrix} = \begin{bmatrix} 0.69 \\ 1 \\ 0.525 \end{bmatrix}.$$

Na sliki 4.3 je prikazana Bézierova ploskev iz zgleda in njena kontrolna mreža. Točka $\mathbf{b}_{0,0}^{2,2}$ na Bézierovi ploskvi, ki smo jo dobili v zgledu pri parametrih $(u, v) = (0.3, 0.5)$, je označena na sliki z rdečim kvadratom.

4.3 Bernsteinova oblika Bézierovih ploskev ali tenzorski produkt

Bézierove ploskve lahko zapišemo tudi eksplicitno z Bernsteinovi polinomi, kot smo že storili pri krivuljah. V ta namen moramo uvesti pojem *tenzorskega produkta*. Poleg rekurzivne definicije z de Casteljauovim algoritmom, lahko željeno ploskev dobimo tudi tako, da dano začetno krivuljo premikamo vzdolž dveh krivulj. Začetna krivulja bo tako zarisala ploskev v prostoru. Uporabimo ta koncept in definirajmo Bézierovo ploskev s pomočjo Bernsteinovih polinomov.



Slika 4.3: Bézierova ploskev s kontrolno mrežo, kjer rdeči kvadrček predstavlja točko na ploskvi, ki smo jo izračunali pri parametrih $(u, v) = (0.3, 0.5)$.

Predpostavimo, da je krivulja, ki jo premikamo po prostoru, Bézierova krivulja stopnje m z $m + 1$ kontrolnimi točkami, ter da sta obe krivulji, vzdolž katerih premikamo našo krivuljo, Bézierovi krivulji iste stopnje, recimo stopnje n z $n + 1$ kontrolnimi točkami. Krivuljo, ki jo bomo premikali, zapišemo kot

$$\mathbf{b}^m(u) = \sum_{i=0}^m \mathbf{b}_i B_i^m(u), \quad (4.3)$$

vsaka točka \mathbf{b}_i pa se premika vzdolž Bézierovih krivulj stopnje n :

$$\mathbf{b}_i(v) = \sum_{j=0}^n \mathbf{b}_{i,j} B_j^n(v). \quad (4.4)$$

Če hočemo torej najti točko na Bézierovi ploskvi, združimo (4.3) in (4.4) in dobimo točko $\mathbf{b}^{m,n}(u, v)$:

$$\mathbf{b}^{m,n}(u, v) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} B_i^m(u) B_j^n(v). \quad (4.5)$$

V prilogi Č je implementacija relacije (4.5).

4.4 Pomembne lastnosti Bézierovih ploskev

Poglejmo si nekaj lastnosti Bézierovih ploskev in jih dokažimo. Nekaj izmed teh lastnosti smo že srečali pri krivuljah.

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

Dokaz. To hitro vidimo, ko izberemo pare parametrov (u, v) , ki predstavljajo vogale pravokotne domene ploskve $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$. Uporabimo (4.5):

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

saj so vsi Bernsteinovi polinomi enaki nič, razen pri $i = j = 0$. Tako nam v vsoti ostane element

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

Podobno izračunamo še ostale:

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

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

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

□

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. Iz prejšnjega poglavja vemo, da se Bernsteinovi polinomi seštejejo v ena. Torej:

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

□

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

Dokaz. Naj bosta $0 \leq u, v \leq 1$ parametra. Potem bo produkt $B_i^m(u) \cdot B_j^n(v)$ zagotovo nenegativen. Kar pomeni, da bo (4.5) konveksna kombinacija kontrolnih točk, in bo torej Bézierova ploskev ležala znotraj konveksne ovojnice svoje kontrolne mreže. To lahko vidimo tudi s pomočjo de Casteljauovega algoritma: vsaka vmesna točka v algoritmu bo ležala med štirimi vogali kontrolne mreže. Posledično bo celotna ploskev ležala znotraj konveksne ovojnice svoje kontrolne mreže.

□

4. Bézierove ploskve so invariantne za afine preslikave.

Dokaz. Pomagajmo si spet z de Casteljauovim algoritmom, ki je zaporedje bilinearnih interpolacij. Upoštevamo dejstvo, da je bilinearna interpolacija v bistvu aplikacija linearne interpolacije v smeri u in v . V prejšnjem poglavju smo dokazali, da je linearne interpolacije invariantna za afine preslikave. To pomeni, da je tudi bilinearna interpolacija afino invariantna. Bézierove ploskve so torej invariantne za afine preslikave.

To pomeni, da dobimo isti rezultat, če najprej izračunamo točko na Bézierovi ploskvi in jo afino preslikamo, ali pa, če najprej afino preslikamo Bézierovo mrežo in nato izračunamo točko na preslikani ploskvi.

□

5. Robne in izoparametrične krivulje Bézierovih ploskev

Spomnimo se formule (4.5).

Definicija 19. *Robne krivulje Bézierovih ploskev so Bézierove krivulje, ki ležijo na robovih ploskve. Dobimo jih tako, da izberemo parametre $u = 0, v = 0, u = 1$ in $v = 1$.*

Upoštevajoč (4.5) dobimo točke na vseh štirih robnih krivuljah kot: $\mathbf{b}^{m,n}(0, v)$, $\mathbf{b}^{m,n}(u, 0)$, $\mathbf{b}^{m,n}(1, v)$ oz. $\mathbf{b}^{m,n}(u, 1)$.

Definicija 20. *Izoparametrične krivulje Bézierovih ploskev so Bézierove krivulje, ki ležijo na ploskvi in jih dobimo tako, da fiksiramo parameter u ali parameter v .*

Uporabimo (4.5). Točke na izoparametričnih krivuljah Bézierove ploskve dobimo kot

$$\begin{aligned}\mathbf{b}^{m,n}(u_0, v) &= \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} B_i^m(u_0) B_j^n(v), \\ \mathbf{b}^{m,n}(u, v_0) &= \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} B_i^m(u) B_j^n(v_0),\end{aligned}$$

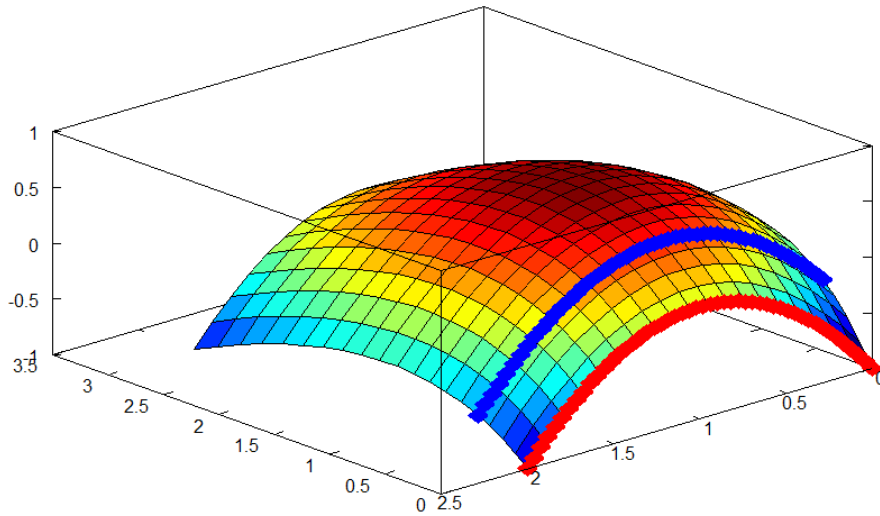
kjer sta u_0 in v_0 fiksni vrednosti za parametra u in v .

Na sliki 4.4 je Bézierova ploskev iz zgleda 4.2, kjer je z rdečo barvo označena robna krivulja pri $u = 0$ in z modro barvo izoparametrična krivulja pri $u = 0.2$.

Analizirali bi lahko tudi druge vrste krivulj na ploskvi, kot npr. krivulje, ki jih dobimo, če preslikamo diagonali domene. Več informacij glede tega najdemo v [4].

4.5 Višanje stopnje

Podobno kot pri krivuljah bomo opisali, kako zvišamo stopnjo Bézierove ploskve. Razlogi, zakaj bi to storili, so isti kot v prejšnjem poglavju: če zvišamo stopnjo ploskve, ne da bi spremenili njene oblike, le-ta postane bolj fleksibilna.



Slika 4.4: Bézierova ploskev z robno in izoparametrično krivuljo.

Naj bo $\mathbf{b}^{m,n}$ Bézierova ploskev stopnje (m, n) . To pomeni, da našo ploskev dobimo tako, da Bézierovo krivuljo stopnje m premikamo vzdolž dveh Bézierovih krivulj stopnje n . Osredotočimo se na Bézierovo krivuljo stopnje m . Želimo zvišati njeno stopnjo, tako da bi dobili ploskev stopnje $(m + 1, n)$. Naša naloga bo torej izračunati nove kontrolne točke $\mathbf{b}_{i,j}^{(1,0)}$, da bo veljalo:

$$\mathbf{b}^{m,n}(u, v) = \sum_{j=0}^n \left[\sum_{i=0}^{m+1} \mathbf{b}_{i,j}^{(1,0)} B_i^{m+1}(u) \right] B_j^n(v). \quad (4.6)$$

Paziti moramo, da zvišamo stopnjo vsem krivuljam stopnje m v tenzorskem produktu. Ker imata krivulji stopnje n $n + 1$ kontrolnih točk, izraz (4.6) predstavlja $n + 1$ višanj stopenj. Uporabimo (3.2) in lahko izračunamo nove kontrolne točke

$$\mathbf{b}_{i,j}^{(1,0)} = \frac{i}{m+1} \mathbf{b}_{i-1,j} + \left(1 - \frac{i}{m+1}\right) \mathbf{b}_{i,j}, \quad (4.7)$$

kjer je

$$i = 0, 1, \dots, m+1, \quad j = 0, 1, \dots, n.$$

S tako definiranimi novimi kontrolnimi točkami smo zvišali stopnjo vsem krivuljam stopnje m v tenzorskem produktu. Na enak način bi lahko zvišali stopnjo krivuljam stopnje n in dobili Bézierovo ploskev stopnje $(m, n + 1)$. Nove kontrolne točke $\mathbf{b}_{i,j}^{(0,1)}$ bi v tem primeru izračunali kot

$$\mathbf{b}_{i,j}^{(0,1)} = \frac{j}{n+1} \mathbf{b}_{i,j-1} + \left(1 - \frac{j}{n+1}\right) \mathbf{b}_{i,j}, \quad (4.8)$$

kjer je

$$i = 0, 1, \dots, m, \quad j = 0, 1, \dots, n+1.$$

Seveda lahko zvišamo stopnjo v obe smeri u in v ter dobimo Bézierovo ploskev stopnje $(m + 1, n + 1)$. V tem primeru moramo točki $\mathbf{b}_{i,j-1}$ in $\mathbf{b}_{i,j}$ v izrazu (4.8) izraziti s pomočjo

(4.7). Kontrolne točke $\mathbf{b}_{i,j}^{(1,1)}$ Bézierove ploskve stopnje $(m+1, n+1)$ so potem enake

$$\mathbf{b}_{i,j}^{(1,1)} = \frac{j}{n+1} \left(\frac{i}{m+1} \mathbf{b}_{i-1,j-1} + \left(1 - \frac{i}{m+1}\right) \mathbf{b}_{i,j-1} \right) + \left(1 - \frac{j}{n+1}\right) \left(\frac{i}{m+1} \mathbf{b}_{i-1,j} + \left(1 - \frac{i}{m+1}\right) \mathbf{b}_{i,j} \right),$$

kjer je

$$i = 0, 1, \dots, m+1, \quad j = 0, 1, \dots, n+1.$$

Zaradi enostavnega zapisa lahko točke $\mathbf{b}_{i,j}^{(1,1)}$ zapišemo v matrični obliki:

$$\mathbf{b}_{i,j}^{(1,1)} = \begin{bmatrix} \frac{i}{m+1}, & \left(1 - \frac{i}{m+1}\right) \end{bmatrix} \begin{bmatrix} \mathbf{b}_{i-1,j-1} & \mathbf{b}_{i-1,j} \\ \mathbf{b}_{i,j-1} & \mathbf{b}_{i,j} \end{bmatrix} \begin{bmatrix} \frac{j}{n+1} \\ \left(1 - \frac{j}{n+1}\right) \end{bmatrix}. \quad (4.9)$$

Zgled 4.3

Vzemimo Bézierovo ploskev stopnje $(2, 2)$ iz zgleda 4.2. Zvišajmo njeno stopnjo v smer u -ja in v smer v -ja, tako da dobimo ploskev, ki je navidezno stopnje $(3, 3)$. Izračunajmo zato nove kontrolne točke s pomočjo (4.9) in dobimo novo kontrolno mrežo:

$$\begin{bmatrix} \mathbf{b}_{0,0}^{(1,1)} \\ \mathbf{b}_{1,0}^{(1,1)} \\ \mathbf{b}_{2,0}^{(1,1)} \\ \mathbf{b}_{3,0}^{(1,1)} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,1}^{(1,1)} \\ \mathbf{b}_{1,1}^{(1,1)} \\ \mathbf{b}_{2,1}^{(1,1)} \\ \mathbf{b}_{3,1}^{(1,1)} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,2}^{(1,1)} \\ \mathbf{b}_{1,2}^{(1,1)} \\ \mathbf{b}_{2,2}^{(1,1)} \\ \mathbf{b}_{3,2}^{(1,1)} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,3}^{(1,1)} \\ \mathbf{b}_{1,3}^{(1,1)} \\ \mathbf{b}_{2,3}^{(1,1)} \\ \mathbf{b}_{3,3}^{(1,1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 0 \\ 2/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 0 \\ 4/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 0 \\ 2/3 \\ 4/3 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2/3 \\ 5/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 3 \\ 2/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 3 \\ 4/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ -1/3 \end{bmatrix} \begin{bmatrix} 2/3 \\ 2 \\ -1/3 \end{bmatrix} \begin{bmatrix} 5/3 \\ 2 \\ -1/3 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ -1 \end{bmatrix}.$$

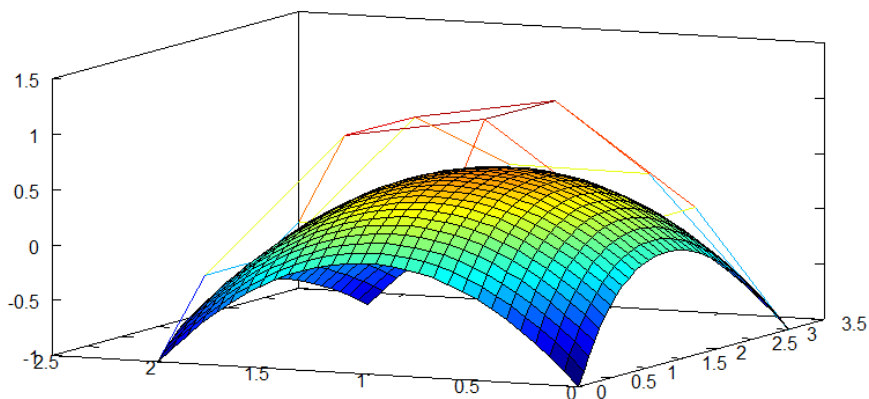
Na sliki 4.5 si lahko ogledamo Bézierovo ploskev definirano z novo kontrolno mrežo. Za primerjavo lahko pogledamo sliko 4.3 in vidimo, da je ploskev po višanju stopnje ostala nespremenjena.

4.6 Odvodi

Pri Bézierovih ploskvah imamo opravka z bilinearno interpolacijo. Ker jih hočemo odvajati, bomo morali upoštevati odvod v smeri parametra u in v smeri parametra v . Uporabljali bomo torej *parcialne odvode* $\frac{\partial}{\partial u}$ in $\frac{\partial}{\partial v}$.

Trditev 3. Naj bo $\mathbf{b}^{m,n}$ Bézierova ploskev stopnje (m, n) . Potem velja:

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



Slika 4.5: Bézierova ploskev z novo kontrolno mrežo po višanju stopnje.

kjer je

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

Dokaz. Vzemimo tenzorski produkt (4.5). Ko odvajamo po spremenljivki u , moramo odvajati le del tenzorskega produkta, ki je vezan na parameter u , ter dobimo

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

Opazimo, da je $\frac{\partial}{\partial u} \sum_{i=0}^m \mathbf{b}_{i,j} B_i^m(u) = \frac{d}{du} \mathbf{b}^m(u)$, in upoštevajoč (3.5):

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

Ko odvajamo po parametru v , računamo analogno kot prej:

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

Z dobljenimi rezultati si lahko pomagamo, da izračunamo še mešani odvod:

$$\begin{aligned} \frac{\partial^2}{\partial u \partial v} \mathbf{b}^{m,n}(u, v) &= \frac{\partial}{\partial u} \left(\frac{\partial}{\partial v} \mathbf{b}^{m,n}(u, v) \right) \\ &= \frac{\partial}{\partial u} \left(n \cdot \sum_{i=0}^m \sum_{j=0}^{n-1} \Delta^{(0,1)} \mathbf{b}_{i,j} B_i^m(u) B_j^{n-1}(v) \right) \\ &= n \cdot \sum_{j=0}^{n-1} B_j^{n-1}(v) \frac{\partial}{\partial u} \left(\sum_{i=0}^m (\mathbf{b}_{i,j+1} - \mathbf{b}_{i,j}) B_i^m(u) \right). \end{aligned} \quad (4.10)$$

V izrazu (4.10) moramo pri parcialnem odvodu le paziti na kontrolne točke. Ker odvajamo po u , bo to vplivalo le na indeks i tako pri točki $\mathbf{b}_{i,j+1}$ kot pri $\mathbf{b}_{i,j}$. Torej:

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

in trditev je dokazana. □

Opomba: Diference $\Delta^{(0,1)}$, $\Delta^{(1,0)}$, $\Delta^{(1,1)}$ so po definiciji vektorji. Parcialni odvod lahko namreč geometrijsko interpretiramo kot tangentni vektor neke izoparametrične krivulje. Za več informacij glej [5].

V prejšnjem poglavju smo definirali tudi višje odvode Bézierovih krivulj. S pomočjo (3.6) bomo sedaj definirali višje parcialne odvode Bézierovih ploskev:

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

kjer je

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

Zdaj pa lahko definiramo tudi mešane parcialne odvode višjega reda:

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

Kot smo že povedali, lahko parcialni odvod interpretiramo kot tangentni vektor na izoparametrično krivuljo. Poglejmo si, kaj se dogaja na robovih Bézierove ploskve.

Definicija 21. Naj bo $\mathbf{b}^{m,n}(u, v)$ Bézierova ploskev stopnje (m, n) . Parcialne odvode robnih krivulj Bézierove ploskve imenujemo **robni odvodi**.

Vzemimo npr. robno krivuljo pri $u = 0$ in dobimo parcialni odvod:

$$\frac{\partial^r}{\partial u^r} \mathbf{b}^{m,n}(0, v) = \frac{m!}{(m-r)!} \sum_{j=0}^n \Delta^{(r,0)} \mathbf{b}_{0,j} B_j^n(v). \quad (4.11)$$

Podobno bi dobili še ostale odvode. Robne odvode smo definirali, ker jih bomo uporabili pri določanju pogojev za C^r zveznost zlepka dveh ali več ploskev.

Preden preidemo na zlepke in zveznost Bézierovih ploskev, si pogledjmo še pojem *normalnih vektorjev*, ki jih sicer ne bomo uporabljali v tej nalogi, ampak jih je vredno omeniti, saj jih večkrat uporabljamo v računalniško podprtem geometrijskem načrtovanju.

4.7 Normalni vektorji

Definicija 22. *Normalni vektor \mathbf{n} dane parametrične ploskve je enotska normala na ploskev v dani točki.*

Izračunamo ga tako, da vektorsko zmnožimo poljubna dva vektorja, ki sta tangenta na ploskvi v dani točki. Ker smo videli, da sta parcialna odvoda $\frac{\partial}{\partial u}$ oz. $\frac{\partial}{\partial v}$ tangenta vektorja, ju lahko uporabimo, da dobimo normalni vektor. Torej, normalni vektor ploskve v dani točki $\mathbf{b}^{m,n}(u, v)$ dobimo kot

$$\mathbf{n}(u, v) = \frac{\frac{\partial}{\partial u} \mathbf{b}^{m,n}(u, v) \times \frac{\partial}{\partial v} \mathbf{b}^{m,n}(u, v)}{\left\| \frac{\partial}{\partial u} \mathbf{b}^{m,n}(u, v) \times \frac{\partial}{\partial v} \mathbf{b}^{m,n}(u, v) \right\|}.$$

Zgled 4.4

Vzemimo Bézierovo ploskev stopnje (2, 3) s kontrolno mrežo:

$$\begin{array}{cccc} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} & \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 1 \\ 0.5 \end{bmatrix} & \begin{bmatrix} 0.9 \\ 1 \\ 1.2 \end{bmatrix} & \begin{bmatrix} 2.2 \\ 1 \\ -0.9 \end{bmatrix} & \begin{bmatrix} 2.8 \\ 1.2 \\ 0.5 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} & \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix} & \begin{bmatrix} 3 \\ 3 \\ 0 \end{bmatrix} \end{array}$$

Izračunajmo normalni vektor na ploskev v točki $\mathbf{b}^{2,3}(0.2, 0.3)$. Najprej pogledjmo parcialne odvode:

$$\begin{aligned} \frac{\partial}{\partial u} \mathbf{b}^{2,3}(0.2, 0.3) &= [-0.014, 2.406, 0.351]^T \\ \frac{\partial}{\partial v} \mathbf{b}^{2,3}(0.2, 0.3) &= [2.026, 0.012, -0.618]^T. \end{aligned}$$

Torej:

$$\begin{aligned} \mathbf{n}(0.2, 0.3) &= \frac{[-0.014, 2.406, 0.351]^T \times [2.026, 0.012, -0.618]^T}{\left\| [-0.014, 2.406, 0.351]^T \times [2.026, 0.012, -0.618]^T \right\|} \\ &= \frac{[-1.491, 0.702, -4.876]^T}{5.147} = [-0.289, 0.136, -0.947]^T. \end{aligned}$$

Če izračunamo parcialne odvode v točkah $\mathbf{b}_{0,0}$, $\mathbf{b}_{0,n}$, $\mathbf{b}_{m,0}$ in $\mathbf{b}_{m,n}$, vidimo, da so kar enaki končnim diferencam, ki smo jih definirali v prejšnjem podpoglavju. Potem so normalni vektorji v teh točkah enaki:

$$\begin{aligned}\mathbf{n}(0,0) &= \frac{\Delta^{(1,0)}\mathbf{b}_{0,0} \times \Delta^{(0,1)}\mathbf{b}_{0,0}}{\|\Delta^{(1,0)}\mathbf{b}_{0,0} \times \Delta^{(0,1)}\mathbf{b}_{0,0}\|} \\ \mathbf{n}(0,1) &= \frac{\Delta^{(1,0)}\mathbf{b}_{0,n} \times \Delta^{(0,1)}\mathbf{b}_{m,0}}{\|\Delta^{(1,0)}\mathbf{b}_{0,n} \times \Delta^{(0,1)}\mathbf{b}_{m,0}\|} \\ \mathbf{n}(1,0) &= \frac{\Delta^{(1,0)}\mathbf{b}_{m-1,0} \times \Delta^{(0,1)}\mathbf{b}_{0,n-1}}{\|\Delta^{(1,0)}\mathbf{b}_{m-1,0} \times \Delta^{(0,1)}\mathbf{b}_{0,n-1}\|} \\ \mathbf{n}(1,1) &= \frac{\Delta^{(1,0)}\mathbf{b}_{m-1,n} \times \Delta^{(0,1)}\mathbf{b}_{m,n-1}}{\|\Delta^{(1,0)}\mathbf{b}_{m-1,n} \times \Delta^{(0,1)}\mathbf{b}_{m,n-1}\|}.\end{aligned}$$

V teh točkah imamo torej precej poenostavljeno računanje, vendar moramo biti pazljivi. Če sta, recimo v prvem primeru, vektorja $\Delta^{(1,0)}\mathbf{b}_{0,0}$ in $\Delta^{(0,1)}\mathbf{b}_{0,0}$ linearno odvisna, potem se bo kvocient spremenil v izraz oblike $\frac{0}{0}$. Isto velja tudi za ostale tri primere. To pomeni, da normalni vektorji v točkah $\mathbf{b}_{0,0}$, $\mathbf{b}_{0,n}$, $\mathbf{b}_{m,0}$ in $\mathbf{b}_{m,n}$ v tem primeru niso definirani. Če imamo takšno situacijo, potem pravimo, da je kontrolna točka *degenerirana*. Za več informacij glede normalnih vektorjev in degeneriranih točk glej [5], [12] in [19].

4.8 C^r zveznost

Podobno kot pri Bézierovih zlepkih dveh ali več krivulj bomo tu preučili, kaj se dogaja z zveznostjo, ko zlepimo dve ali več Bézierovih ploskev. Ko imamo opravka s ploskvami, ki imajo veliko število kontrolnih točk, potem se nam splača zlepit več ploskev nižjih stopenj med seboj, ker bi bilo na ta način računanje manj zahtevno. Navedli bomo pogoje za splošno C^r zveznost, $r = 0, 1, \dots$, osredotočili pa se bomo na C^0 in C^1 zveznost ([5]).

Opis geometrijske zveznosti G^r za Bézierove ploskve je dostopen v [5] in ga tukaj ne bomo obravnavali.

Naj bo $\mathbf{x}(u, v)$ Bézierova ploskev, definirana nad domeno $[u_{I-1}, u_I] \times [v_J, v_{J+1}]$ in naj bo $\mathbf{y}(u, v)$ Bézierova ploskev, definirana nad domeno $[u_I, u_{I+1}] \times [v_J, v_{J+1}]$. Potem sta seveda obe ploskvi vsaka zase neskončnokrat zvezno odvedljivi. Kaj pa se dogaja pri krivulji, ki je skupna obema ploskvama? Če želimo, da bo zlepek C^r zvezen, potem morajo biti vsi parcialni odvodi reda r v smeri u enaki v vseh točkah na skupni krivulji:

$$\frac{\partial^r}{\partial u^r} \mathbf{x}(u_I, v) = \frac{\partial^r}{\partial u^r} \mathbf{y}(u_I, v). \quad (4.12)$$

Upoštevajmo sedaj dejstvo, da sta naši ploskvi Bézierovi ploskvi. Naj bo prva definirana s kontrolno mrežo $\{\mathbf{b}_{i,j}\} : 0 \leq i \leq m, 0 \leq j \leq n$, druga pa z mrežo $\{\mathbf{b}_{i,j}\} : m \leq i \leq 2m, 0 \leq j \leq n$. Uporabimo robne odvode, ki smo jih definirali v podpoglavju 4.6. Ker pa moramo preiti iz lokalnih parametrov v (4.11) v globalne parametre, potem se pogoj (4.12) spremeni v

$$\left(\frac{1}{\Delta u_{I-1}}\right)^r \sum_{j=0}^n \Delta^{(r,0)} \mathbf{b}_{m-r,j} B_j^n(v) = \left(\frac{1}{\Delta u_I}\right)^r \sum_{j=0}^n \Delta^{(r,0)} \mathbf{b}_{m,j} B_j^n(v), \quad (4.13)$$

kjer je $\Delta u_I := u_{I+1} - u_I$. Primerjajmo koeficiente v (4.13) pri $B_j^n(v)$:

$$\left(\frac{1}{\Delta u_{I-1}}\right)^r \Delta^{(r,0)} \mathbf{b}_{m-r,j} = \left(\frac{1}{\Delta u_I}\right)^r \Delta^{(r,0)} \mathbf{b}_{m,j}, \quad j = 0, 1, \dots, n. \quad (4.14)$$

Kar dobimo pa je pogoj za \mathcal{C}^r zveznost za Bézierove krivulje. Prišli smo torej do pogojev za \mathcal{C}^r zveznost pri zlepku ploskev.

Trditev 4. *Zlepek dveh ali več Bézierovih ploskev je \mathcal{C}^r zvezen vzdolž skupne krivulje, natanko takrat ko so vse vrstice kontrolnih točk kontrolni poligoni odsekoma Bézierovih krivulj, ki so \mathcal{C}^r zvezne.*

Podobne rezultate bi dosegli v primeru, da bi imeli ploskvi skupno krivuljo v smeri v . S pomočjo teh ugotovitev pogledajmo, kateri so pogoji za \mathcal{C}^0 in \mathcal{C}^1 zveznost.

- Naj bosta dani dve Bézierovi ploskvi, kot smo ju definirali na začetku tega podpoglavja. Zlepek dveh Bézierovih ploskev je \mathcal{C}^0 zvezen vzdolž skupnega roba oz. skupne krivulje $\mathbf{x}(u_I, v) = \mathbf{y}(u_I, v)$, če velja, da so točke $\mathbf{b}_{m,j}$ skupne tako prvi kot drugi ploskvi. V tem primeru se lahko zgodi, da imamo skupno krivuljo, ampak da zlepek ne zgleda "gladek" vzdolž te krivulje. Če želimo imeti tako zveznost kot gladkost zlepka, potem moramo imeti vsaj \mathcal{C}^1 zveznost;
- Če uporabimo (4.14), potem dobimo pogoj za \mathcal{C}^1 zveznost:

$$\mathbf{b}_{m+1,j} = \mathbf{b}_{m,j} + \frac{\Delta u_I}{\Delta u_{I-1}} (\mathbf{b}_{m,j} - \mathbf{b}_{m-1,j}), \quad j = 0, 1, \dots, n.$$

To pomeni, da morajo biti kontrolne točke $\mathbf{b}_{m-1,j}, \mathbf{b}_{m,j}, \mathbf{b}_{m+1,j}$ kolinearne in biti v razmerju $\Delta u_{I-1} : \Delta u_I$.

Zgled 4.5

Izberimo Bézierovi ploskvi, ki sta obe stopnje (2, 3). Izberimo takšno domeno, da bo veljalo $\Delta u_{I-1} : \Delta u_I = \frac{2}{3} : \frac{1}{3}$. Ta domena je prikazana na sliki 4.8. To pomeni, da bodo točke $\mathbf{b}_{1,j}, \mathbf{b}_{2,j}$ in $\mathbf{b}_{3,j}$ za vsak j kolinearne in v razmerju $\frac{2}{3} : \frac{1}{3}$. To dobro vidimo na sliki 4.7, kjer sta prikazani kontrolni mreži obeh ploskev. Na sliki 4.6 pa sta prikazani ploskvi, ki ju definirata kontrolni mreži na sliki 4.7.

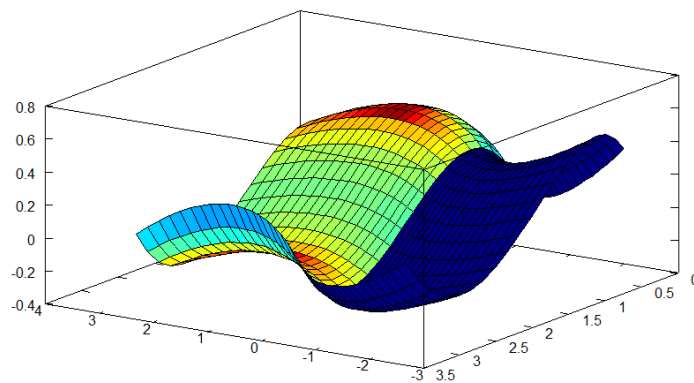
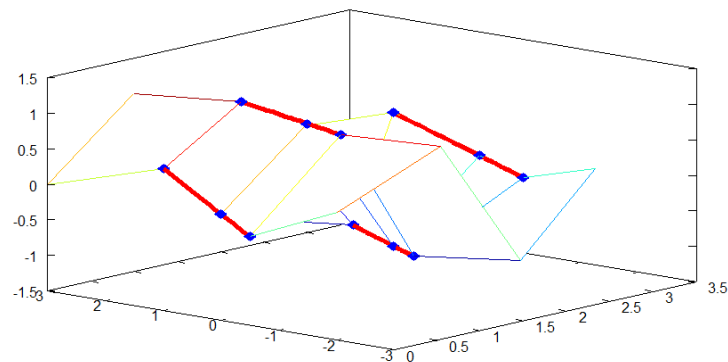
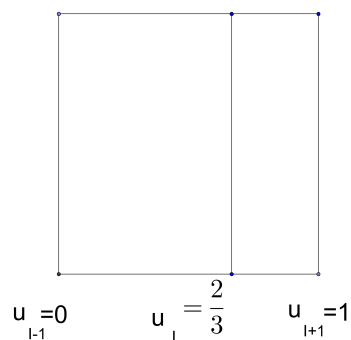
Veliko informacij glede zlepkov Bézierovih ploskev in njihove uporabe v industrijskem svetu je dostopnih v [2], [9] in [15].

4.9 Racionalne Bézierove ploskve

Ali je možno preoblikovati Bézierovo ploskev tako, da *ne* vplivamo na kontrolno mrežo, kot smo to storili pri višanju stopnje? Podobno kot pri krivuljah nam odgovor na to vprašanje ponujajo *racionalne Bézierove ploskve*. Definirali jih bomo s pomočjo Bernsteinovih polinomov in pokazali, kako jih lahko preoblikujemo. Več informacij je na voljo v [5].

Definicija 23. *Racionalna Bézierova ploskev stopnje (m, n) je definirana kot*

$$\mathbf{x}(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n \omega_{i,j} \mathbf{b}_{i,j} B_i^m(u) B_j^n(v)}{\sum_{i=0}^m \sum_{j=0}^n \omega_{i,j} B_i^m(u) B_j^n(v)}, \quad \mathbf{x}(u, v), \mathbf{b}_{i,j} \in \mathbb{E}^3, \omega_{i,j} \in \mathbb{R}. \quad (4.15)$$

Slika 4.6: Prikaz dveh Bézierovih ploskev stopnje (2, 3), ki skupaj tvorita C^1 zlepek.Slika 4.7: Kontrolni mreži dveh Bézierovih ploskev, ki skupaj tvorita C^1 zlepek.

Slika 4.8: Domena zlepka.

Izraz (4.15) je podoben Bernsteinovi obliki Bézierovih ploskev. Tu nastopajo realna števila $\omega_{i,j}$, ki jih imenujemo *uteži*, in igrajo isto vlogo kot pri krivuljah. Vsaka utež $\omega_{i,j}$ vpliva na ustrezno kontrolno točko $\mathbf{b}_{i,j}$:

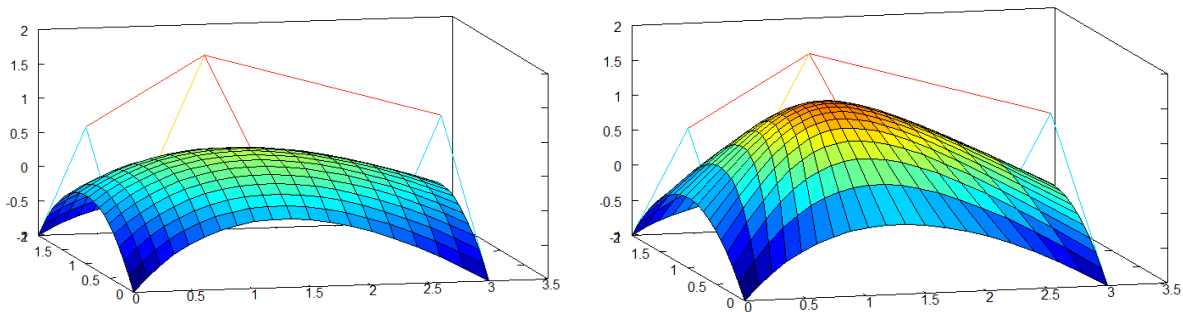
- če je $\omega_{i,j} > 1$, potem se ploskev približa pripadajoči kontrolni točki $\mathbf{b}_{i,j}$;
- če je $\omega_{i,j} < 1$, potem se ploskev oddalji od pripadajoče kontrolne točke $\mathbf{b}_{i,j}$;
- če velja $\omega_{i,j} = 1, \forall i, j$, potem (4.15) definira polinomsko Bézierovo ploskev. Polinomsko ploskev dobimo tudi takrat, ko so vse uteži enake istemu poljubnemu realnemu

številu. Uteži pa ne smejo biti vse enake nič, ker bi potem v (4.15) imeli deljenje z nič.

Opomba: Če so vse uteži pozitivne, veljajo za racionalne Bézierove ploskve lastnosti, ki smo jih srečali v podpoglavju 4.4. Če pa je katera izmed uteži negativna, potem je treba biti pazljivi, saj iste lastnosti ne držijo več.

Zgled 4.6

Vzemimo racionalno Bézierovo ploskev stopnje (2, 2). Na sliki 4.9 imamo na levi racionalno Bézierovo ploskev, kjer velja $\omega_{i,j} = 1, \forall i, j$, in je torej polinomska ploskev. Na desni pa je ista ploskev, le da vzamemo $\omega_{1,1} = 5$, kar povzroči, da se ploskev približa kontrolni točki $\mathbf{b}_{1,1}$.



Slika 4.9: Primerjava med polinomsko Bézierovo ploskvijo in racionalno Bézierovo ploskvijo, kjer je $\omega_{1,1} = 5$.

4.10 Matrična oblika Bézierove ploskve

Zapišimo Bézierovo ploskev (4.5) v matrični obliki. Naj bo \mathbf{u} vektor vseh potenc parametra u dolžine $m + 1$:

$$\mathbf{u} = [u^m, u^{m-1}, \dots, u^1, u^0]^T.$$

Podobno, naj bo \mathbf{v} vektor vseh potenc parametra v dolžine $n + 1$:

$$\mathbf{v} = [v^n, v^{n-1}, \dots, v^1, v^0]^T.$$

Definirajmo sedaj matriko P velikosti $(m+1) \times (m+1)$, matriko Q velikosti $(n+1) \times (n+1)$ in matriko B velikosti $(m+1) \times (n+1)$ na sledeči način:

$$P_{i+1,k+1} = (-1)^{m-k-i} \binom{m-i}{k} \binom{m}{i}, \quad i = 0, \dots, m, \quad k = 0, \dots, m,$$

$$Q_{j+1,\ell+1} = (-1)^{n-\ell-j} \binom{n-j}{\ell} \binom{n}{j}, \quad j = 0, \dots, n, \quad \ell = 0, \dots, n,$$

$$B_{i+1,j+1} = \mathbf{b}_{i,j}, \quad i = 0, \dots, m, \quad j = 0, \dots, n.$$

Potem lahko točko na Bézierovi ploskvi za dana parametra u in v izračunamo kot

$$\mathbf{b}^{m,n}(u, v) = \mathbf{u}^T \cdot P \cdot B \cdot Q \cdot \mathbf{v}. \quad (4.16)$$

Dokaz. Velja

$$\begin{aligned} \mathbf{b}^{m,n}(u, v) &= \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} B_i^m(u) B_j^n(v) = \\ &= \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} \binom{m}{i} u^i (1-u)^{m-i} \binom{n}{j} v^j (1-v)^{n-j}. \end{aligned} \quad (4.17)$$

Podobno kot v prejšnjem poglavju bomo izrazili elementa $(1-u)^{m-i}$ in $(1-v)^{n-j}$ z binomskim izrekom in dobljene vsote vstavili v (4.17):

$$\begin{aligned} \mathbf{b}^{m,n}(u, v) &= \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} \binom{m}{i} u^i \left(\sum_{k=0}^{m-i} \binom{m-i}{k} (-1)^{m-k-i} u^{m-k-i} \right) \cdot \\ &\cdot \binom{n}{j} v^j \left(\sum_{\ell=0}^{n-j} \binom{n-j}{\ell} (-1)^{n-\ell-j} v^{n-\ell-j} \right). \end{aligned}$$

Elemente izven oklepajev damo v vsote in dobimo:

$$\begin{aligned} \mathbf{b}^{m,n}(u, v) &= \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} \left(\sum_{k=0}^{m-i} \binom{m-i}{k} \binom{m}{i} (-1)^{m-k-i} u^{m-k} \right) \cdot \\ &\cdot \left(\sum_{\ell=0}^{n-j} \binom{n-j}{\ell} \binom{n}{j} (-1)^{n-\ell-j} v^{n-\ell} \right). \end{aligned}$$

Preoblikujmo in dobimo:

$$\mathbf{b}^{m,n}(u, v) = \sum_{i=0}^m \sum_{j=0}^n \left(\sum_{k=0}^{m-i} \binom{m-i}{k} \binom{m}{i} (-1)^{m-k-i} u^{m-k} \right). \quad (4.18)$$

$$\cdot \mathbf{b}_{i,j} \cdot \left(\sum_{\ell=0}^{n-j} \binom{n-j}{\ell} \binom{n}{j} (-1)^{n-\ell-j} v^{n-\ell} \right). \quad (4.19)$$

Tako smo dobili željeni rezultat 4.16. □

Opomba: Matriki P in Q sta trikotni, saj sta pri indeksih v spodnjem delu matrik pod diagonalo elementa $\binom{m-i}{k}$ in $\binom{n-j}{\ell}$ enaka nič.

Zgled 4.7

Vzemimo podatke iz zgleda 4.2. Naj bodo kontrolne točke Bézierove ploskve enake

$$\begin{bmatrix} \mathbf{b}_{0,0} \\ \mathbf{b}_{1,0} \\ \mathbf{b}_{2,0} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,1} \\ \mathbf{b}_{1,1} \\ \mathbf{b}_{2,1} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,2} \\ \mathbf{b}_{1,2} \\ \mathbf{b}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 1 \\ 3 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 2 \\ 3 \\ 3 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ -1 \\ 1 \\ 2 \\ 0 \\ 3 \\ 2 \\ -1 \end{bmatrix}$$

in naj bosta $u = 0.3$ in $v = 0.5$ parametra. Dobimo naslednje podatke:

$$\begin{aligned} \mathbf{u} &= [0.09, 0.3, 1]^T, \quad \mathbf{v} = [0.25, 0.5, 1]^T, \\ B &= \begin{bmatrix} \mathbf{b}_{0,0} & \mathbf{b}_{0,1} & \mathbf{b}_{0,2} \\ \mathbf{b}_{1,0} & \mathbf{b}_{1,1} & \mathbf{b}_{1,2} \\ \mathbf{b}_{2,0} & \mathbf{b}_{2,1} & \mathbf{b}_{2,2} \end{bmatrix}, \quad P = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \end{aligned}$$

Izračunajmo točko na Bézierovi ploskvi za dana parametra u in v :

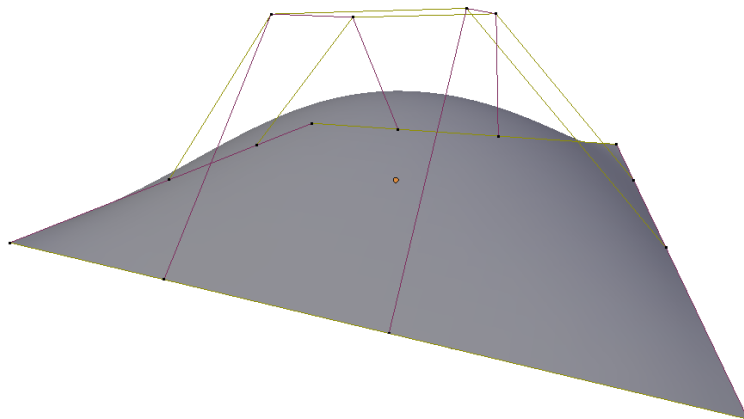
$$\mathbf{b}^{2,2}(0.3, 0.5) = \mathbf{u}^T \cdot P \cdot B \cdot Q \cdot \mathbf{v} = \begin{bmatrix} 0.69 \\ 1 \\ 0.525 \end{bmatrix}.$$

Dobili smo isti rezultat kot v zgledu 4.2, ko smo ga izračunali z de Casteljauovim algoritmom.

4.11 Računalniško oblikovanje z Bézierovimi ploskvami

Ko računalniško oblikujemo objekte, lahko poljubno izberemo metodo dela. Nekatere predmete raje oblikujemo s krivuljami kot s ploskvami, ker imamo manj dela. V tem pod poglavju bomo pokazali, kako preoblikovati *stekleno posodo in mizo* samo z Bézierovimi ploskvami. Opisali bomo, kako lahko z uporabo ene same Bézierove ploskve, modeliramo posodo. To pomeni, da bomo z Blenderjem skonstruirali ploskev in jo preoblikovali do posode. Nato bomo povedali nekaj besed glede zveznosti in zlepkov.

Ko z ustreznim ukazom narišemo racionalno Bézierovo ploskev, Blender generira ploskev stopnje $(3, 3)$, kar pomeni, da bomo imeli kontrolno mrežo s 16 kontrolnimi točkami (slika 4.10).

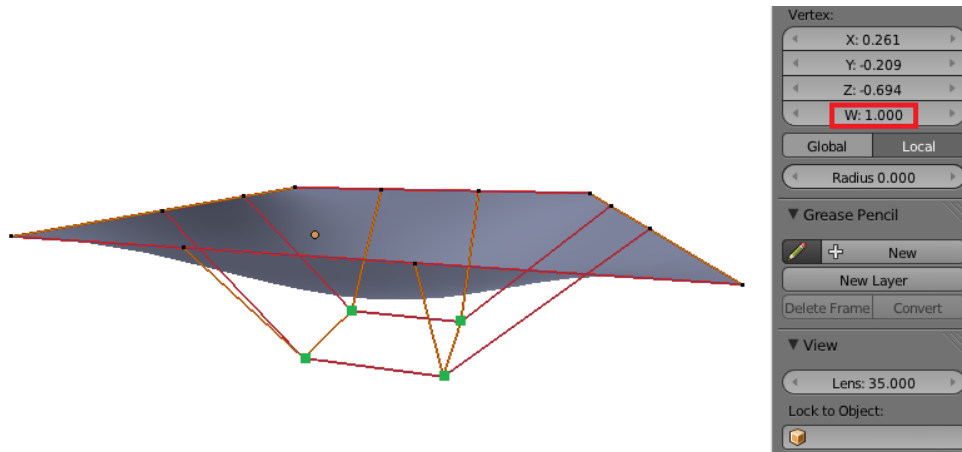


Slika 4.10: Bézierova ploskev stopnje $(3, 3)$.

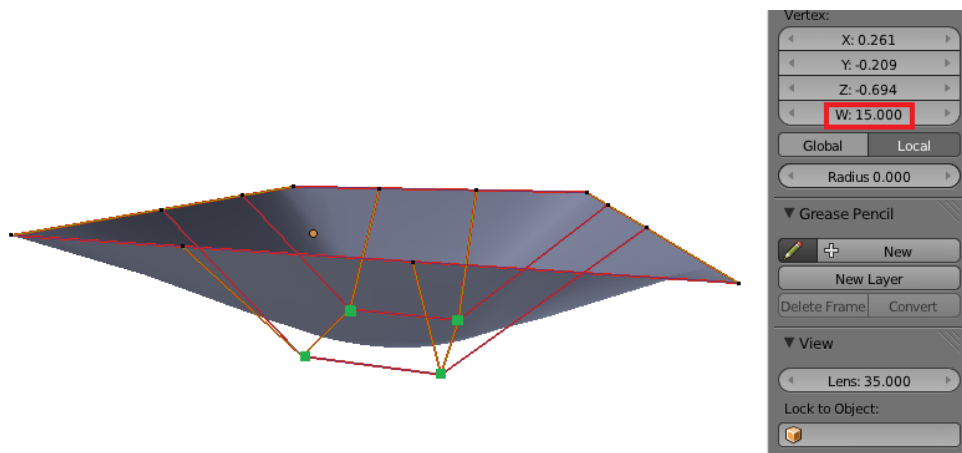
Ploskev zarotiramo za 180° , da bomo imeli boljši pregled nad našim delom. Poglejmo si zeleno obarvane točke na sliki 4.11.

Ker je Bézierova ploskev racionalna, imamo na voljo uteži, ki nam pomagajo pri oblikovanju. V našem primeru bi radi vsem štirim kontrolnim točkam na sliki 4.11 zvišali utež, tako da bi se ploskev približala pripadajočim točkam in bi tako preoblikovali dno posode. Rezultat je prikazan na sliki 4.12.

Da bo končni rezultat bolj realističen, bomo dodali robove oz. ročaje posode. Označimo vse štiri kontrolne točke poljubne robne krivulje. Z ukazom *Extrude* se ploskev pri označenih točkah “podaljša” in dobimo nove točke, ki so na sliki 4.13 označene z rdečo barvo. Isti rezultat bi dobili, če bi dodali novo ploskev, jo ustrezno preoblikovali in ustvarili zlepek s prvo ploskvijo. V tem primeru bi morali označiti ustrezne stične točke in jim določiti

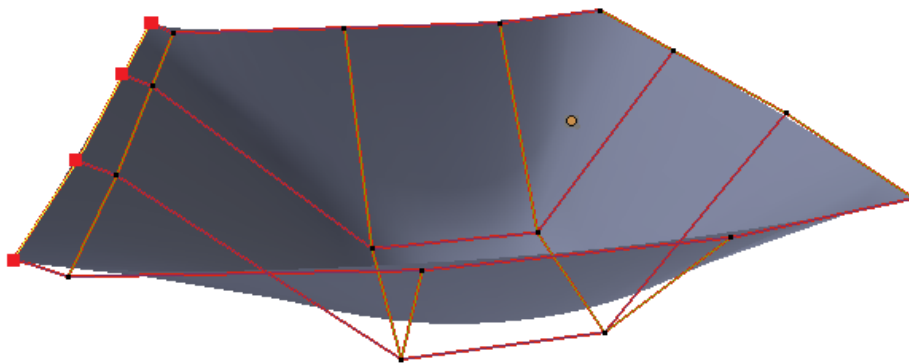


Slika 4.11: Racionalna Bézierova ploskev, ki ima vse uteži enake ena.



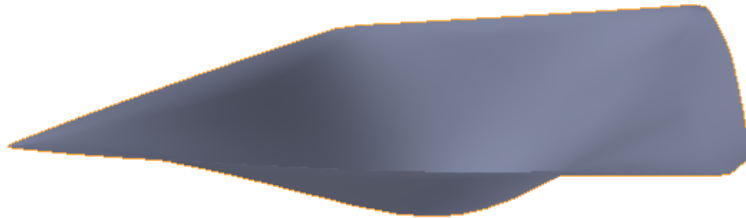
Slika 4.12: Višanje uteži vsem štirim označenim točkam.

iste koordinate, da bi dosegli C^0 oz. G^0 zveznost. Več o tem bomo povedali na koncu podpoglavja.



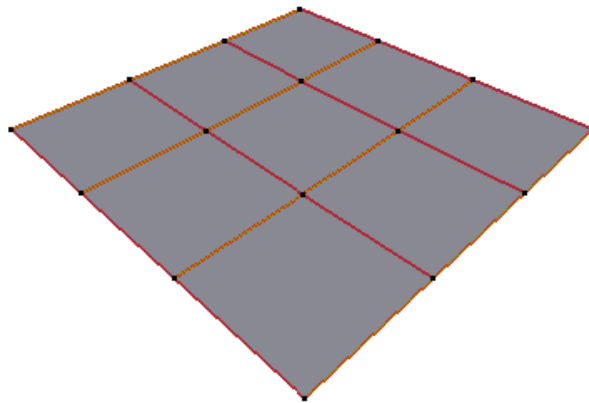
Slika 4.13: Nove točke bomo uporabili, da bomo oblikovali ročaje posode.

Ko to operacijo izvedemo na vseh robnih krivuljah, dobimo ploskev, ki predstavlja posodo s preoblikovanim dnom in ročaji (slika 4.14).



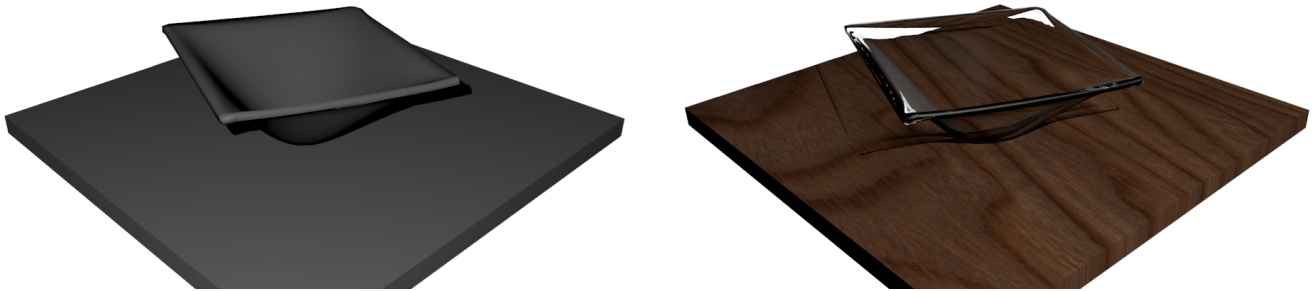
Slika 4.14: Posoda ima sedaj že oblikovano dno in ročaje.

Ko smo zaključili z modeliranjem posode, lahko dodamo novo racionalno Bézierovo ploskev, s katero bomo oblikovali *mizo*. Naloga je zelo preprosta, ker je dovolj označiti štiri kontrolne točke, ki niso na isti višini kot ostale (zeleno obarvane točke na sliki 4.11), in jih znižati (z operacijo premikanja v Blenderju) do višine ostalih kontrolnih točk. Rezultat je na sliki 4.15.



Slika 4.15: Bézierova ploskev stopnje (3, 3), ki predstavlja mizo.

Sedaj lahko posodo premaknemo na nastalo mizo in za oba objekta uporabimo Blenderjev ukaz *Solidify*, ki danemu objektu doda debelino. Slika 4.16 prikazuje končni rezultat, kateremu potem dodamo nekaj barv in lastnosti: posode ne pobarvamo, ampak ji dodamo prozornost, mizo pa obarvamo z ustreznimi barvami, da dobimo efekt lesa. Na celotno sceno dodamo luč, ki osvetli objekte in ustvari sence.



Slika 4.16: Prikaz posode na mizi, potem ko dodamo debeline, barve, prozornost in luč.

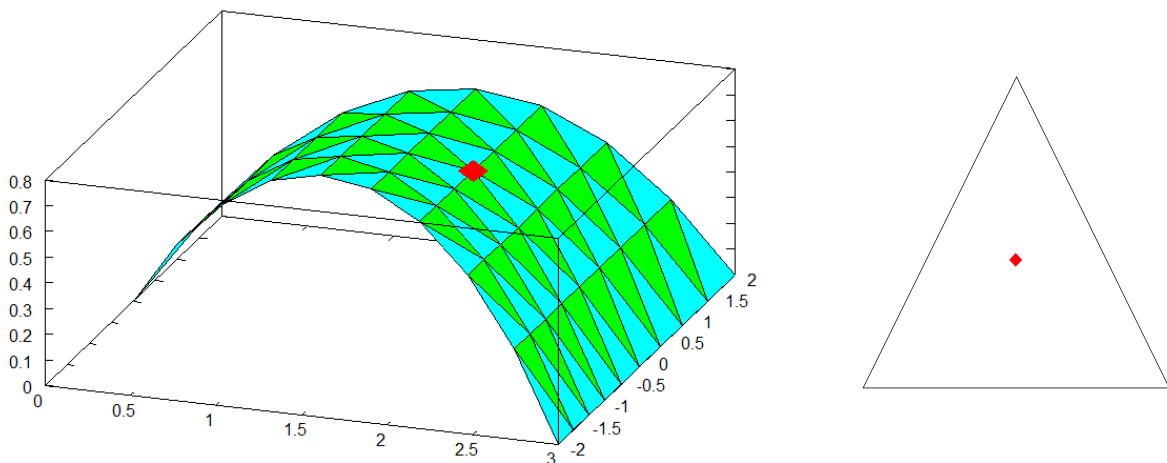
Opomba: Pri oblikovanju posode in mize nismo uporabljali zlepkov in zveznosti, ker smo imeli opravka samo z eno Bézierovo ploskvijo. Kar se tiče zveznosti v Blenderju velja podobno kot pri krivuljah. Če dodamo dve Bézierovi ploskvi in ju hočemo zlepiti, potem je enostavno določiti \mathcal{C}^0 oz. G^0 zveznost, saj moramo samo označiti kontrolne točke ustrezne robne krivulje in jim določiti takšne koordinate, da bosta ploskvi imeli skupno robno krivuljo. Določitev \mathcal{C}^1 zveznosti pa je bolj zahtevna naloga, ker bi morali paziti na kolinearnost točk pri skupni robni krivulji. S prostim očesom tega ne moremo narediti, vsekakor pa se lahko s prostim očesom približamo željeni rešitvi.

Poglavje 5

Bézierovi trikotniki

Ko je de Casteljau v avtomobilski hiši Citroën odkril Bézierove krivulje, je ugotovil, da lahko uporabi podoben koncept, da bi opisal ploskve. Ploskve, ki jih je opisal, niso bile v obliki tenzorskega produkta. Definiral jih je po bolj "naravni" poti s pomočjo svojega algoritma, le da ga je malo preoblikoval in posplošil s pomočjo trikotnikov. Obravnavana snov v tem poglavju je povzeta po [5]. Med matematike, ki so se ukvarjali z Bézierovimi trikotniki, spadajo R. Barnhill ([1]), C. de Boor ([3]), G. Farin ([7]), L. Frederickson ([11]), M. Sabin ([17]), P. Sablonnière ([18])...

V prejšnjem poglavju smo obravnavali parametrične ploskve, ki so bile slika domene pravokotne oblike. Bézierovi trikotniki so primer parametričnih ploskev, kjer imamo opravka z domenami trikotnih oblik. Na sliki 5.1 se rdeče označena točka v domeni preslika v označeno točko na ploskvi.



Slika 5.1: Primer Bézierovega trikotnika in njegove trikotne domene.

Pri Bézierovih trikotnikih imamo nekaj razlik glede notacije, kar bomo predstavili v naslednjem podpoglavju.

5.1 de Casteljauov algoritem

Uporabili bomo nekaj konceptov, ki smo jih srečali v podpoglavju 2.5, ko smo govorili o baricentričnih koordinatah pri trikotnikih v ravnini. Končen algoritem je posplošitev

podobnega algoritma pri krivuljah, pri čemer so največje razlike v notaciji. Ko smo govorili o de Casteljauovem algoritmu za Bézierove krivulje, smo uvedli pojem de Casteljauove sheme. Tu bomo imeli podobno shemo, le da bo v obliki trikotnika, in bo treba biti pazljivi na indekse kontrolnih točk. Pri Bézierovih trikotnikih ne bomo imeli opravka s ploskvami stopnje (m, n) kot pri tenzorskem produktu, ampak imamo tu opravka le s parametrom n . Rekli bomo torej, da je Bézierov trikotnik stopnje n . Kontrolna mreža bo v tem primeru imela $\binom{n+2}{2}$ kontrolnih točk, ki jih zapišemo v trikotno shemo:

$$\begin{array}{ccccccc}
 & & & & \mathbf{b}_{(0,n,0)} & & \\
 & & & & & & \\
 & & & & \mathbf{b}_{(0,n-1,1)} & & \mathbf{b}_{(1,n-1,0)} \\
 & & \cdot \cdot & & \vdots & & \cdot \cdot \\
 & & & & & & \\
 \mathbf{b}_{(0,0,n)} & & \cdots & & \cdots & & \mathbf{b}_{(n,0,0)}
 \end{array}$$

Preden opišemo de Casteljauov algoritem, uvedimo nekaj notacije. Kontrolne točke, kot smo že videli, bomo predstavili z indeksom (i, j, k) , kar bomo na kratko pisali \mathbf{i} . Definirajmo dolžino indeksa kot

$$|\mathbf{i}| := i + j + k, \quad i, j, k \geq 0.$$

Točke v ravnini bomo predstavljali z baricentričnimi koordinatami glede na oglišča trikotnika, kar bomo pisali kot $\mathbf{u} = (u, v, w)$, $u + v + w = 1$.

de Casteljauov algoritem

Naj bodo $\mathbf{b}_{\mathbf{i}} \in \mathbb{E}^3$, $|\mathbf{i}| = n$, kontrolne točke. Naj bo \mathbf{u} točka v ravnini v baricentričnih koordinatah. Korak de Casteljauovega algoritma se glasi:

$$\mathbf{b}_{\mathbf{i}}^r(\mathbf{u}) = u \cdot \mathbf{b}_{\mathbf{i}+\mathbf{e}_1}^{r-1}(\mathbf{u}) + v \cdot \mathbf{b}_{\mathbf{i}+\mathbf{e}_2}^{r-1}(\mathbf{u}) + w \cdot \mathbf{b}_{\mathbf{i}+\mathbf{e}_3}^{r-1}(\mathbf{u}), \quad r = 1, 2, \dots, n, \quad |\mathbf{i}| = n - r, \quad (5.1)$$

kjer je

$$\mathbf{e}_1 = (1, 0, 0), \quad \mathbf{e}_2 = (0, 1, 0), \quad \mathbf{e}_3 = (0, 0, 1).$$

Pri tem velja $\mathbf{b}_{\mathbf{i}}^0(\mathbf{u}) := \mathbf{b}_{\mathbf{i}}$. Potem je $\mathbf{b}_{(0,0,0)}^n(\mathbf{u})$ točka na Bézierovem trikotniku \mathbf{b}^n pri danem parametru \mathbf{u} . Iz podpoglavja 2.5 vemo, da so $u, v, w \geq 0$, če je točka \mathbf{u} znotraj trikotnika, ki predstavlja domeno.

V prilogi D je implementacija de Casteljauovega algoritma za Bézierove trikotnike.

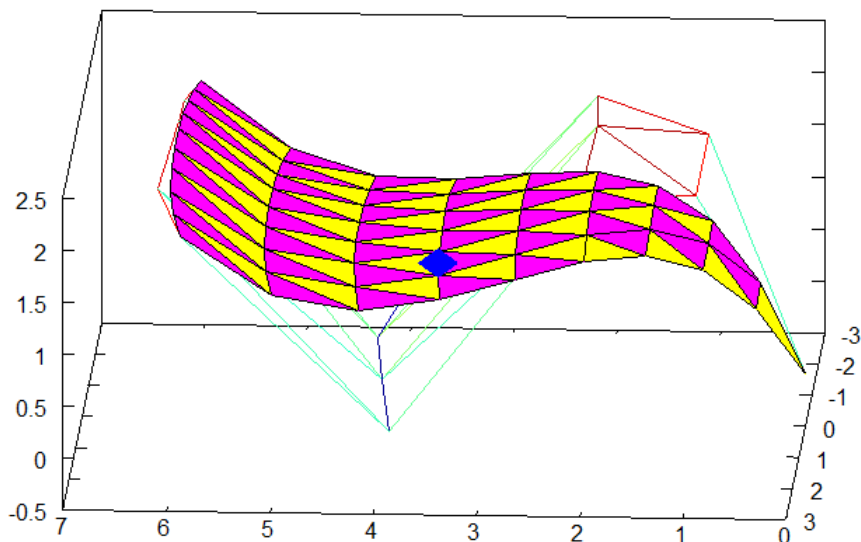
po tretjem

$$\begin{bmatrix} 4.14 \\ -0.34 \\ 1.2 \end{bmatrix} \begin{bmatrix} 2.33 \\ 0.66 \\ 1.42 \end{bmatrix} \begin{bmatrix} 4.07 \\ 1.17 \\ 1.14 \end{bmatrix}$$

in končno dobimo v zadnjem koraku točko na ploskvi pri danem parametru \mathbf{u} :

$$\mathbf{b}_{(0,0,0)}^4(\mathbf{u}) = \begin{bmatrix} 3.5 \\ 0.75 \\ 1.24 \end{bmatrix}.$$

Na sliki 5.2 je narisan Bézierov trikotnik iz zgleda 4.1 in njegova kontrolna mreža. Točko $\mathbf{b}_{(0,0,0)}^4\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right)$, ki smo jo izračunali v zgledu, je na ploskvi označena z modrim kvadratom.



Slika 5.2: Bézierov trikotnik s kontrolno mrežo.

5.2 Bernsteinovi polinomi

Bézierove krivulje in ploskve smo definirali tudi s pomočjo Bernsteinovih polinomov, ki so bili odvisni le od parametra t . Zapisali bomo tudi Bézierove trikotnike s pomočjo Bernsteinovih polinomov, ki pa so drugače definirani.

Definicija 24. Naj bo $\mathbf{u} = (u, v, w)$ parameter in označimo trojico (i, j, k) , $i + j + k = n$, z \mathbf{i} . Bernsteinovi polinomi so polinomi oblike

$$B_{\mathbf{i}}^n(\mathbf{u}) := \binom{n}{\mathbf{i}} u^i v^j w^k, \quad \binom{n}{\mathbf{i}} := \begin{cases} \frac{n!}{i!j!k!}, & 0 \leq i, j, k \leq n, \\ 0, & \text{sicer.} \end{cases}$$

Čeprav bi sprva rekli, da so Bernsteinovi polinomi odvisni od treh parametrov, so v resnici odvisni le od dveh parametrov u in v , saj je $w = 1 - u - v$. Bernsteinove polinome lahko zapišemo v shemo trikotne oblike s pomočjo de Casteljauove trikotne sheme.

Zgled 5.2

Izberimo $n = 3$. De Casteljauova trikotna shema se glasi:

$$\begin{array}{ccccccc}
 & & & & \mathbf{b}_{(0,3,0)} & & \\
 & & & & & & \\
 & & & & \mathbf{b}_{(0,2,1)} & & \mathbf{b}_{(1,2,0)} \\
 & & & & & & \\
 & & & & \mathbf{b}_{(0,1,2)} & & \mathbf{b}_{(1,1,1)} & & \mathbf{b}_{(2,1,0)} \\
 & & & & & & & & \\
 & & & & \mathbf{b}_{(0,0,3)} & & \mathbf{b}_{(1,0,2)} & & \mathbf{b}_{(2,0,1)} & & \mathbf{b}_{(3,0,0)}.
 \end{array}$$

Bodimo pozorni na indekse (i, j, k) kontrolnih točk. Potem lahko Bernsteinove polinome zapišemo v spodnjo shemo:

$$\begin{array}{ccccccc}
 & & & & v^3 & & \\
 & & & & & & \\
 & & & & 3v^2w & & 3uv^2 \\
 & & & & & & \\
 & & & & 3vw^2 & & 6uvw & & 3u^2v \\
 & & & & & & & & \\
 & & & & w^3 & & 3uw^2 & & 3u^2w & & u^3.
 \end{array}$$

Navedimo nekaj lastnosti Bernsteinovih polinomov:

1. Naj bo $\binom{n}{\mathbf{i}}$ koeficient, ki nastopa v Bernsteinovih polinomih. Potem velja:

$$\binom{n}{\mathbf{i}} = 1 \Leftrightarrow \mathbf{i} \in \{(n, 0, 0), (0, n, 0), (0, 0, n)\}.$$

Dokaz.

(\Leftarrow): brez izgube za splošnost predpostavimo, da je $\mathbf{i} = (n, 0, 0)$. Potem velja:

$$\binom{n}{\mathbf{i}} = \frac{n!}{n!0!0!} = 1.$$

(\Rightarrow): upoštevajoč, da je $|\mathbf{i}| = n$, zapišimo koeficient kot

$$\binom{n}{\mathbf{i}} = \binom{i+j}{j} \cdot \binom{i+j+k}{k} = \binom{i+j}{j} \cdot \binom{n}{k} = 1. \quad (5.2)$$

Produkt v (5.2) bo enak ena, natanko takrat ko bosta oba člena enaka ena. Analizirajmo naslednje enakosti:

$$\binom{n}{k} = 1 \Leftrightarrow k \in \{0, n\}.$$

- V primeru, da je $k = n$, dobimo:

$$\binom{i+j}{j} = 1 \Leftrightarrow \binom{0}{j} = 1 \Leftrightarrow j = 0.$$

Potem mora biti tudi $i = 0$ in zaželjeni rezultat sledi.

- Če pa je $k = 0$, mora veljati

$$\binom{i+j}{j} = 1 \Leftrightarrow \binom{n}{j} = 1 \Leftrightarrow j \in \{0, n\}.$$

V primeru, da je $j = 0$, sledi $i = n$. Če pa je $j = n$, potem je $i = 0$.

□

2. Koefficient $\binom{n}{\mathbf{i}}$ lahko zapišemo kot

$$\binom{n}{\mathbf{i}} = \binom{n-1}{\mathbf{i}-\mathbf{e}_1} + \binom{n-1}{\mathbf{i}-\mathbf{e}_2} + \binom{n-1}{\mathbf{i}-\mathbf{e}_3}. \quad (5.3)$$

Dokaz.

$$\begin{aligned} \binom{n}{\mathbf{i}} &= \frac{n!}{i!j!k!} = \frac{(n-1)!(i+j+k)}{i!j!k!} = \frac{(n-1)!}{(i-1)!j!k!} + \frac{(n-1)!}{i!(j-1)!k!} + \frac{(n-1)!}{i!j!(k-1)!} \\ &= \binom{n-1}{\mathbf{i}-\mathbf{e}_1} + \binom{n-1}{\mathbf{i}-\mathbf{e}_2} + \binom{n-1}{\mathbf{i}-\mathbf{e}_3}. \end{aligned}$$

□

3. Bernsteinove polinome lahko izračunamo rekurzivno:

$$B_{\mathbf{i}}^n(\mathbf{u}) = uB_{\mathbf{i}-\mathbf{e}_1}^{n-1}(\mathbf{u}) + vB_{\mathbf{i}-\mathbf{e}_2}^{n-1}(\mathbf{u}) + wB_{\mathbf{i}-\mathbf{e}_3}^{n-1}(\mathbf{u}).$$

Dokaz. Uporabimo enakost (5.3):

$$\begin{aligned} B_{\mathbf{i}}^n(\mathbf{u}) &= \binom{n}{\mathbf{i}} u^i v^j w^k \\ &= u \cdot \binom{n-1}{\mathbf{i}-\mathbf{e}_1} u^{i-1} v^j w^k + v \cdot \binom{n-1}{\mathbf{i}-\mathbf{e}_2} u^i v^{j-1} w^k + w \cdot \binom{n-1}{\mathbf{i}-\mathbf{e}_3} u^i v^j w^{k-1} \\ &= uB_{\mathbf{i}-\mathbf{e}_1}^{n-1}(\mathbf{u}) + vB_{\mathbf{i}-\mathbf{e}_2}^{n-1}(\mathbf{u}) + wB_{\mathbf{i}-\mathbf{e}_3}^{n-1}(\mathbf{u}). \end{aligned}$$

□

Dokaza naslednjih dveh trditev bomo izpustili. Dostopna sta v [5] in [20].

Trditev 5. Bézierov trikotnik lahko definiramo s pomočjo Bernsteinovih polinomov:

$$\mathbf{b}(\mathbf{u}) = \sum_{|\mathbf{i}|=n} \mathbf{b}_{\mathbf{i}} B_{\mathbf{i}}^n(\mathbf{u}).$$

Trditev 6. Vmesne točke $\mathbf{b}_{\mathbf{i}}^r(\mathbf{u})$ de Casteljaouvega algoritma lahko izrazimo z Bernsteinovi polinomi:

$$\mathbf{b}_{\mathbf{i}}^r(\mathbf{u}) = \sum_{|\mathbf{j}|=r} \mathbf{b}_{\mathbf{i}+\mathbf{j}} B_{\mathbf{j}}^r(\mathbf{u}), \quad |\mathbf{i}| = n - r.$$

5.3 Pomembne lastnosti Bézierovih trikotnikov

V tem podpoglavju si bomo pogledali lastnosti, ki smo jih že srečali v prejšnjih poglavjih, in sicer z vidika Bézierovih trikotnikov.

1. Bézierovi trikotniki so invariantni za afine preslikave.

Dokaz. Ker je de Casteljauov algoritem za Bézierove trikotnike zaporedje linearnih interpolacij in ker smo dokazali, da je linearna interpolacija invariantna za afine preslikave, lahko zaključimo, da so Bézierovi trikotniki invariantni za afine preslikave.

To pomeni, da lahko izračunamo točko na Bézierovem trikotniku in jo afino preslikamo ali pa najprej afino preslikamo kontrolno mrežo in nato izračunamo točko na ploskvi. Rezultat bo isti. □

2. Bézierov trikotnik leži znotraj konveksne ovojnice svoje kontrolne mreže.

Dokaz. V de Casteljauovem algoritmu je točka $\mathbf{b}_i^r(\mathbf{u})$ konveksna kombinacija prejšnjih točk $\mathbf{b}_{i+\mathbf{e}_1}^{r-1}(\mathbf{u})$, $\mathbf{b}_{i+\mathbf{e}_2}^{r-1}(\mathbf{u})$ in $\mathbf{b}_{i+\mathbf{e}_3}^{r-1}(\mathbf{u})$, saj velja $0 \leq u, v, w \leq 1$ (glej podpoglavje 1.1). □

3. Oglišča trikotne domene se preslikajo v oglišča Bézierovega trikotnika oz. v oglišča kontrolne mreže.

Dokaz. Vzemimo vsa tri oglišča trikotne domene $(1, 0, 0)$, $(0, 1, 0)$ in $(0, 0, 1)$ ter izračunajmo Bézierove točke pri teh parametrih s pomočjo Bernsteinovih polinomov:

$$\begin{aligned}\mathbf{b}((1, 0, 0)) &= \sum_{|\mathbf{i}|=n} \mathbf{b}_i \binom{n}{\mathbf{i}} 1^i \cdot 0^j \cdot 0^k = \mathbf{b}_{(n,0,0)}, \\ \mathbf{b}((0, 1, 0)) &= \sum_{|\mathbf{i}|=n} \mathbf{b}_i \binom{n}{\mathbf{i}} 0^i \cdot 1^j \cdot 0^k = \mathbf{b}_{(0,n,0)}, \\ \mathbf{b}((0, 0, 1)) &= \sum_{|\mathbf{i}|=n} \mathbf{b}_i \binom{n}{\mathbf{i}} 0^i \cdot 0^j \cdot 1^k = \mathbf{b}_{(0,0,n)}.\end{aligned}$$

□

4. Robne krivulje Bézierovih trikotnikov.

Pri tenzorskem produktu smo robne krivulje dobili tako, da smo vzeli parametra $u = 0$ oz. $v = 0$. Tu bomo uporabili isti pristop za parametre (u, v, w) . Uporabimo korak de Casteljauovega algoritma in dobimo

$$\mathbf{b}_i^r(0, v, w) = v\mathbf{b}_{i+\mathbf{e}_2}^{r-1} + w\mathbf{b}_{i+\mathbf{e}_3}^{r-1}, \quad v + w = 1. \quad (5.4)$$

Opazimo, da je (5.4) korak de Casteljauovega algoritma za Bézierove krivulje. Podobno računamo

$$\mathbf{b}_i^r(u, 0, w) = u\mathbf{b}_{i+\mathbf{e}_1}^{r-1} + w\mathbf{b}_{i+\mathbf{e}_3}^{r-1}, \quad u + w = 1,$$

in

$$\mathbf{b}_i^r(u, v, 0) = u\mathbf{b}_{i+\mathbf{e}_1}^{r-1} + v\mathbf{b}_{i+\mathbf{e}_2}^{r-1}, \quad u + v = 1.$$

Robne krivulje so torej Bézierove krivulje določene z robnimi kontrolnimi točkami.

5.4 Višanje stopnje

Recimo, da bi želeli imeti bolj fleksibilne ploskve, da bi jih lahko bolj oblikovali. Podobno kot pri tenzorskem produktu bomo tukaj zvišali stopnjo Bézierovega trikotnika stopnje n .

Trditev 7. Naj bo \mathbf{b}^n Bézierov trikotnik stopnje n s kontrolnimi točkami \mathbf{b}_i , $\mathbf{i} = (i, j, k)$. Pri zvišanju stopnje so nove kontrolne točke Bézierovega trikotnika definirane kot

$$\mathbf{b}_i^{(1)} := \frac{1}{n+1} (i\mathbf{b}_{i-\mathbf{e}_1} + j\mathbf{b}_{i-\mathbf{e}_2} + k\mathbf{b}_{i-\mathbf{e}_3}). \quad (5.5)$$

Dokaz. Želimo zvišati stopnjo Bézierovega trikotnika, pri čemer naj le-ta ostane nespremenjen. To pomeni, da moramo izračunati kontrolne točke $\mathbf{b}_i^{(1)}$, tako da bo veljala enakost

$$\sum_{|\mathbf{i}|=n} \mathbf{b}_i B_i^n(\mathbf{u}) = \sum_{|\mathbf{i}|=n+1} \mathbf{b}_i^{(1)} B_i^{n+1}(\mathbf{u}). \quad (5.6)$$

Preoblikovali bomo levo stran enakosti (5.6) tako, da bomo dobili željene koeficiente pri elementu $B_i^{n+1}(\mathbf{u})$. Ker velja $u + v + w = 1$, lahko zapišemo:

$$\begin{aligned} \sum_{|\mathbf{i}|=n} \mathbf{b}_i B_i^n(\mathbf{u}) &= \sum_{|\mathbf{i}|=n} \mathbf{b}_i B_i^n(\mathbf{u})(u + v + w) \\ &= \sum_{|\mathbf{i}|=n} \mathbf{b}_i (uB_i^n(\mathbf{u}) + vB_i^n(\mathbf{u}) + wB_i^n(\mathbf{u})). \end{aligned} \quad (5.7)$$

Poglejmo si sedaj element $B_{i+\mathbf{e}_1}^{n+1}(\mathbf{u})$ in ga skušajmo izraziti z elementom $B_i^n(\mathbf{u})$. Po definiciji je enak

$$B_{i+\mathbf{e}_1}^{n+1}(\mathbf{u}) = \frac{(n+1)!}{(i+1)!j!k!} u^{i+1}v^jw^k = u \frac{n+1}{i+1} B_i^n(\mathbf{u}).$$

Podobno dobimo

$$B_{i+\mathbf{e}_2}^{n+1}(\mathbf{u}) = v \frac{n+1}{j+1} B_i^n(\mathbf{u}), \quad B_{i+\mathbf{e}_3}^{n+1}(\mathbf{u}) = w \frac{n+1}{k+1} B_i^n(\mathbf{u}).$$

Uporabimo te ugotovitve in jih vstavimo v (5.7). Dobimo

$$\begin{aligned} \sum_{|\mathbf{i}|=n} \mathbf{b}_i B_i^n(\mathbf{u}) &= \sum_{|\mathbf{i}|=n} \mathbf{b}_i \left(\frac{i+1}{n+1} B_{i+\mathbf{e}_1}^{n+1}(\mathbf{u}) + \frac{j+1}{n+1} B_{i+\mathbf{e}_2}^{n+1}(\mathbf{u}) + \frac{k+1}{n+1} B_{i+\mathbf{e}_3}^{n+1}(\mathbf{u}) \right) \\ &= \sum_{|\mathbf{i}|=n} \frac{1}{n+1} \mathbf{b}_i ((i+1)B_{i+\mathbf{e}_1}^{n+1}(\mathbf{u}) + (j+1)B_{i+\mathbf{e}_2}^{n+1}(\mathbf{u}) + (k+1)B_{i+\mathbf{e}_3}^{n+1}(\mathbf{u})). \end{aligned} \quad (5.8)$$

Če vzamemo v vsoti indekse \mathbf{i} , za katere velja $|\mathbf{i}| = n+1$, se vsi Bernsteinovi polinomi v (5.8) spremenijo v $B_i^{n+1}(\mathbf{u})$, ki jih izpostavimo, ter premaknemo spremenjene kontrolne točke v oklepaj. Torej:

$$\sum_{|\mathbf{i}|=n} \mathbf{b}_i B_i^n(\mathbf{u}) = \sum_{|\mathbf{i}|=n+1} \frac{1}{n+1} (i\mathbf{b}_{i-\mathbf{e}_1} + j\mathbf{b}_{i-\mathbf{e}_2} + k\mathbf{b}_{i-\mathbf{e}_3}) B_i^{n+1}(\mathbf{u}).$$

□

Zgled 5.3

Izberimo Bézierov trikotnik stopnje $n = 2$ s sledečimi kontrolnimi točkami:

$$\begin{array}{ccccc} & & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & & \\ & & & & \\ & & & & \\ & & \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} & & \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix} \\ & & & & \\ & & & & \\ \begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix} & & \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix} & & \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \end{array}.$$

Zvišajmo stopnjo na $n = 3$ z uporabo (5.5) in dobimo naslednje kontrolne točke $\mathbf{b}_i^{(1)}$:

$$\begin{array}{ccccccc} & & & & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & & \\ & & & & & & \\ & & & & & & \\ & & & & \begin{bmatrix} 5/3 \\ 1/3 \\ 1/3 \end{bmatrix} & & \begin{bmatrix} 5/3 \\ 5/3 \\ 1/3 \end{bmatrix} \\ & & & & & & \\ & & & & & & \\ & & & & \begin{bmatrix} 7/3 \\ 2/3 \\ 1/3 \end{bmatrix} & & \begin{bmatrix} 7/3 \\ 2 \\ 1/3 \end{bmatrix} \\ & & & & & & \\ \begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix} & & \begin{bmatrix} 7/3 \\ -1/3 \\ 1/3 \end{bmatrix} & & \begin{bmatrix} 3 \\ -1/3 \\ 1 \end{bmatrix} & & \begin{bmatrix} 3 \\ 2/3 \\ 1 \end{bmatrix} & & \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \end{array}.$$

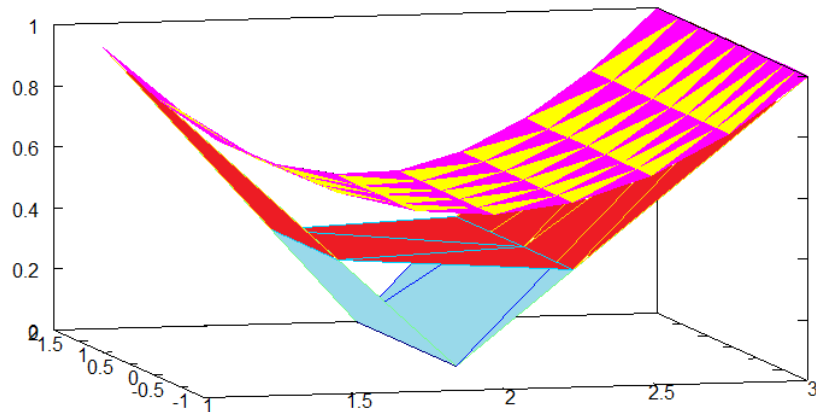
Na sliki 5.3 je prikazan Bézierov trikotnik s prvotno kontrolno mrežo in z novo kontrolno mrežo po zvišanju stopnje. Bézierov trikotnik se po zvišanju stopnje ne spremeni.

Višanje stopnje lahko večkrat ponovimo, kar smo že videli pri krivuljah. Kontrolne mreže konvergirajo v tem primeru proti Bézierovemu trikotniku. Več informacij o tem najdemo v [8].

5.5 Odvodi

Bézierove trikotnike lahko zlepimo skupaj za kar potrebujemo pogoje zveznosti, ki jih bomo spoznali v naslednjem podpoglavju. Pri tem igrajo pomembno vlogo odvodi. Pri tenzorskem produktu smo si pomagali s parcialnimi odvodi, ker smo imeli dve smeri odvajanja u in v . Pri Bézierovih trikotnikih pa je potrebno uporabiti t.i. *smerne odvode*. V tem podpoglavju bomo predstavili, kako so smerni odvodi definirani, kako jih izračunamo, podali pa bomo tudi zgled njihove uporabe. Teoretično ozadje odvodov Bézierovih trikotnikov je dostopno v [5].

Definicija 25. Naj bosta \mathbf{u}_1 in \mathbf{u}_2 točki iz trikotne domene. Potem je njuna razlika po definiciji vektor, ki ga označimo z $\mathbf{d} = (d, e, f)$. **Smerni odvod** ploskve $\mathbf{x}(\mathbf{u})$ glede na



Slika 5.3: Bézierov trikotnik s prvotno kontrolno mrežo (modra barva) in s kontrolno mrežo po višanju stopnje (rdeča barva).

vektor \mathbf{d} je definiran kot

$$D_{\mathbf{d}}\mathbf{x}(\mathbf{u}) = \lim_{\mathbf{d} \rightarrow 0} \frac{1}{|\mathbf{d}|} (\mathbf{x}(\mathbf{u} + \mathbf{d}) - \mathbf{x}(\mathbf{u})).$$

Opomba: Ko smo govorili o točkah, ki jih izrazimo kot baricentrične kombinacije drugih točk, je veljalo $u + v + w = 1$. Pri vektorjih oblike $\mathbf{d} = (d, e, f)$ pa velja $d + e + f = 0$.

Poglejmo si dva ekvivalentna načina izračuna smernega odvoda:

1. način

Naj bo \mathbf{b}^n Bézierov trikotnik stopnje n . Naj velja $|\mathbf{i}| = n - 1$ in naj bodo točke $\mathbf{b}_i^1(\mathbf{d})$ definirane kot

$$\mathbf{b}_i^1(\mathbf{d}) = d \mathbf{b}_{\mathbf{i}+\mathbf{e}_1} + e \mathbf{b}_{\mathbf{i}+\mathbf{e}_2} + f \mathbf{b}_{\mathbf{i}+\mathbf{e}_3}.$$

Potem smerni odvod Bézierovega trikotnika glede na vektor $\mathbf{d} = (d, e, f)$ izračunamo kot

$$D_{\mathbf{d}}\mathbf{b}(\mathbf{u}) = n \cdot \sum_{|\mathbf{i}|=n-1} \mathbf{b}_i^1(\mathbf{d}) B_i^{n-1}(\mathbf{u}), \quad (5.9)$$

kjer velja $d + e + f = 0$ in $u + v + w = 1$.

2. način

Naj bo \mathbf{b}^n Bézierov trikotnik stopnje n . Naj bodo točke $\mathbf{b}_i^r(\mathbf{u})$ definirane kot

$$\mathbf{b}_i^r(\mathbf{u}) = u \mathbf{b}_{\mathbf{i}+\mathbf{e}_1}^{r-1}(\mathbf{u}) + v \mathbf{b}_{\mathbf{i}+\mathbf{e}_2}^{r-1}(\mathbf{u}) + w \mathbf{b}_{\mathbf{i}+\mathbf{e}_3}^{r-1}(\mathbf{u}),$$

kjer velja $\mathbf{b}_i^0(\mathbf{u}) := \mathbf{b}_i$. Potem smerni odvod Bézierovega trikotnika glede na vektor $\mathbf{d} = (d, e, f)$ izračunamo kot

$$D_{\mathbf{d}}\mathbf{b}(\mathbf{u}) = n \cdot (d \mathbf{b}_{\mathbf{e}_1}^{n-1}(\mathbf{u}) + e \mathbf{b}_{\mathbf{e}_2}^{n-1}(\mathbf{u}) + f \mathbf{b}_{\mathbf{e}_3}^{n-1}(\mathbf{u})),$$

kjer je $d + e + f = 0$ in $u + v + w = 1$.

Zgled 5.4

Naj bo $n = 3$. Vzemimo naslednjo kontrolno mrežo Bézierovega trikotnika:

$$\begin{array}{ccccccc}
 & & & & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & & \\
 & & & & & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & \\
 & & & \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} & & & \\
 & & & & \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} & & \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} \\
 & & \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} & & & & \\
 & & & \begin{bmatrix} 3 \\ -1 \\ 0 \end{bmatrix} & & \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 3 \\ 2 \\ 0 \end{bmatrix} \\
 & \begin{bmatrix} 3 \\ -2 \\ 0 \end{bmatrix} & & & & &
 \end{array}$$

in izberimo $\mathbf{u} = (\frac{1}{2}, \frac{1}{3}, \frac{1}{6})$ ter vektor $\mathbf{d} = (1, -1, 0)$.

Izračunajmo smerni odvod najprej s prvim načinom. Elementi v vsoti v (5.9) so enaki

$$\begin{aligned}
 B_{(0,2,0)}^2(\mathbf{u}) \cdot \mathbf{b}_{(0,2,0)}^1(\mathbf{d}) &= \frac{1}{9} (\mathbf{b}_{(1,2,0)} - \mathbf{b}_{(0,3,0)}) = \begin{bmatrix} 1/9 \\ 1/9 \\ 1/9 \end{bmatrix}, \\
 B_{(0,1,1)}^2(\mathbf{u}) \cdot \mathbf{b}_{(0,1,1)}^1(\mathbf{d}) &= \frac{1}{9} (\mathbf{b}_{(1,1,1)} - \mathbf{b}_{(0,2,1)}) = \begin{bmatrix} 1/9 \\ 1/9 \\ 0 \end{bmatrix}, \\
 B_{(1,1,0)}^2(\mathbf{u}) \cdot \mathbf{b}_{(1,1,0)}^1(\mathbf{d}) &= \frac{1}{3} (\mathbf{b}_{(2,1,0)} - \mathbf{b}_{(1,2,0)}) = \begin{bmatrix} 1/3 \\ 0 \\ 0 \end{bmatrix}, \\
 B_{(0,0,2)}^2(\mathbf{u}) \cdot \mathbf{b}_{(0,0,2)}^1(\mathbf{d}) &= \frac{1}{36} (\mathbf{b}_{(1,0,2)} - \mathbf{b}_{(0,1,2)}) = \begin{bmatrix} 1/36 \\ 0 \\ -1/36 \end{bmatrix}, \\
 B_{(1,0,1)}^2(\mathbf{u}) \cdot \mathbf{b}_{(1,0,1)}^1(\mathbf{d}) &= \frac{1}{6} (\mathbf{b}_{(2,0,1)} - \mathbf{b}_{(1,1,1)}) = \begin{bmatrix} 1/6 \\ 0 \\ -1/6 \end{bmatrix}, \\
 B_{(2,0,0)}^2(\mathbf{u}) \cdot \mathbf{b}_{(2,0,0)}^1(\mathbf{d}) &= \frac{1}{4} (\mathbf{b}_{(3,0,0)} - \mathbf{b}_{(2,1,0)}) = \begin{bmatrix} 1/4 \\ 1/4 \\ -1/4 \end{bmatrix}.
 \end{aligned}$$

Torej velja

$$D_{\mathbf{d}}\mathbf{b}(\mathbf{u}) = 3 \cdot \sum_{|i|=2} \mathbf{b}_i^1(\mathbf{d}) B_i^2(\mathbf{u}) = 3 \cdot \begin{bmatrix} 1 \\ 17/36 \\ -1/3 \end{bmatrix} = \begin{bmatrix} 3 \\ 17/12 \\ -1 \end{bmatrix}.$$

Izračunajmo sedaj smerni odvod še z drugim načinom, s katerim pridemo do istega rezultata. Smerni odvod računamo v tem primeru kot

$$D_{\mathbf{d}}\mathbf{b}(\mathbf{u}) = 3 \cdot (\mathbf{b}_{(1,0,0)}^2(\mathbf{u}) - \mathbf{b}_{(0,1,0)}^2(\mathbf{u})).$$

Izračunajmo elemente

$$\begin{aligned}\mathbf{b}_{(2,0,0)}^1(\mathbf{u}) &= \frac{1}{2}\mathbf{b}_{(3,0,0)} + \frac{1}{3}\mathbf{b}_{(2,1,0)} + \frac{1}{6}\mathbf{b}_{(2,0,1)} = \begin{bmatrix} 8/3 \\ 4/3 \\ 1/3 \end{bmatrix}, \\ \mathbf{b}_{(1,1,0)}^1(\mathbf{u}) &= \frac{1}{2}\mathbf{b}_{(2,1,0)} + \frac{1}{3}\mathbf{b}_{(1,2,0)} + \frac{1}{6}\mathbf{b}_{(1,1,1)} = \begin{bmatrix} 5/3 \\ 5/6 \\ 1 \end{bmatrix}, \\ \mathbf{b}_{(1,0,1)}^1(\mathbf{u}) &= \frac{1}{2}\mathbf{b}_{(2,0,1)} + \frac{1}{3}\mathbf{b}_{(1,1,1)} + \frac{1}{6}\mathbf{b}_{(1,0,2)} = \begin{bmatrix} 8/3 \\ -1/6 \\ 1/3 \end{bmatrix}.\end{aligned}$$

Torej dobimo

$$\mathbf{b}_{(1,0,0)}^2(\mathbf{u}) = \frac{1}{2} \begin{bmatrix} 8/3 \\ 4/3 \\ 1/3 \end{bmatrix} + \frac{1}{3} \begin{bmatrix} 5/3 \\ 5/6 \\ 1 \end{bmatrix} + \frac{1}{6} \begin{bmatrix} 8/3 \\ -1/6 \\ 1/3 \end{bmatrix} = \begin{bmatrix} 7/3 \\ 11/12 \\ 5/9 \end{bmatrix}.$$

Podobno bi izračunali še

$$\mathbf{b}_{(0,1,0)}^2(\mathbf{u}) = \begin{bmatrix} 4/3 \\ 4/9 \\ 8/9 \end{bmatrix}.$$

Končno lahko zaključimo:

$$D_{\mathbf{d}}\mathbf{b}(\mathbf{u}) = 3 \cdot \left(\begin{bmatrix} 7/3 \\ 11/12 \\ 5/9 \end{bmatrix} - \begin{bmatrix} 4/3 \\ 4/9 \\ 8/9 \end{bmatrix} \right) = \begin{bmatrix} 3 \\ 17/12 \\ -1 \end{bmatrix}.$$

5.6 C^r zveznost

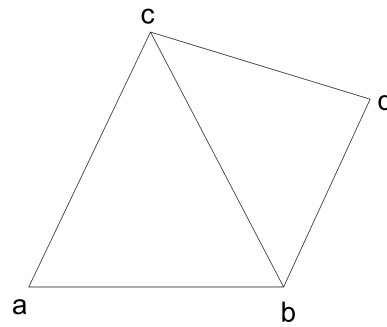
Podobno kot pri tenzorskem produktu bomo v tem podpoglavju opisali samo C^r zveznost in podrobneje pogledali C^0 in C^1 zveznost. Za več informacij glede geometrijske zveznosti C^r za Bézierove trikotnike glej [5].

Čeprav bomo tukaj obravnavali samo zlepke dveh Bézierovih trikotnikov, lahko analizo razširimo tudi na zlepke več Bézierovih trikotnikov ([5]). Predpostavimo, da imamo domeno trikotne oblike z oglišči \mathbf{a} , \mathbf{b} , \mathbf{c} . Dodajmo še eno domeno trikotne oblike, ki bo z začetno domeno delila eno stranico. Novo točko imenujmo \mathbf{d} , za katero vemo, da velja

$$\mathbf{d} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}, \quad \alpha + \beta + \gamma = 1. \quad (5.10)$$

Prikaz domene, ki je sestavljena iz dveh trikotnih domen, ki si delita stranico \mathbf{bc} , je na sliki 5.4.

Definicija 26. *Daljica, ki leži na sestavljeni domeni in prečka rob, ki je skupen obema domenama, se preslika na ploskev kot **sestavljena krivulja**. Del te krivulje leži na ploskvi, ki je slika trikotne domene \mathbf{abc} , drugi del krivulje pa na ploskvi, ki je slika domene \mathbf{bcd} .*



Slika 5.4: Prikaz sestavljene domene.

Z uporabo sestavljenih krivulj definirajmo \mathcal{C}^r zveznost.

Definicija 27. Naj imata dva Bézierova trikotnika skupno robno krivuljo in naj bodo vse sestavljene krivulje na zlepku \mathcal{C}^r zvezne krivulje. Potem je zlepek dveh Bézierovih trikotnikov tudi \mathcal{C}^r zvezen.

Sedaj pa pogledajmo, kako se pogoji \mathcal{C}^r zveznosti odražajo na kontrolnih točkah. Ko bomo govorili o \mathcal{C}^r zveznosti zlepkov dveh Bézierovih trikotnikov, bo pomembno upoštevati, katero robno krivuljo imata oba Bézierova trikotnika skupno. Za nadaljno obravnavo zveznosti predpostavimo, da ju zlepimo pri $w = 0$.

Trditev 8. Izberimo kot domeno zleпка dveh Bézierovih trikotnikov domeno na sliki 4.4. Naj bo $\mathbf{b}(\mathbf{u})$ Bézierov trikotnik definiran nad domeno \mathbf{abc} in $\mathbf{c}(\mathbf{u})$ Bézierov trikotnik definiran nad domeno \mathbf{bcd} . Potreben in zadosten pogoj za \mathcal{C}^r zveznost zleпка je

$$\mathbf{c}_{(i,j,\rho)} = \mathbf{b}_{(i,j,0)}^\rho(\mathbf{d}), \quad i + j + \rho = n, \quad \rho = 0, 1, \dots, r. \quad (5.11)$$

Dokaz trditve izpustimo, je pa dostopen v [5]. Paziti moramo, ker se enačba (5.11) spremeni, če od vsega začetka pri sestavljeni domeni predpostavimo, da je $u = 0$ ali $v = 0$. V vsakem primeru bi razmišljanje potekalo na podoben način.

Izraz (5.11) nam pove, kako moramo postaviti kontrolne točke Bézierovega trikotnika \mathbf{c} , da bo zlepek z Bézierovim trikotnikom \mathbf{b} \mathcal{C}^r zvezen. Pogledajmo si podrobno izraz (5.11) za $r = 0$ in $r = 1$ ter pri tem upoštevajmo (5.10) in na kratko pišimo $\mathbf{d} = (\alpha, \beta, \gamma)$.

- Zlepek dveh Bézierovih trikotnikov je \mathcal{C}^0 zvezen, če velja

$$\mathbf{c}_{(i,j,0)} = \mathbf{b}_{(i,j,0)}^0(\mathbf{d}) = \mathbf{b}_{(i,j,0)}^0, \quad i + j = n. \quad (5.12)$$

Geometrijsko to pomeni, da morata Bézierova trikotnika imeti skupni robni kontrolni poligon pri $w = 0$.

- Zlepek dveh Bézierovih trikotnikov je \mathcal{C}^1 zvezen, če poleg (5.12) velja še

$$\begin{aligned} \mathbf{c}_{(i,j,1)} &= \mathbf{b}_{(i,j,0)}^1(\mathbf{d}) \\ &= \alpha \mathbf{b}_{(i+1,j,0)} + \beta \mathbf{b}_{(i,j+1,0)} + \gamma \mathbf{b}_{(i,j,1)}, \quad i + j = n - 1. \end{aligned} \quad (5.13)$$

Vidimo, da točke $\mathbf{c}_{(i,j,1)}$ dobimo kot baricentrično kombinacijo točk nekega podtrikotnika na kontrolni mreži. Pogoj (5.13) geometrijsko pomeni, da je vsak par trikotnikov

ob skupni robni krivulji zleпка koplanaren in je afina preslikava para trikotnikov v sestavljeni domeni, kar pomeni, da morata ploščini vsakega para trikotnikov ohranjati razmerje, v katerem sta ploščini trikotnikov v domeni zleпка. Koplanarnost je torej potreben pogoj za \mathcal{C}^1 zveznost, ne pa zadostni.

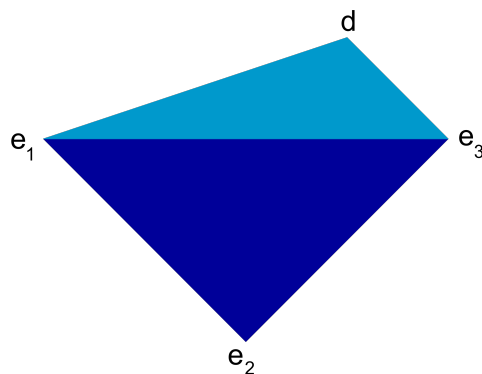
Podajmo zgled za \mathcal{C}^1 zveznost dveh Bézierovih trikotnikov.

Zgled 5.5

Vzemimo Bézierov trikotnik podan z naslednjo kontrolno mrežo:

$$\begin{array}{ccccccc} & & & & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & & \\ & & & & & & \\ & & & & \begin{bmatrix} 2 \\ 0 \\ 1/2 \end{bmatrix} & & \begin{bmatrix} 2 \\ 2 \\ 1/2 \end{bmatrix} \\ & & & & & & \\ & & & & \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix} & & \begin{bmatrix} 3 \\ 3 \\ 0 \end{bmatrix} \\ & & & & & & \\ \begin{bmatrix} 4 \\ -2 \\ 1 \end{bmatrix} & & \begin{bmatrix} 3 \\ -1 \\ 0 \end{bmatrix} & & \begin{bmatrix} 4 \\ -1 \\ 1 \end{bmatrix} & & \begin{bmatrix} 4 \\ 0 \\ 1 \end{bmatrix} & & \begin{bmatrix} 4 \\ 4 \\ 1 \end{bmatrix} \end{array} .$$

Njegova domena je običajne trikotne oblike. Naj bodo v baricentričnih koordinatah oglišča te domene točke $\mathbf{e}_1 = (1, 0, 0)$, $\mathbf{e}_2 = (0, 1, 0)$ in $\mathbf{e}_3 = (0, 0, 1)$. Dodajmo še drugo domeno, ki bo imela s prvo skupno stranico pri $v = 0$. V ta namen definirajmo novo točko $\mathbf{d} = (\frac{1}{2}, -\frac{1}{2}, 1)$. Oglišča druge domene bodo torej $\mathbf{e}_1, \mathbf{e}_3$ in \mathbf{d} . Dobili smo domeno za zlepek dveh Bézierovih trikotnikov, ki je prikazana na sliki 5.5.



Slika 5.5: Domena za zlepek dveh Bézierovih trikotnikov.

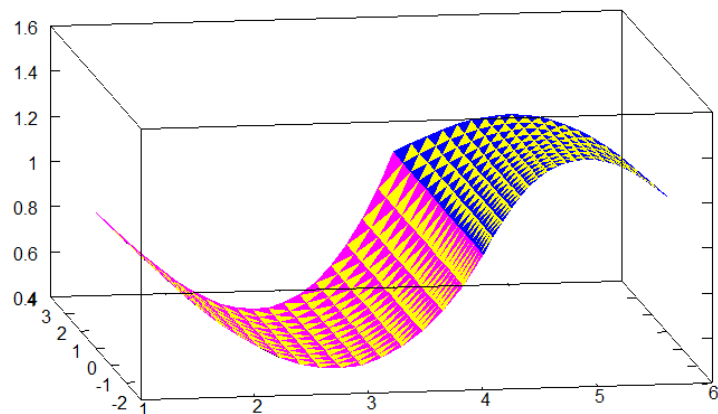
S pomočjo (5.13) izračunajmo točke $\mathbf{c}_{(i,1,k)}$ in dobimo

$$\begin{array}{ccccc} & & \begin{bmatrix} 5/2 \\ 1/2 \\ 1/4 \end{bmatrix} & & \\ & & & & \\ & & & & \\ & & \begin{bmatrix} 7/2 \\ -1/2 \\ -1/4 \end{bmatrix} & & \begin{bmatrix} 7/2 \\ 3/2 \\ -1/4 \end{bmatrix} & & \\ & & & & & & \\ \begin{bmatrix} 9/2 \\ -2 \\ 3/2 \end{bmatrix} & & \begin{bmatrix} 9/2 \\ -3/2 \\ 3/2 \end{bmatrix} & & \begin{bmatrix} 9/2 \\ 1/2 \\ 3/2 \end{bmatrix} & & \end{array}.$$

Ker smo domeno zlepk zlepi pri $v = 0$, bo naš začetni Bézierov trikotnik delil z drugim robno krivuljo pri $v = 0$. Kontrolna mreža drugega Bézierovega trikotnika, ki bo z začetnim tvoril \mathcal{C}^1 zvezen zlepek, bo potem enaka

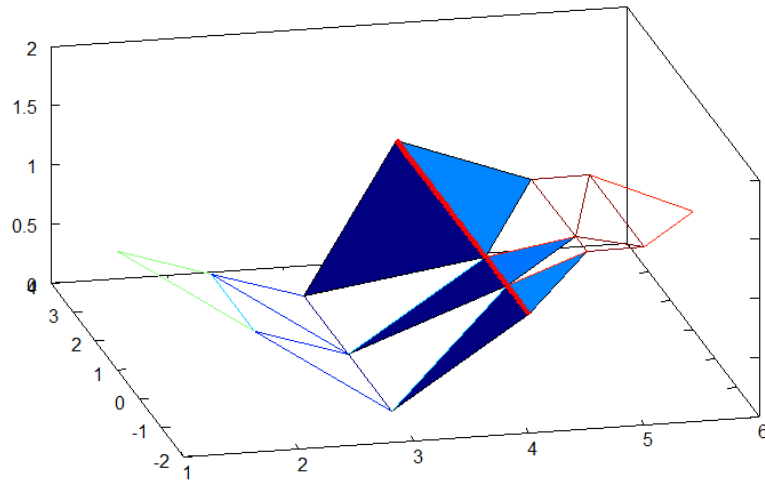
$$\begin{array}{ccccccc} & & & & \begin{bmatrix} 6 \\ 1 \\ 1 \end{bmatrix} & & \\ & & & & & & \\ & & & & & & \\ & & & & \begin{bmatrix} 5 \\ -2 \\ 3/2 \end{bmatrix} & & \begin{bmatrix} 5 \\ 1/2 \\ 3/2 \end{bmatrix} & & \\ & & & & & & & & \\ \begin{bmatrix} 9/2 \\ -2 \\ 3/2 \end{bmatrix} & & \begin{bmatrix} 9/2 \\ -3/2 \\ 3/2 \end{bmatrix} & & \begin{bmatrix} 9/2 \\ 1/2 \\ 3/2 \end{bmatrix} & & & & \\ \begin{bmatrix} 4 \\ -2 \\ 1 \end{bmatrix} & & \begin{bmatrix} 4 \\ -1 \\ 1 \end{bmatrix} & & \begin{bmatrix} 4 \\ 0 \\ 1 \end{bmatrix} & & \begin{bmatrix} 4 \\ 4 \\ 1 \end{bmatrix} & & \end{array}.$$

Točke v rdečem smo izbrali poljubno, saj ne vplivajo na \mathcal{C}^1 zveznost zlepk. Na sliki 5.6 je prikazan zlepek dveh Bézierovih trikotnikov iz zglada.



Slika 5.6: Zlepek dveh Bézierovih trikotnikov, ki je \mathcal{C}^1 zvezen.

Na sliki 5.7 sta prikazani kontrolni mreži obeh ploskev, kjer rdeča črta predstavlja skupni kontrolni poligon, ki ga imata obe ploskvi. Vsi trije pari trikotnikov so koplanarni. Poleg tega sta ploščini vsakega para trikotnikov v razmerju 2 : 1, kar velja tudi za ploščini trikotnikov v domeni na sliki 5.5.



Slika 5.7: Kontrolni mreži obeh Bézierovih trikotnikov.

Poglavje 6

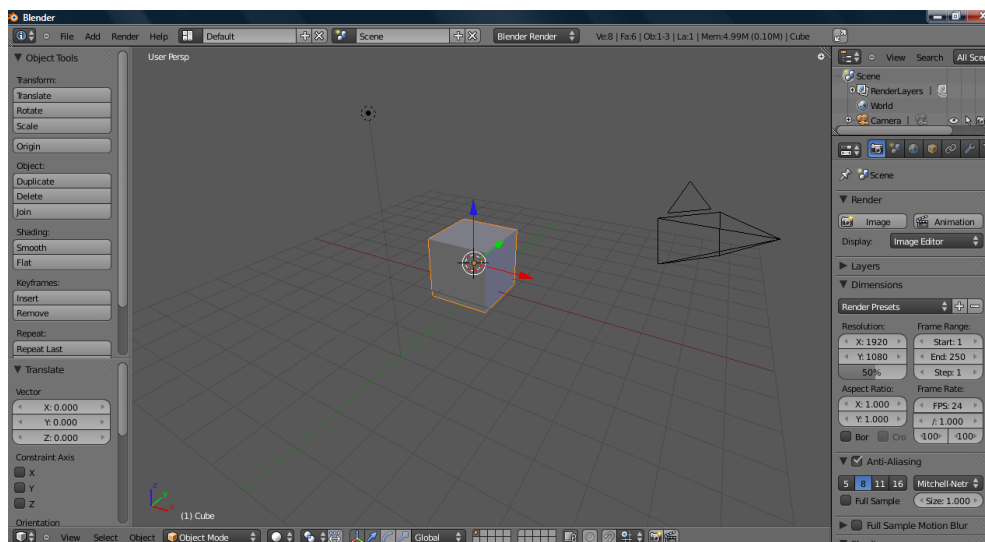
Računalniške aplikacije

Po teoretični utemeljitvi pomembnosti Bézierovih krivulj in ploskev pri računalniškem oblikovanju objektov v prejšnjih poglavjih, se bomo v tem poglavju osredotočili na oblikovanje avtomobila in spoznali praktično uporabo opisanih krivulj in ploskev. Najprej bomo podrobno predstavili računalniški program Blender, ki smo ga že uporabili v podpoglavjih 3.9 in 4.11. Nato se bomo s pomočjo orodij, ki jih bomo spoznali v naslednjem podpoglavju, posvetili modeliranju avtomobila.

6.1 Računalniški program Blender

6.1.1 Grafični uporabniški vmesnik programa

Ko odpremo program Blender, se nam pojavi okno, ki je prikazano na sliki 6.1. Na začetku imamo na sceni tri objekte: kocko, kamero in luč. Nekar objekte, ki jih modeliramo,

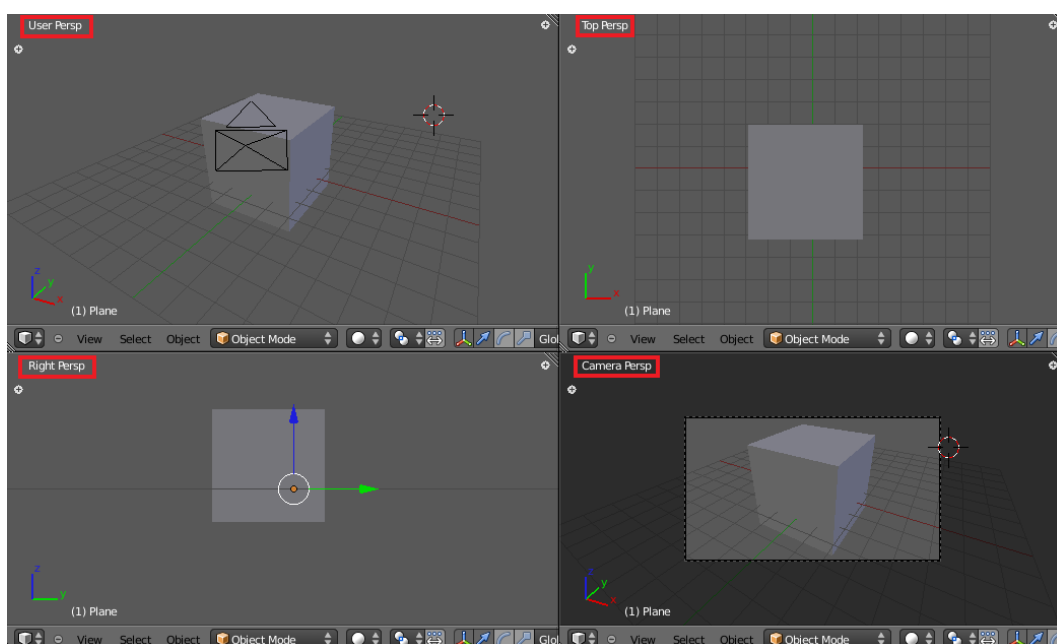


Slika 6.1: Začetna scena Blenderja.

postavimo na sceno in z izbiro ustreznega položaja kamere ustvarimo realnejši prikaz. Z uporabo ukaza *Render* namreč dosežemo, da kamera deluje kot fotografski aparat. Pri tem luč osvetli sceno in ustvari sence, saj jo lahko premikamo kamor želimo in tako ustvarimo različne svetlobne efekte. Ko s klikom miške izberemo poljubni objekt, se robovi objekta

obarvajo z oranžno barvo (kocka na sliki 6.1) in se na objektu prikažejo tri puščice (zeleno, modre in rdeče barve), ki predstavljajo koordinatni sistem x, y, z prostora. Puščice so prikazane tudi v levem spodnjem kotu scene.

V Blenderju imamo možnost, da glavno okno razdelimo na več manjših podoken, kot je razvidno na sliki 6.2. To je lahko zelo uporabna lastnost, predvsem ko imamo opravka z modeliranjem bolj kompleksnih objektov, kar bomo videli v naslednjem podpoglavju. V zgornjem levem oknu je prikazana kocka v perspektivi *User*; to je osnovni zorni kot, ki ga imamo že od samega začetka v glavnem oknu. V desnem zgornjem podoknu smo izbrali perspektivo *Top*, t.j. kocko gledamo z vrha osi z . V spodnjem levem podoknu gledamo kocko s stranskega zornega kota (perspektiva *Right*), v zadnjem podoknu pa je prikazano, kako kamera na sceni vidi kocko.

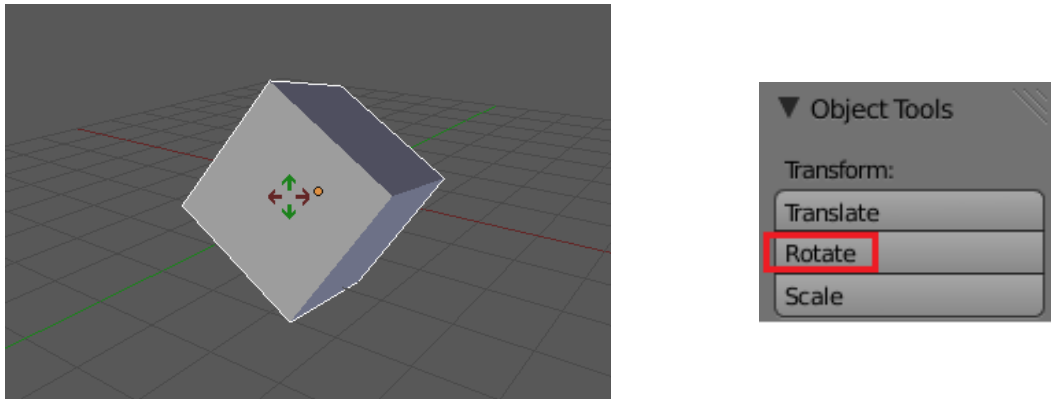


Slika 6.2: Glavno okno razdelimo na štiri podokna.

6.1.2 Glavni ukazi in lastnosti

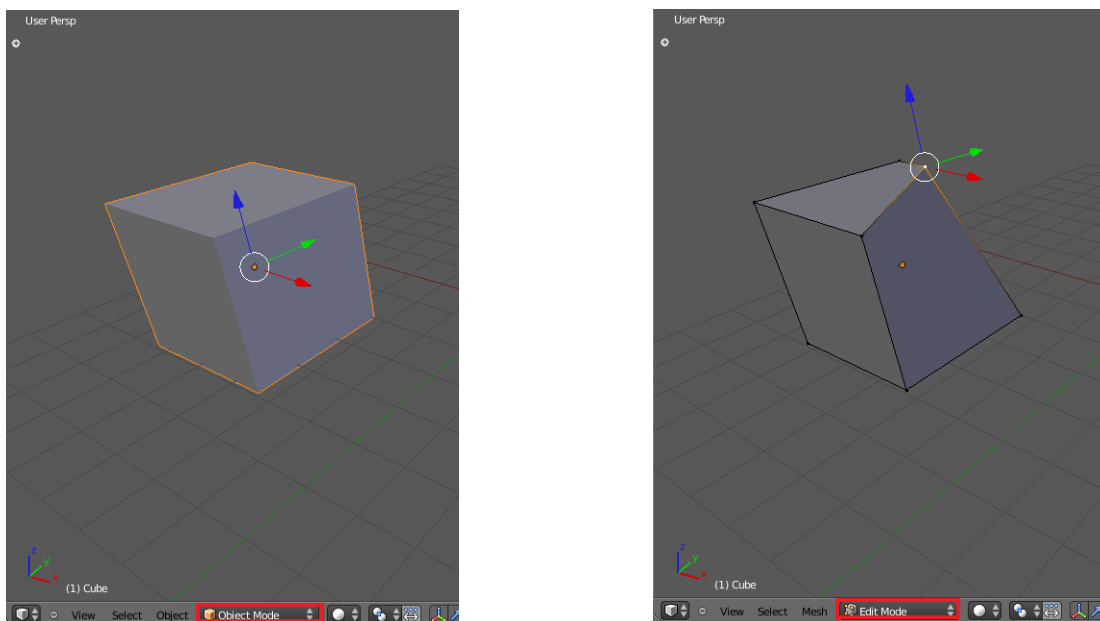
Na sliki 6.1 lahko vidimo, da scena vključuje raznorazne ukaze in možnosti. Opišimo le najpomembnejše in predvsem tiste, ki jih bomo potrebovali v nadaljevanju.

Za nas bodo zelo uporabni ukazi *Rotate*, *Translate* in *Scale*, ki objekt zarotirajo, translirajo vzdolž poljubne osi ali skalirajo. Ti ukazi se nahajajo pod skupnim imenom *Object Tools*, kjer so naštet še številna druga glavna orodja. Najdemo jih levo zgoraj v Blenderju. Na sliki 6.3 si lahko ogledamo primer, ko označimo kocko in apliciramo ukaz *Rotate*.



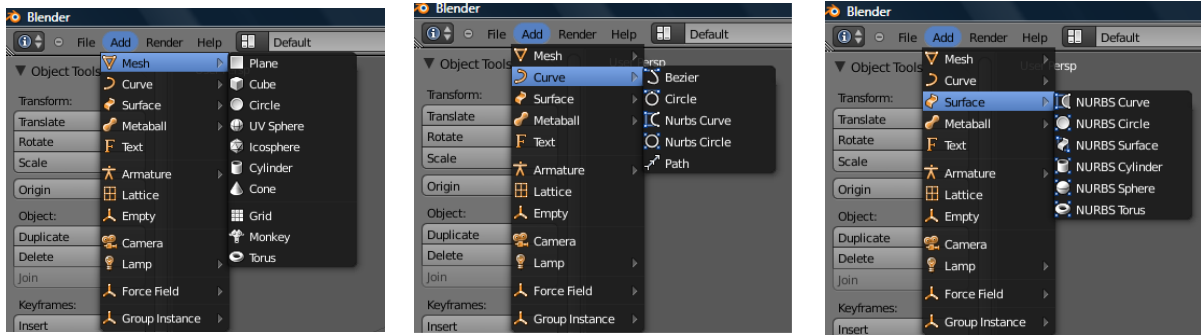
Slika 6.3: Primer uporabe ukaza *Rotate* za rotacijo objekta.

Pomembna lastnost Blenderja je, da lahko objekte na sceni vidimo in spreminjamo v dveh pogledih: *Object Mode* in *Edit Mode*. Vsak objekt in telo je na sceni sestavljeno iz točk, robov in ploskev. V pogledu *Object Mode* nimamo opravka s točkami, ampak predmet obravnavamo kot enotno telo, kar lahko vidimo na sliki 6.4 (levo). V pogledu *Edit Mode* pa se na objektu prikažejo točke, ki jih lahko odstranimo ali spreminjamo njihov položaj. Na sliki 6.4 (desno) smo prešli v pogled *Edit Mode*, označili točko in jo vzporedno premaknili vzdolž x osi (rdeča puščica).



Slika 6.4: V pogledu *Edit Mode* lahko objektu spremenimo strukturo.

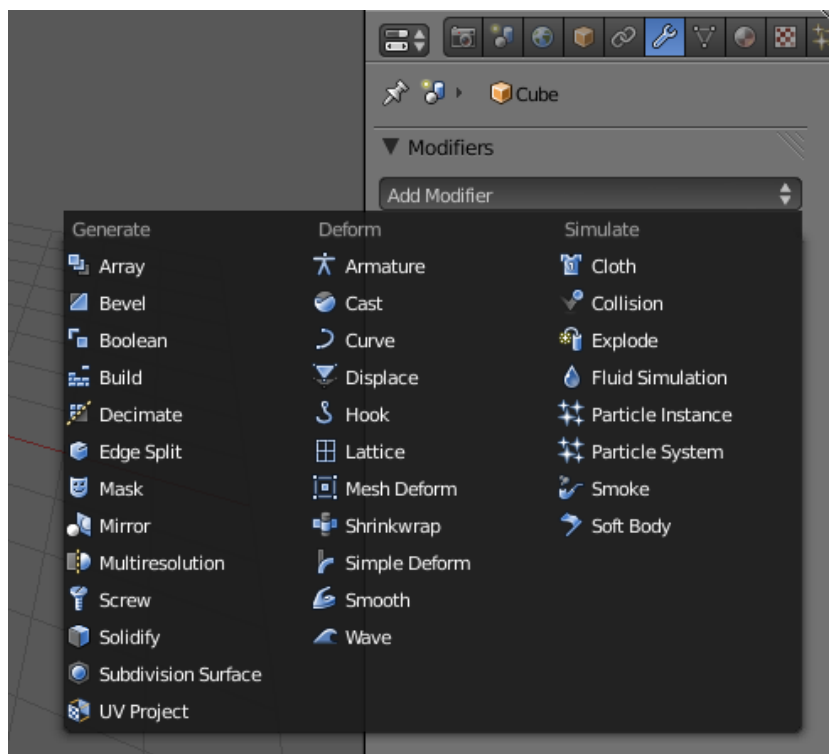
Iz zadnjega primera je razvidno, da lahko ukaze *Rotate*, *Scale* in *Translate* apliciramo tudi na točke, robove in ploskve objekta, pri čemer se moramo seveda nahajati v pogledu *Edit Mode*.



Slika 6.5: Krivulje, ploskve in druge objekte dodajamo na sceno z gumbom *Add*.

Z gumbom *Add*, ki se nahaja v zgornji orodni vrstici programa, lahko na sceno dodajamo raznorazne objekte. Na sliki 6.5 (levo) je prikazano, kako v Blenderju dodajamo objekte tipa *Mesh*, to so navadna geometrijska telesa, ki so sestavljena iz točk, robov in ploskev. Objekti tipa *Mesh* so: kocka, kroglja, valj, stožec, torus in drugi. Dodajamo pa lahko tudi krivulje (slika 6.5 - sredinska slika) in ploskve (slika 6.5 - desna slika). Krivulje in ploskve, ki smo jih uporabljali v podpoglavjih 3.9 in 4.11, so bile krivulje in ploskve tipa *NURBS*, ki kot podrazred vsebujejo racionalne Bézierove krivulje in ploskve. Posluževali se jih bomo tudi v tem poglavju pri modeliranju avtomobila.

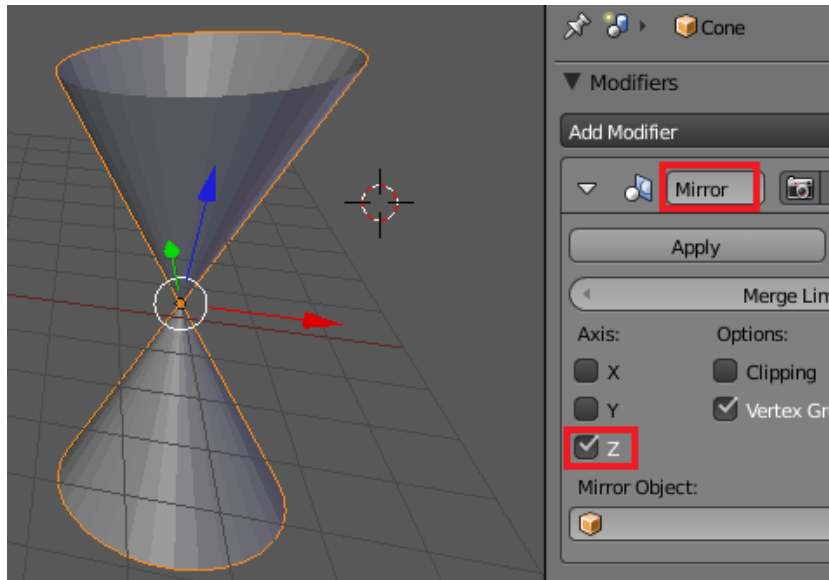
Naslednje pomembno orodje, ki se ga pogosto poslužujemo v Blenderju, je t.i. *Modifier*, ki omogoča spreminjanje že obstoječih objektov. Orodij tipa *Modifier* imamo več vrst (glej sliko 6.6), do njih pa pridemo z ukazom *Add Modifier*, ki se nahaja na desni strani ekrana.



Slika 6.6: Raznolikost orodja *Modifier*.

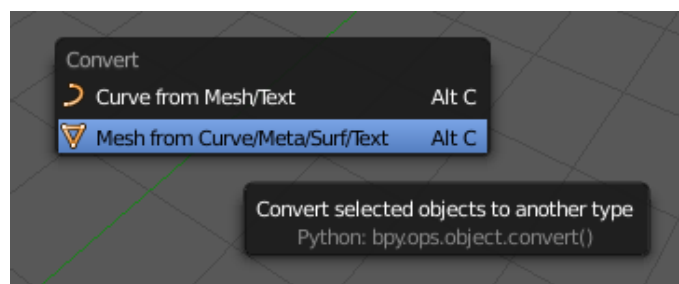
V naslednjem podpoglavju bomo pri modeliranju avtomobila spoznali in podrobno opisali

Modifier tipa *Mirror*, *Subdivision Surface*, *Array* in *Curve*. Na tem mestu podajmo samo primer uporabe *Modifier Mirror*, z namenom boljše predstavitve uporabnosti takega orodja. Predpostavimo, da smo na sceno dodali stožec in da ga želimo prezrcaliti vzdolž osi *z*. Kliknemo na stožec in z ukazom *Add Modifier Mirror* dosežemo željeni rezultat. Na sliki 6.7 vidimo, da moramo med ponujenimi možnostmi odključati pravilno os, vzdolž katere želimo zrcaliti.



Slika 6.7: Zrcaljenje objekta s pomočjo ukaza *Modifier Mirror*.

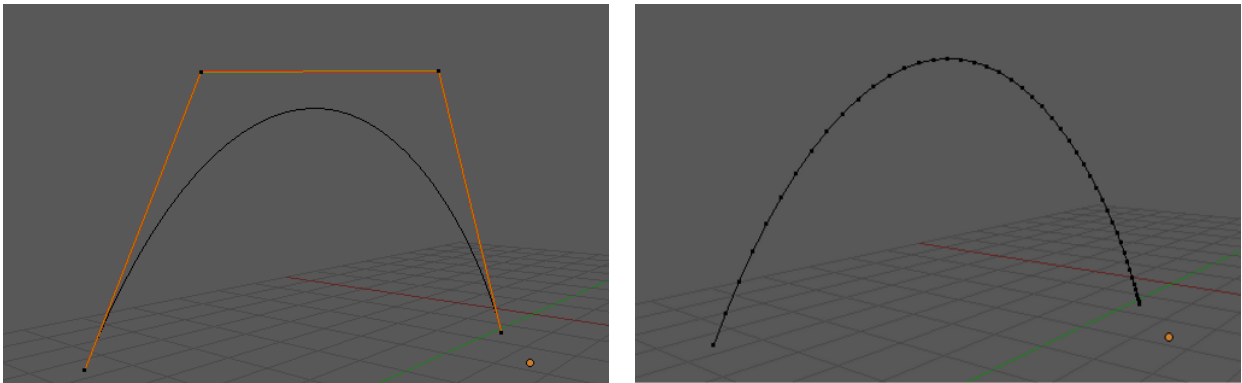
Pozorni moramo biti, da orodje *Modifier* uporabljamo samo na objektih, ki so tipa *Mesh*. Če smo predmet oblikovali z Bézierovimi krivuljami ali ploskvami, ga moramo torej najprej spremeniti v objekt tipa *Mesh* in šele nato dodati poljubni *Modifier*. Navedeno spremembo dosežemo z ukazom *Mesh from Curve/Meta/Surf/Text*, ki je prikazan na sliki 6.8.



Slika 6.8: Ukaz, s katerim spremenimo krivulje ali ploskve v objekte tipa *Mesh*.

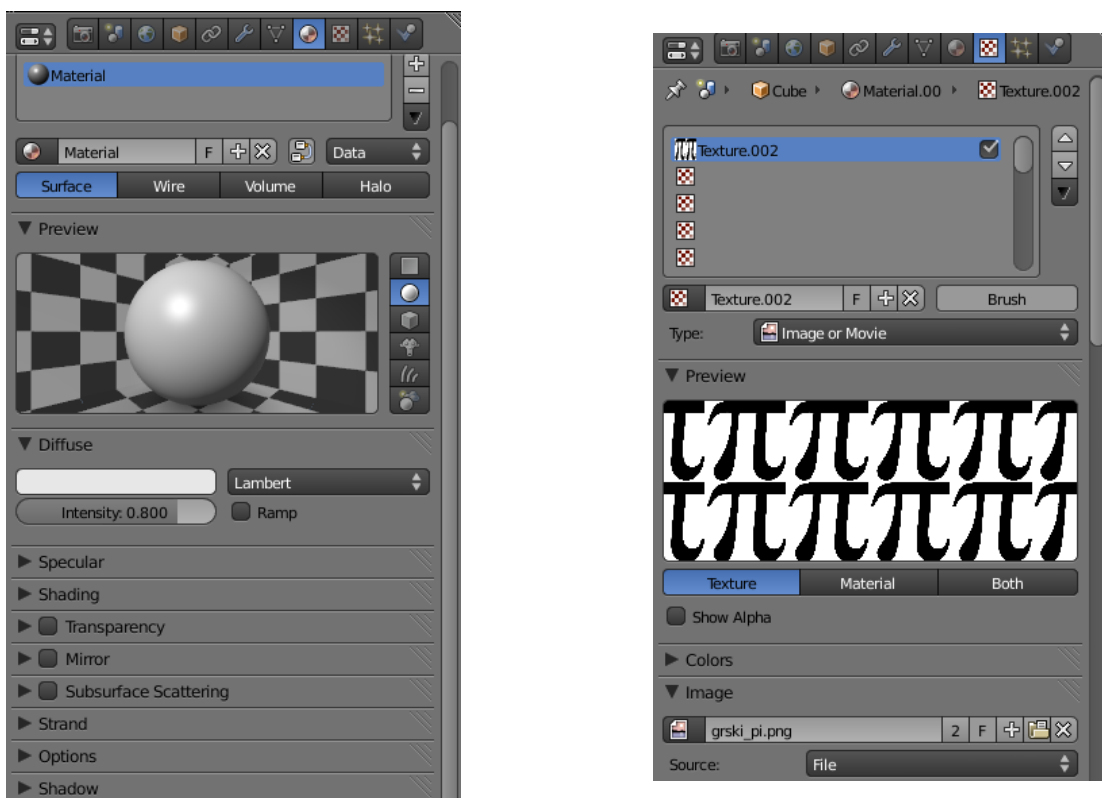
Če recimo dodamo v Blender Bézierovo krivuljo, bo ta določena s kontrolnim poligonom, kot smo jo spoznali v teoretičnem delu magistrskega dela. Če pa jo spremenimo v objekt tipa *Mesh*, potem krivulja ni več določena s kontrolnimi točkami, ampak s točkami, ki ležijo na sami krivulji. Opisana sprememba je prikazana na sliki 6.9.

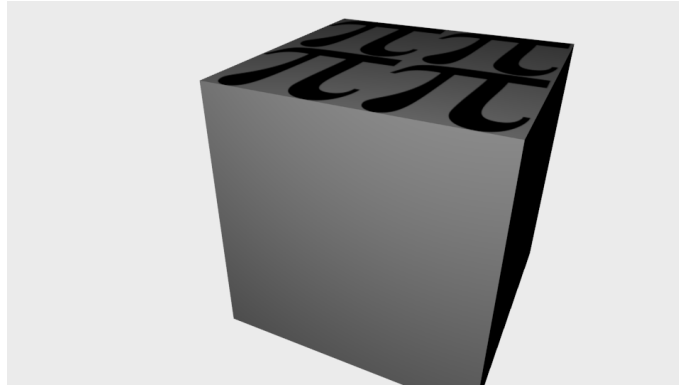
V naslednjem podpoglavju bomo uporabljali Bézierove ploskve za oblikovanje avtomobila, ki pa jih bomo morali v določenem trenutku spremeniti v objekte tipa *Mesh*, da bomo lahko nanje aplicirali orodje *Modifier*, ki nam prihrani veliko dela pri modeliranju kompleksnih objektov.

Slika 6.9: Bézierova krivulja pred in po spremembi v objekt tipa *Mesh*.

6.1.3 Barvanje in dodajanje efektov

Potem ko smo objekte na sceni oblikovali, jim lahko dodamo barvo, prozornost, sence ter določimo, ali se bo svetloba od danega objekta odbijala močneje, kakor od drugih objektov. Vse pravkar naštetosti lahko določimo v oknu *Material*. Ob oknu *Material* imamo še okno *Texture*, ki nam omogoča, da na dani objekt “prilepimo” izbrano sliko. Na sliki 6.11 je prikazan primer kocke, ki ji predpišemo lastnosti, ki so navedene na sliki 6.10: kocka bo sivo-bele barve, ne bo prozorna, ne bo odbijala veliko svetlobe, na njeno zgornjo ploskev pa prilepimo sliko, ki je prikazana v oknu *Texture* (grški pi). Kot vidimo, se lahko dana slika na objektu pojavi tudi večkrat. Potem ko objektu predpišemo vse željene lastnosti, postavimo kamero na objekt, uporabimo ukaz *Render*, s katerim kamera “slika” objekt, in dobimo sliko 6.11.

Slika 6.10: Okni *Materials* in *Texture*.



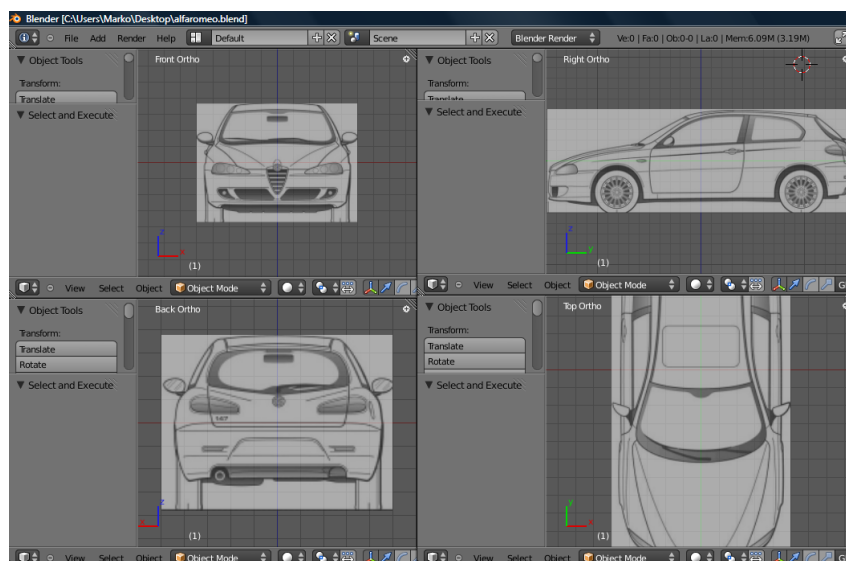
Slika 6.11: Primer kocke z danimi lastnostmi.

6.2 Modeliranje avtomobila

V tem podpoglavju bomo opisali le ukaze in metode, ki jih uporabljamo pri takem modeliranju, ker bi bil sicer podroben opis oblikovanja avtomobila z Blenderjem preobsežen. Osredotočili se bomo na opis modeliranja karoserije in pnevmatik. Konstrukcijo ostalih delov avtomobila ne bomo podrobno opisali, ampak samo navedli, katere metode bomo uporabljali. Modelirati avtomobil s prostim očesom je zahtevna naloga, zato si lahko pomagamo s slikami avtomobila, ki ga prikazujejo z vseh vidikov in perspektiv: od zgoraj, spodaj, spredaj in zadaj. Izbrali si bomo poljuben model avtomobila in poiskali ustrezne slike, kjer bo natančno narisan.

6.2.1 Karoserija

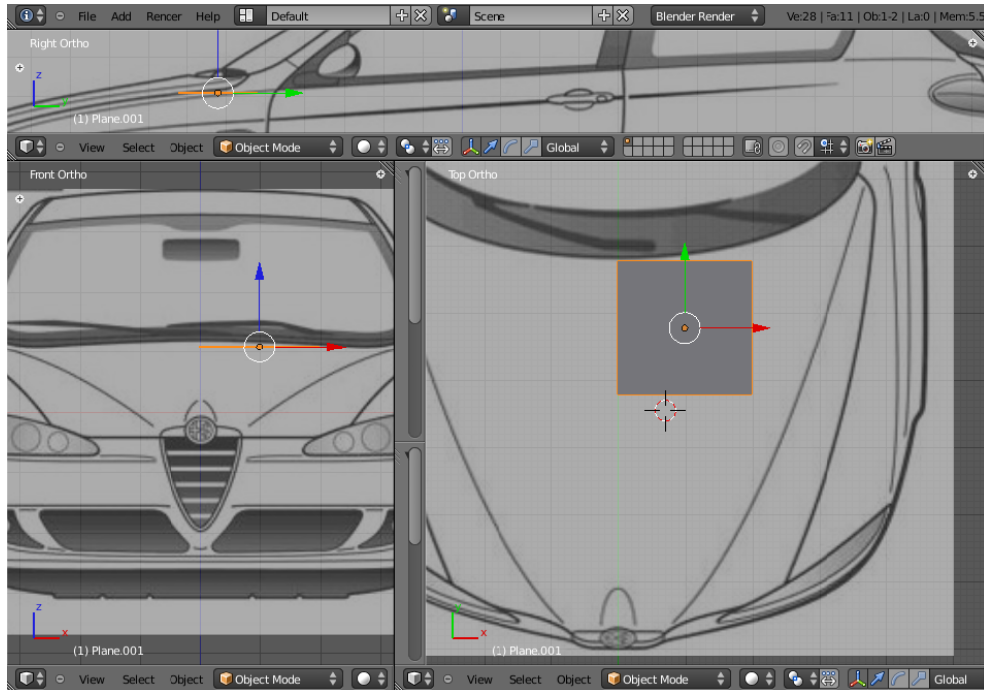
Odprimo Blender in ekran razdelimo na štiri dele, kot smo to storili s kocko v prejšnjem podpoglavju. V vsako ustrezno okno bomo dodali pripadajočo sliko: v okno *Front* dodamo sliko, ki prikazuje sprednji del avtomobila. Podobno naredimo še z okni *Right* (stranska slika), *Back* (zadnji del) in *Top* (zgornji del), kot je prikazano na sliki 6.12. Postavimo



Slika 6.12: Slike, ki prikazujejo vse perspektive avtomobila, vstavimo v Blender.

se v okno *Front* in z ustreznim ukazom dodajmo Bézierovo ploskev ter postavimo vse

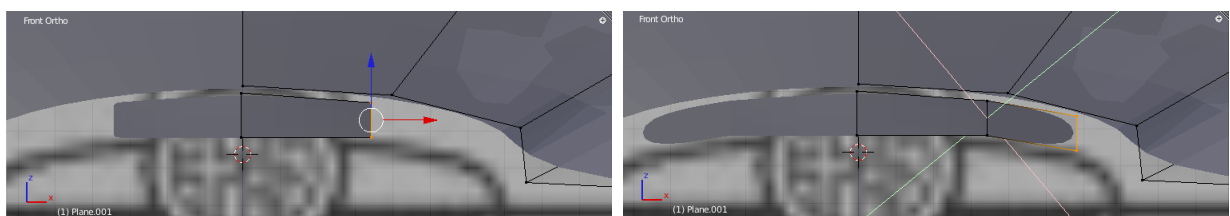
njene kontrolne točke na isto višino. Isto operacijo smo naredili v podpoglavju 4.11, ko smo oblikovali mizo. Uporabiti moramo ukaz *Mesh from Surf*, ki dano Bézierovo ploskev spremeni v objekt tipa *Mesh*. Dobljeno ploskev položimo na sprednji del karoserije, kot je prikazano na sliki 6.13. Avtomobil moramo sedaj “pokriti” s takšnimi ploskvami: slediti



Slika 6.13: Bézierova ploskev s kontrolnimi točkami na isti višini.

moramo slikam, da v Blenderju dobimo željeni model avtomobila. Pri tem si pomagamo s tremi glavnimi ukazi, ki jih bomo sedaj opisali. Ne bomo opisali oblikovanja avtomobila korak za korakom, ampak bomo podrobno pogledali le metode, ki jih uporabljamo pri oblikovanju, sicer bi bilo opisovanje preveč zamudno in obsežno. Glavni ukazi oziroma metode so:

1. *Extrude*. V podpoglavju 3.9 smo ta ukaz že uporabili, ko smo krivuljam dodajali nove kontrolne točke. Ukaz uporabimo tako, da označimo poljubno število točk in apliciramo ukaz *Extrude*. Na sliki 6.14 vidimo primer, kjer označimo dve točki in uporabimo ukaz, nakar nastaneta dve novi točki, ki ju lahko poljubno prestavimo. Na ta način ustvarimo dodatni del ploskve, ki nam bo pomagal pokriti vse dele

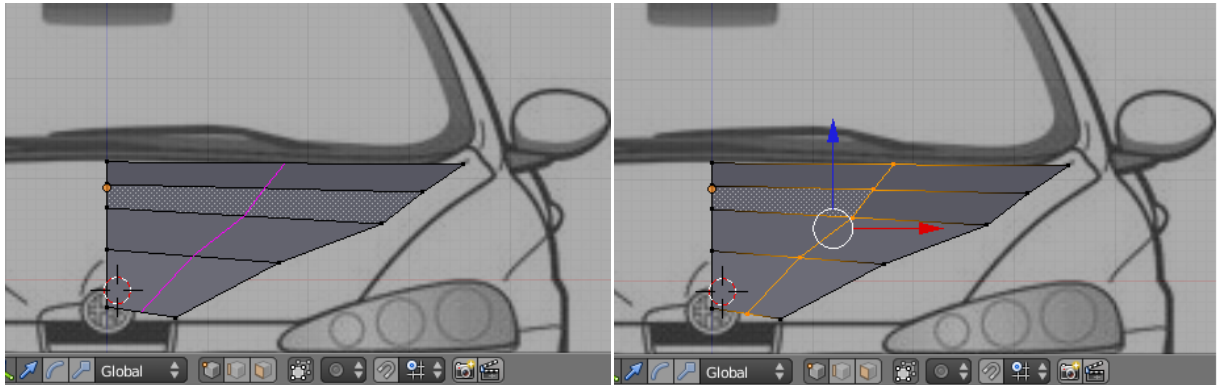


Slika 6.14: Primer uporabe ukaza *Extrude*.

avtomobila.

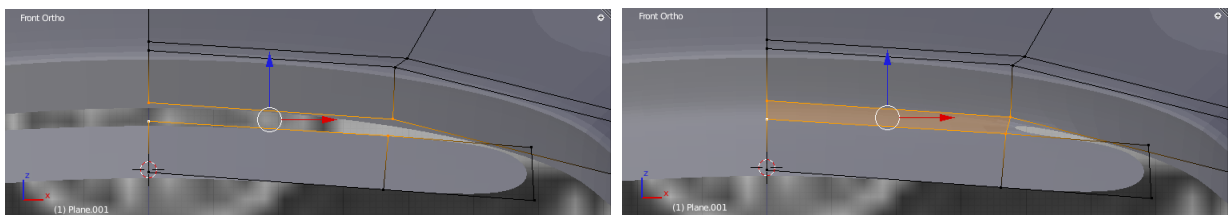
2. *RingCut*. Z uporabo tega ukaza se pojavi vodoravna oz. navpična vijolična črta (slika 6.15). Ko z miško potrdimo ukaz, se črta spremeni v črto z novimi kontrolnimi

točkami na ploskvi. Torej, z ukazom *RingCut* dodajamo nove kontrolne točke na že obstoječo ploskev. Pri ukazu *Extrude* pa smo dodajali nove točke, ki so ustvarile **nov** del ploskve. Z uporabo ukaza *Ringcut* postane naša ploskev bolj fleksibilna.



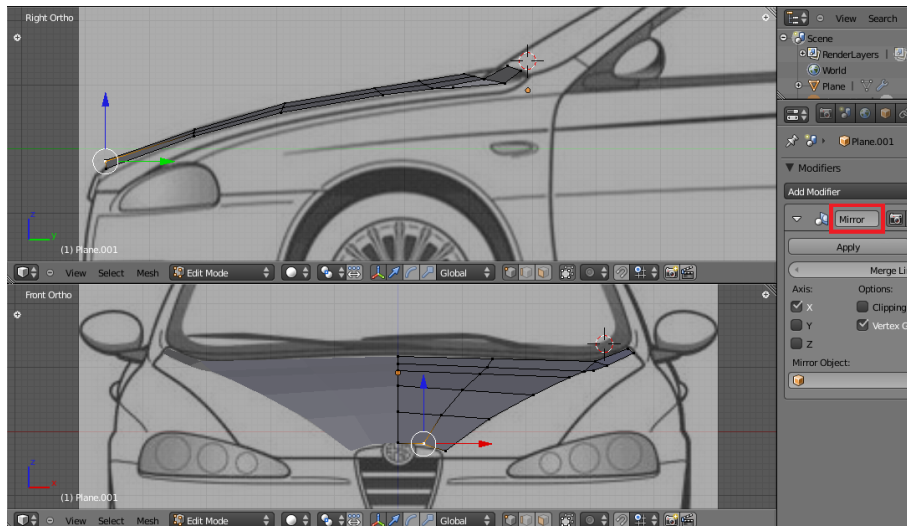
Slika 6.15: Z *RingCut* ploskvi poljubno dodamo nove točke.

3. *Face*. Med modeliranjem se večkrat zgodi, da želimo dve Bézierovi ploskvi združiti, da bosta tvorili eno samo ploskev. Slika 6.16 prikazuje spodnji del ploskve iz slike 6.15 in dodano novo ploskev. Radi bi obe ploskvi združili. V ta namen označimo tri ali štiri točke in uporabimo ukaz *Face*. Na sliki 6.16 vidimo, da se prostor med označenimi točkami "zapolni". To nam pomaga, da pokrijemo prazne prostore, ki nastajajo med ploskvami. Zakaj ne pokrijemo celotnega avtomobila z eno samo ploskvijo in se s tem izogibamo uporabi tega ukaza? Enostavno zato, ker bodo na ta način oblike nekaterih delov karoserije bolj poudarjene. To nam pomaga, ko želimo oblikovati vogale in robove karoserije. Na sliki 6.19 so nekateri vogali in robovi sprednjega dela karoserije poudarjeni. Model tega dela karoserije je namreč sestavljen iz več Bézierovih ploskev, ki smo jih združili z ukazom *Face*.

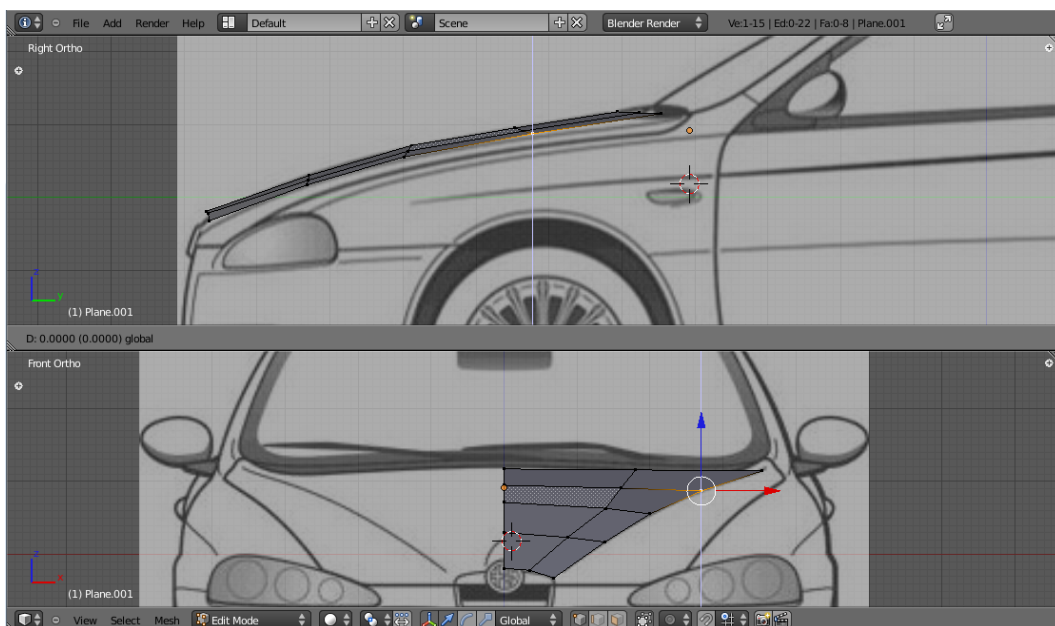


Slika 6.16: Z ukazom *Face* zapolnimo nastale prazne prostore med ploskvami.

Na sliki 6.16 vidimo, da oblikujemo samo polovico karoserije. Ko uporabimo ukaz *Face*, se prostor napolni ne samo med točkami, ki smo jih označili, ampak tudi na drugi polovici, kjer nimamo nobenih točk. To se zgodi, ker smo celotni ploskvi aplicirali *Modifier* tipa *Mirror* (vzdolž osi x). Ta je zelo uporaben pri računalniškem modeliranju, saj nam omogoča, da oblikujemo samo polovico objekta, druga polovica pa se zrcalno ustvari sama, kot je razvidno iz slike 6.17. *Modifier Mirror* lahko aktiviramo že od samega začetka oblikovanja objekta ali pa oblikujemo le polovico objekta in ukaz *Mirror* uporabimo šele na koncu.

Slika 6.17: *Modifier Mirror* nam prihrani veliko dela.

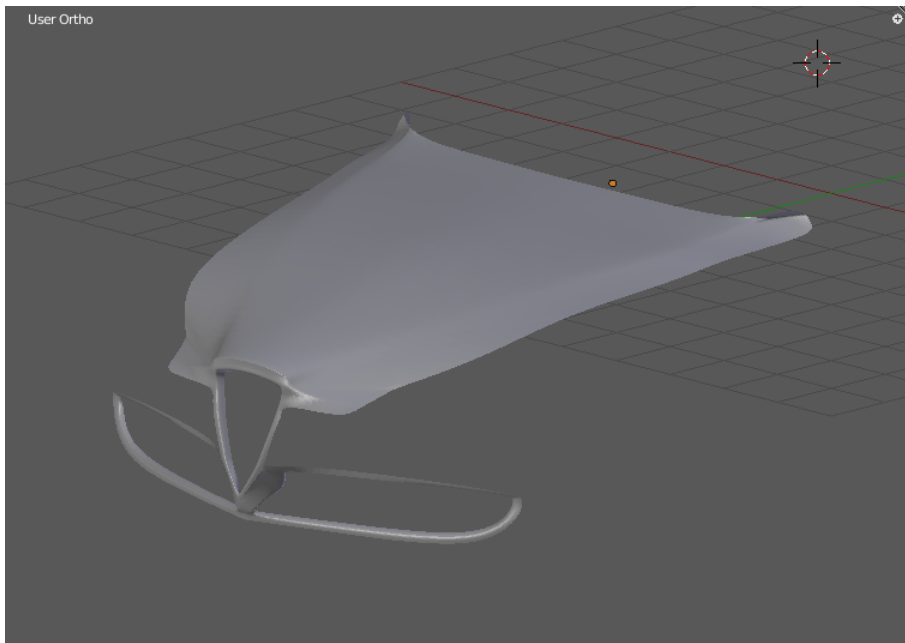
Razložimo sedaj, zakaj od samega začetka uporabljamo slike vseh štirih perspektiv avtomobila. Ko dodamo ploskev v Blender in jo položimo na sliko avtomobila, recimo v okno *Front*, lahko nastanejo hude napake. Oblikujemo namreč v treh dimenzijah in se lahko zgodi, da del avtomobila natančno pokrijemo v nekem oknu, ampak je tretja dimenzija popolnoma napačna. Na sliki 6.18 je prikazan primer, ko dobro pokrijemo sprednji del karoserije v oknu *Front*, ampak je potem potrebno popraviti položaj točk še v oknu *Right*, ki prikazuje stranski del avtomobila. Med modeliranjem je potrebno torej vseskozi paziti, da pravilno oblikujemo v vseh dimenzijah. V primeru, da moramo položaj določene točke popraviti, jo moramo le označiti in ustrezno premakniti. Operacija je prikazana na sliki 6.18.



Slika 6.18: Paziti moramo, da pravilno postavimo ploskve v vseh oknih Blenderja.

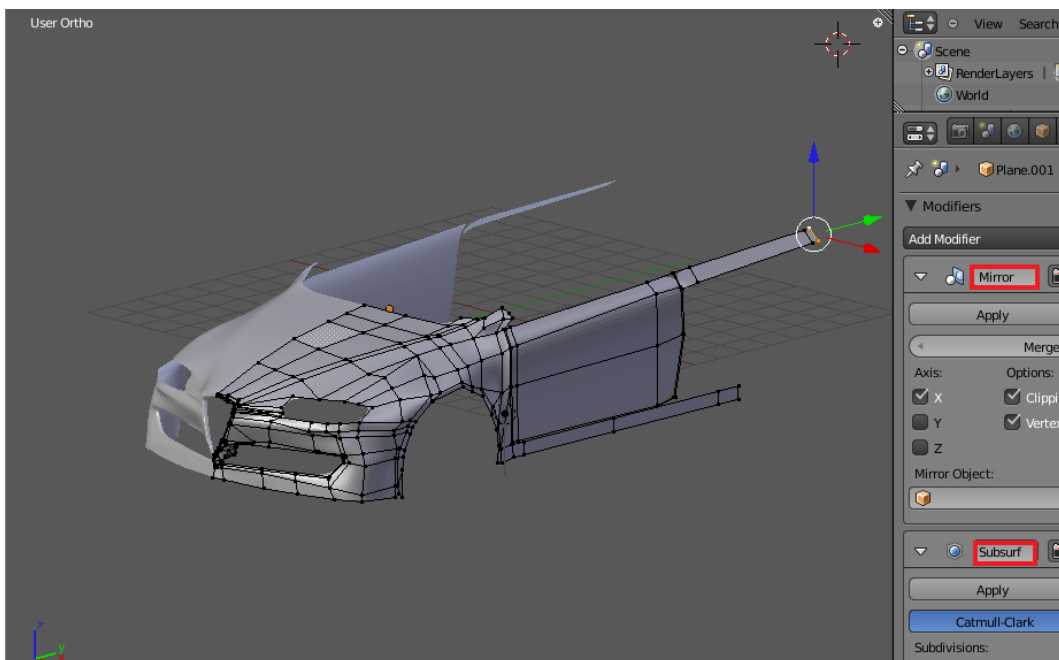
Potem ko zaključimo z modeliranjem sprednjega dela karoserije, dobimo rezultat na sliki 6.19, do katerega smo prišli z uporabo ukazov *Extrude*, *RingCut* in *Face*. Med oblikovan-

jem pa smo imeli *Modifier Mirror* aktiven in smo hkrati točke popravljali v vseh dimenzijah in v vseh perspektivah.



Slika 6.19: Sprednji del karoserije.

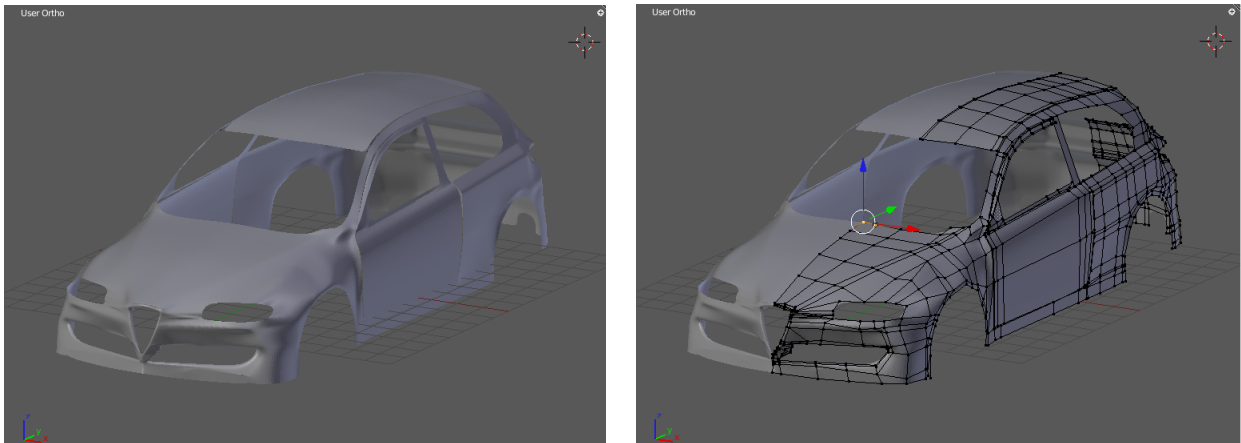
Poudariti moramo, da je med oblikovanjem avtomobila aktiven tudi *Modifier* tipa *Subdivision Surface*, ki “zgladi” ploskve. Spomnimo se, da smo v podpoglavju 3.9 uporabili ukaz *Smooth*, da bi končni videz steklenice zgladili. *Modifier Subdivision Surface* nas pripelje do podobnega rezultata, razlika je le v tem, da imamo tu na voljo parameter, s katerim lahko objekt večkrat zgladimo in torej večkrat popravimo njegovo obliko. Ukaz *Smooth* nam omogoča, da to naredimo samo enkrat.



Slika 6.20: Začetek oblikovanja zadnjega dela karoserije.

Na sliki 6.20 je prikazan začetek modeliranja zadnjega dela karoserije. Na desni strani slike sta z rdečo barvo označena *Modifier Mirror* in *Modifier Subdivision Surface*, ki sta vseskozi aktivirana. Rezultat pa smo dosegli z istimi metodami, ki smo jih že opisali.

Ko pokrijemo s ploskvami polovico avtomobila, smo zaključili z oblikovanjem karoserije. Slika 6.21 prikazuje njen končni videz v *Object Mode* in v *Edit Mode*, kjer lahko vidimo vse točke, ki sestavljajo ploskve.

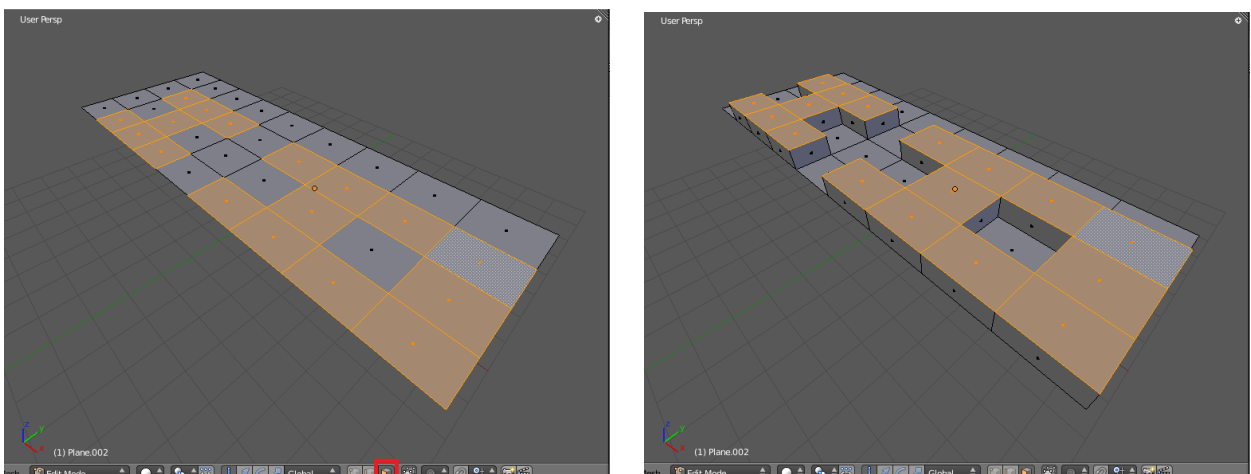


Slika 6.21: Končni model karoserije.

6.2.2 Pnevmatike

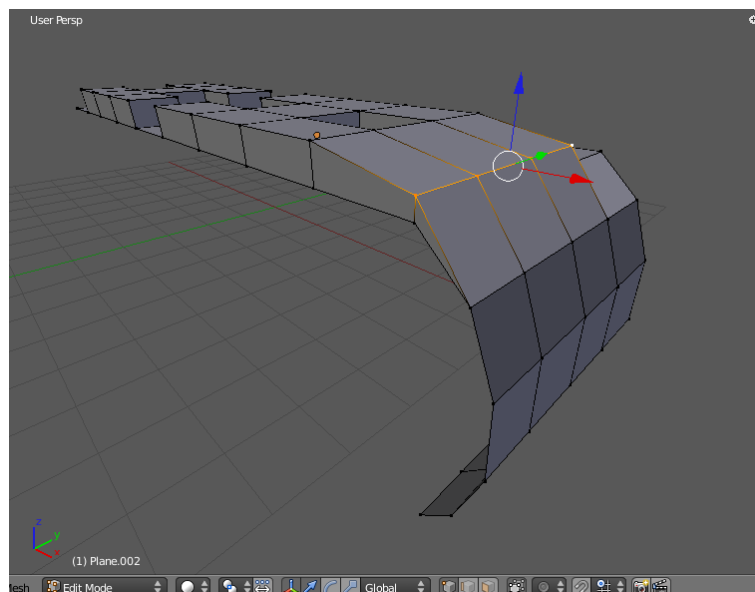
Karoserija je nedvomno najpomembnejši del našega modela avtomobila. Sedaj se moramo osredotočiti še na druge dele avtomobila, kot so pnevmatike, okna, luči in ostali detajli, ki avtomobil dopolnjujejo. Kot že rečeno, bomo pokazali le, kako lahko z Bézierovimi ploskvami oblikujemo avtomobilsko gumo.

V Blender dodajmo Bézierovo ploskev, postavimo vse njene kontrolne točke na isto višino in spremenimo ploskev v ploskev tipa *Mesh*, kot smo že storili pri modeliranju karoserije. Podaljšamo poljubni rob ploskve z ukazom *Extrude*. Preidimo v *Edit Mode* in izberimo možnost, da se na objektu prikažejo manjše ploskve, ki sestavljajo našo originalno ploskev. Na sliki 6.22 je prikazana naslednja operacija: označimo poljubno število majhnih ploskev in jih s pomočjo ukaza *Extrude* dvignemo. Na ta način spremenimo profil naše ploskve.



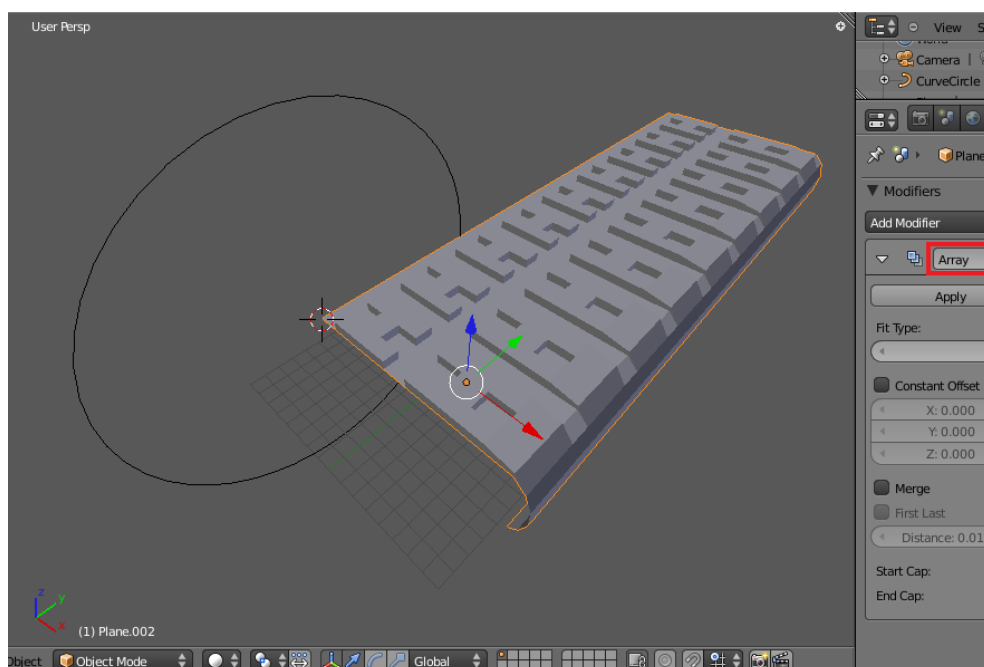
Slika 6.22: Začetek oblikovanja avtomobilske gume.

Označimo krajši rob ploskve in z ukazom *Extrude* podaljšajmo rob navzdol, kot je razvidno na sliki 6.23.



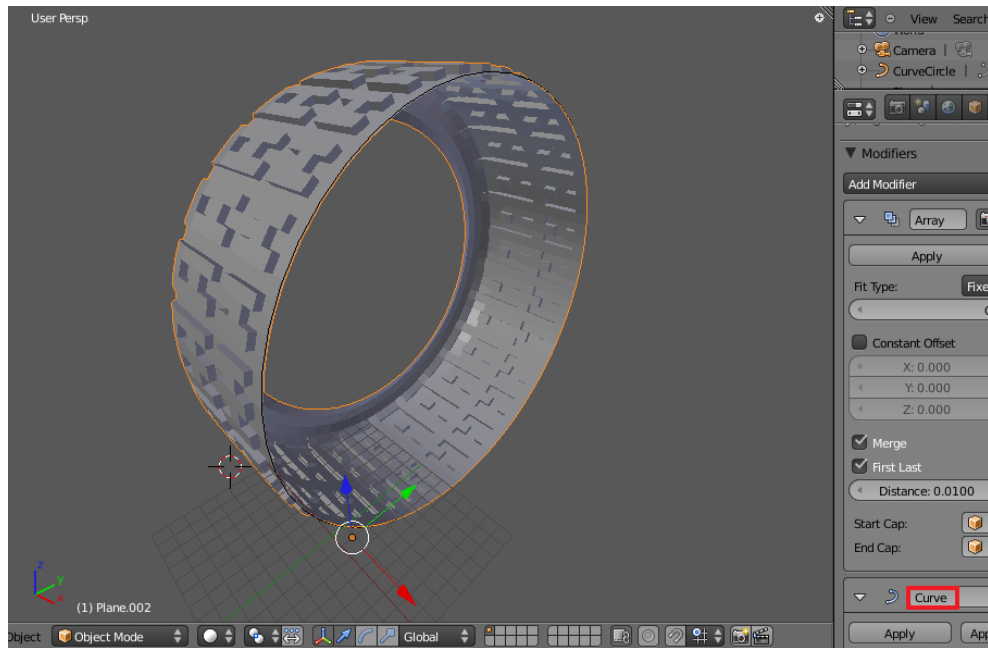
Slika 6.23: Profil avtomobilske gume dopolnimo.

Uporabimo *Modifier* tipa *Array*, ki dan element večkrat ponovi in postavi v vrsto. Ponovitev našega profila bo predstavljala del avtomobilske gume, kot je prikazano na sliki 6.24. Na sceno dodamo tudi krožnico okoli katere bomo ovili dobljen profil.



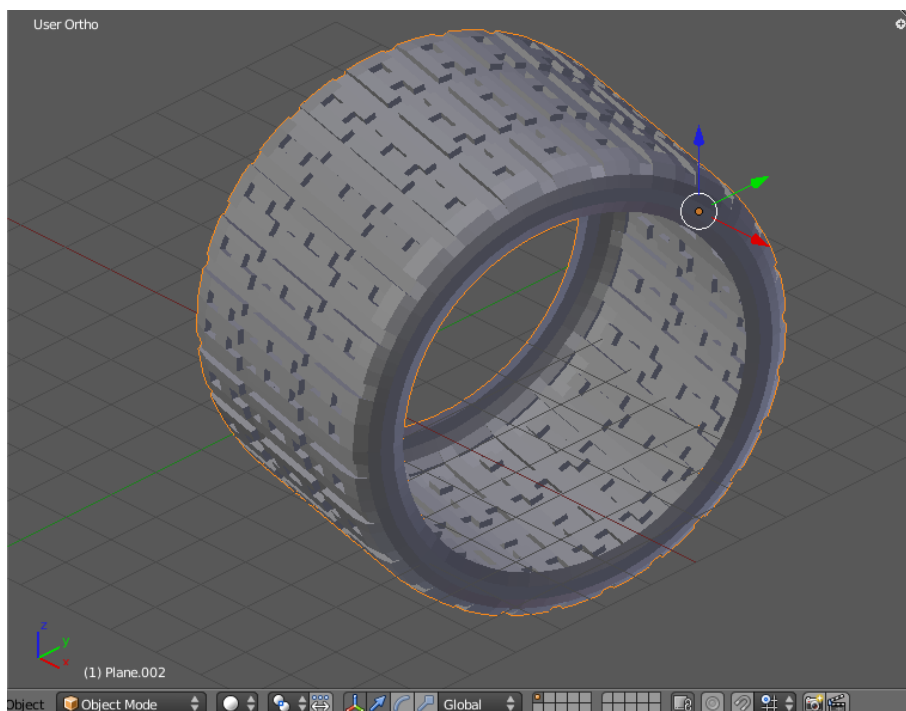
Slika 6.24: Prikaz krožnice okoli katere bomo ovili dobljen del gume.

Kot že rečeno, uporabimo *Modifier Curve*, ki nam omogoča, da dani objekt ovijemo okoli dane krivulje. V našem primeru ovijemo del avtomobilske gume okoli krožnice. Rezultat je prikazan na sliki 6.25.



Slika 6.25: Z *Modifier Curve* smo profil gume postavili v obliko kroga.

V bistvu smo z oblikovanjem pnevmatike že zaključili. Dobljen objekt samo zrealimo s pomočjo *Modifier Mirror*, ki smo ga vseskozi uporabljali za karoserijo avtomobila. Končni rezultat je prikazan na sliki 6.26. Označimo dobljeno pnevmatiko in uporabimo ukaz *Duplicate Object*, s katerim takoj ustvarimo kopijo avtomobilske gume. To ponovimo še dvakrat, da dobimo vse štiri pnevmatike, ki jih bomo dodali h karoseriji.

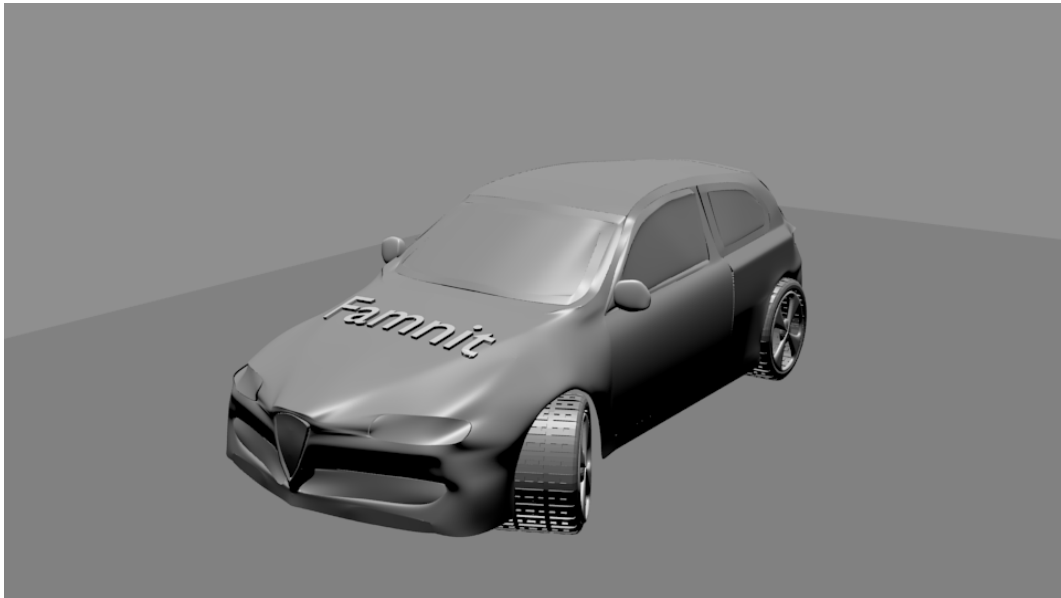


Slika 6.26: Avtomobilska guma, ki jo bomo kasneje dodali h karoseriji.

6.2.3 Končni model

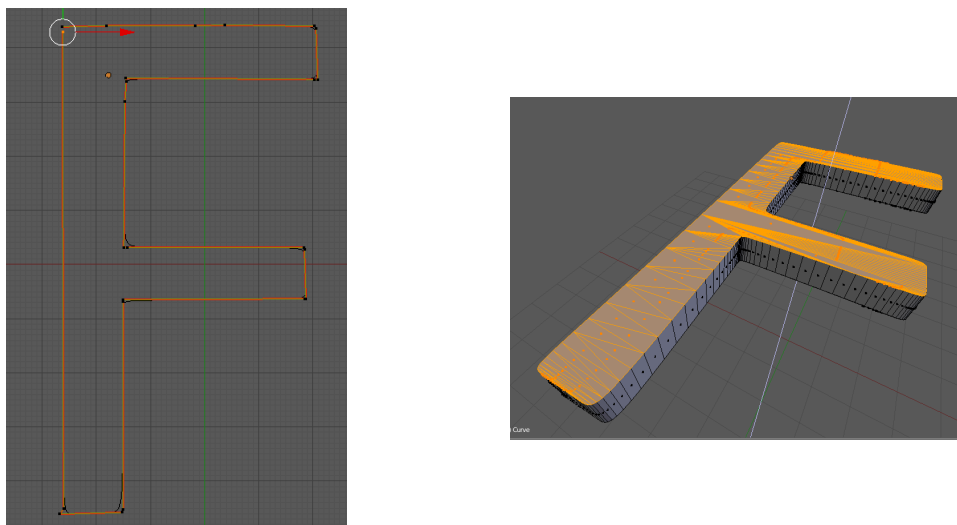
Da bi dokončno zaključili z opisom modeliranja avtomobila, bi se morali ustaviti še na ostalih detajlih, kot so okna, pokrovi koles, luči in podobno. Ker bi bil sicer opis zelo dolg in obsežen, povejmo le, da za to uporabimo vse ukaze in metode, ki smo jih spoznali v tem poglavju in v podpoglavjih 3.9 oz. 4.11.

Ko karoseriji dodamo pnevmatike in vse zgoraj omenjene detajle, dobimo model na sliki 6.27.



Slika 6.27: Končni model avtomobila.

Črke, ki sestavljajo besedo *Famnit*, smo dobili tako, da smo njihovo obliko določili z Bézierovimi krivuljami, kot smo že storili v podpoglavju 3.9 pri oblikovanju prereza steklenice. Kar smo dobili, smo spremenili v ploskev tipa *Mesh*, da smo napolnili prostor med krivuljami. Ploskev v obliki črke smo na koncu dvignili s pomočjo ukaza *Extrude* (slika 6.28).



Slika 6.28: Modeliranje črk z Bézierovimi krivuljami.

Glavno fazo modeliranja smo zaključili. Končnemu modelu lahko dodamo barve in prozornost ter dodamo luč in posledično sence. Želimo se približati realističnemu prikazu avtomobila. Na sliki 6.29 je razvidno, koliko različnih efektov lahko dobimo z barvanjem poljubnega objekta. Karoserijo smo pobarvali s svetlo zeleno barvo in dodali tak efekt barvi, da se svetloba odbija na njej. Okna imajo svetlo modro barvo, katerim smo dodali tudi malo prozornosti in odbojnosti. V ta namen smo dodali Bézierovo ploskev, ki prikazuje tla, na katerih stoji avtomobil, in s pomočjo orodja *Texture* smo nanjo zalpili sliko, ki prikazuje površino s kamni. Vidimo, da se na nekaterih delih avtomobila, kot so npr. okna, odbija svetloba in vidi odsev tal. Avtomobilske gume smo obarvali s svetlejšo sivo barvo, da poudarimo izgled profila gume, ki smo ga oblikovali v prejšnjem podpoglavju. Napis "Famnit" je rdeče barve. Tukaj smo črke pobarvali z navadno rdečo barvo brez kakršnihkoli efektov prozornosti ali odbojnosti.



Slika 6.29: Barve in drugi efekti nas privedejo do realističnega prikaza avtomobila.

Poglavje 7

Zaključek

V magistrskem delu smo spoznali matematično ozadje Bézierovih krivulj in ploskev, ki so predstavljale pravo novost pri geometrijskem načrtovanju. Videli smo, da so Bézierove krivulje primer parametričnih krivulj, ki so definirane s kontrolnimi točkami, v katerih se skriva moč takih krivulj: kontrolne točke definirajo in tudi spreminjajo same krivulje. Spoznali smo, da velja podobno tudi za Bézierove ploskve. Danes se matematiki veliko posvečajo raziskovanju Bézierovih trikotnikov in njihovih lastnosti, ker poljubne površine bolje pokrivamo s trikotniki kakor s pravokotniki. V takih primerih se raje posvečajo trikotnim domenam. V magistrskem delu smo opisali le glavne lastnosti Bézierovih trikotnikov. Kdor se želi še bolj poglobiti, ima na voljo knjigo G. Farina *Curves and Surfaces for CAGD - A practical guide*, po kateri je bilo vzetih veliko definicij in trditev magistrskega dela. Glavni cilj magistrskega dela je bila predstavitev Bézierovih krivulj in ploskev tako na teoretični kot na praktični ravni, da bi pridobili splošno predstavo glede njihove uporabnosti. V ta namen smo predstavili računalniški program Blender in si ogledali njegove glavne ukaze in metode, s katerimi smo oblikovali steklenico, mizo, posodo in avtomobil. Zavedati pa se moramo, da nam Blender nudi še veliko več orodij in zanimivosti, ki omogočajo, da naše modele objektov predstavimo še bolj realistično. Z Blenderjem lahko recimo ustvarimo animacije z našimi modeli. Za več informacij je na voljo spletna stran www.blender.org in [22]. V zadnjem poglavju smo spoznali primer, kako so lahko Bézierove krivulje in ploskve uporabne v avtomobilski industriji. Danes jih veliko uporabljajo tudi v letalski industriji, animacijah, risankah in na drugih področjih, kjer se uporablja računalniška grafika. Vsekakor sta francoska matematika P. Bézier in P. de Casteljau bila začetnika matematičnega področja, ki je in bo veliko prispeval k računalniški grafiki.

Literatura

- [1] R. Barnhill, Surfaces in computer aided geometric design: a survey with new results, *Computer Aided Geometric Design* 2 (1985), 1-17.
- [2] C. de Boor, Bicubic spline interpolation, *Journal of Mathematical Physics* 41 (1962), 212-218.
- [3] C. de Boor, *Geometric Modeling: Algorithms and New Trends*, Society for Industrial and Applied Mathematics, Philadelphia, 1987.
- [4] T. DeRose, Composing Bézier simplices, *ACM Transactions on Graphics* 7 (1988), 198-221.
- [5] G. Farin, *Curves and surfaces for CAGD*, Fifth ed., Morgan Kaufmann publishers, San Francisco, 2006.
- [6] G. Farin, *Konstruktion und Eigenschaften von Bézier-Kurven und Flächen*, Master's thesis, Technical University Braunschweig, Braunschweig, Germany, 1977.
- [7] G. Farin, *Bézier polynomials over triangles and the construction of piecewise C^r polynomials*, Technical report TR/91, Brunel University, Uxbridge, England, 1980.
- [8] G. Farin, Triangular Bernstein-Bézier patches, *Computer Aided Geometric Design* 3 (1986), 83-128.
- [9] J. Ferguson, Multivariable curve interpolation, *Journal of the Association for Computing Machinery* 11 (1964), 221-228.
- [10] A. R. Forrest, Interactive interpolation and approximation by Bézier polynomials, *The Computer Journal* 15 (1972), 71-79.
- [11] L. Frederickson, *Triangular spline interpolation/generalized triangular splines*, Technical report no. 6/70 and 7/71, Department of Mathematics, Lakehead University, Orillia, Canada, 1971.
- [12] D. Kim, *Cones on Bézier curves and surfaces*, PhD thesis, Industrial and Operations Engineering Department, University of Michigan, Ann Arbor, United States of America, 1990.
- [13] C. S. Leang, Composite cubic Bézier surface with C^r continuity, *Sunway Academic Journal* 3 (2006), 21-33.
- [14] F. Moebius, *August Ferdinand Moebius, Gesammelte Werke*, Verlag von S. Hirzel, Leipzig, 1885.

- [15] G. Peters, *Interactive computer graphics application of the parametric bicubic surface to engineering design problems*, McDonnell Douglas Automation Company, Missouri, 1973.
- [16] J. Petrišič, *Interpolacija in osnove računalniške grafike*, Fakulteta za strojništvo, Ljubljana, 1999.
- [17] M. Sabin, *The use of piecewise forms for the numerical representation of shape*, PhD thesis, Hungarian Academy of Sciences, Budapest, Hungary, 1976.
- [18] P. Sablonnière, *Bases de Bernstein et approximants splines*, PhD thesis, University of Lille, Lille, France, 1982.
- [19] F. Wolter, S. Tuohy, Curvature computations for degenerate surface patches, *Computer Aided Geometric Design* 9 (1992), 241-270.
- [20] <http://ucilnica.fmf.uni-lj.si/file.php/21/CAGD.pdf>. Dostopno na spletu 1. julij 2012.
- [21] http://www.fmf.uni-lj.si/~juvan/Racunalnistvo3/gradivo/opt_konveksnost_mnoz.pdf. Dostopno na spletu 1. julij 2012.
- [22] <http://wiki.blender.org/index.php/Doc:2.6/Manual>. Dostopno na spletu 1. julij 2012.

Priloga A

Funkcija `deCasteljau` izračuna za dane točke Bézierovo krivuljo z de Casteljauovim algoritmom in na isto sliko nariše krivuljo in njen kontrolni poligon.

```

1
2 function [b]=deCasteljau(kontrolne,stTock)
3 % funkcija narise Bezierovo krivuljo, ki ima kontrolne tocke
4 % 'kontrolne', ki je tabela velikosti d x n+1. d je 2 ali 3 in je
5 % dimenzija prostora. stTock je parameter, ki pove, kako na
6 % gosto naracunamo in narisemo tocke na krivulji
7 if size(kontrolne)(1)==2 %preverimo, ali racunamo v ravnini ali
8 % v prostoru
9 n = size(kontrolne)(2)-1;
10 t=linspace(0,1,stTock); % interval [0,1] razdelimo na toliko
11 % delov kot jih je predpisano v spremenljivki stTock
12 B=zeros(n+1,n+1,size(kontrolne)(1)); % ustvarimo tenzor velikosti
13 % (n+1) x (n+1) x 2 s samimi niclami
14 for k=1:(n+1)
15 B(k,1,:)=kontrolne(:,k); % s prvo for zanko ustavimo v prvi
16 % stolpec tenzorja vektorje kontrolnih tock
17 end
18 for l=2:n+1
19 for m=1:n
20 B(m,l,:)=zeros(1,size(kontrolne)(1));
21 end
22 end
23 b=zeros(2,stTock); % pripravimo matriko velikosti 2 x stTock,
24 % ki bo vsebovala koordinate tock, ki bodo lezale na Bez.krivulji
25 for w=1:stTock
26 b(:,w)=ses_mtr(w,kontrolne,stTock); % funkcija ses_mtr sestavi
27 % matriko in bomo tako dobili tocke krivulje
28 end
29 clf;
30 hold on; % narisemo vse na isto sliko
31 plot(kontrolne(1,:),kontrolne(2,:),'o'); % narisemo
32 % kontrolne tocke
33 plot(kontrolne(1,:),kontrolne(2,:),'b'); % jih povezemo
34 plot(b(1,:),b(2,:),'r'); % povezemo se Bezierove tocke
35 hold off;
36 endif
37
38 if size(kontrolne)(1)==3 % ce pa dane tocke leziyo v
39 % prostoru, se izvede ta del programa, ki je isti kot

```

```

40 %prejsnji. Razlika je v tem, da bomo imeli tenzor
41 % velikosti n+1 x n+1 x 3 in da se bo krivulja izrisala
42 % v prostoru zato uporabimo ukaz plot3
43
44 n = size(kontrolne)(2)-1;
45 t=linspace(0,1,stTock);
46 B=zeros(n+1,n+1,size(kontrolne)(1));
47 for k=1:(n+1)
48 B(k,1,:)=kontrolne(:,k);
49 end
50 for l=2:n+1
51 for m=1:n
52 B(m,l,:)=zeros(1,size(kontrolne)(1));
53 end
54 end
55 b=zeros(3,stTock);
56 for w=1:stTock
57 b(:,w)=ses_mtr(w,kontrolne,stTock);
58 end
59 clf;
60 grid on;
61 hold on;
62 plot3(kontrolne(1,:),kontrolne(2,:),kontrolne(3,:), 'o');
63 plot3(kontrolne(1,:),kontrolne(2,:),kontrolne(3,:), 'r');
64 plot3(b(1,:),b(2,:),b(3,:), 'r');
65 hold off;
66 endif
67 end

```

Funkcija ses_mtr, ki jo uporabimo v funkciji deCasteljau pa je naslednja:

```

1
2 function [l]=ses_mtr(w,kontrolne,stTock)
3 n = size(kontrolne)(2)-1; % definiramo se enkrat
4 % spremenljivke n,t in nastavitve tenzorja B
5 t=linspace(0,1,stTock);
6
7 B=zeros(n+1,n+1,size(kontrolne)(1));
8 for k=1:(n+1)
9 B(k,1,:)=kontrolne(:,k);
10 end
11
12 for l=2:n+1
13 for m=1:n
14 B(m,l,:)=zeros(1,size(kontrolne)(1));
15 end
16 end
17 for j=2:(n+1)
18 for i=2:(n+1)
19 B(i,j,:)=(1-t(w))*B(i-1,j-1,:)+t(w)*B(i,j-1,:); % predpis,
20 % ki definira Bezierovo krivuljo. S tem lahko pridemo do vektorja,
21 % ki bo predstavljal točko na krivulji za dani t

```

```
22 end
23 end
24 l=B(n+1,n+1,:); % tocka (oziroma vektor dolzine 2,ki predstavlja
25 % tocko na krivulji), ki lezi na tej poziciji tenzorja, lezi na
26 % krivulji in s ponavljanjem teh for zank dobimo vse tocke
27 % na krivulji za vse dane t-je v vektorju t
```


Priloga B

Funkcija `matricna_oblika` izračuna točko na Bézierovi krivulji za dani parameter `t` in pri tem uporabi matrike in vektorje, ki smo jih definirali v podpoglavju 3.8

```
1
2 function c=matricna_oblika(kontrolne,t) % vhodna podatka
3 % sta matrika kontrolnih točk 'kontrolne' velikosti
4 % (n+1) x d, kjer je n stopnja krivulje in d dimenzija prostora,
5 % in parameter t
6 n=size(b)(1)-1;
7 A=zeros(n+1,n+1);
8 t_vektor=zeros(1,n+1);
9
10 for k=1:n+1
11 t_vektor(k)=t^(n-(k-1));% sestavimo vektor t, ki bo vseboval
12 % potence parametra t
13 endfor
14
15 for i=1:n+1
16 for j=1:n+1
17 A(i,j)=(-1)^(n-(i-1)-(j-1))*bincoeff(n-(i-1),(j-1))*bincoeff(n,(i-1));
18 % sestavimo matriko A, kot je definirana v podpoglavju 3.8
19 endfor
20 endfor
21
22 B=t_vektor*A; % zmnožimo vektor t in matriko A
23
24 v=zeros(n+1,size(b)(2)); % kar dobimo pomnožimo z vektorjem
25 % kontrolnih točk in dobimo Bezierovo točko
26 for r=1:n+1
27 v(r,:)=B(1,r)*b(r,:);
28 endfor
29
30 c=sum(v);
```

Priloga C

Funkcija `hiper_par` implementira matricno obliko enačbe (4.2) in izračuna točke na hiperboličnem paraboloidu.

```

1
2 function hiper_par(A) % matrika A je velikosti 2x6 in vsebuje
3 % vse stiri dane točke b_00,b_01,b_10,b_11
4 s=0; % stevec s potrebujemo zato, da bomo končne rezultate
5 % zapisovali v stolpce matrike Q
6 u=linspace(0,1,100); % interval [0,1] razdelimo na 100 delov
7 B=zeros(length(u),2*length(u)); % matrika B bo vsebovala vse
8 % možne pare (u,v), kjer 0 ≤ u,v ≤ 1
9
10 for i=1:length(u)
11 for j=1:length(u)
12     B(j,2*i-1)=u(j); % sestavimo matriko B, ki bo vsebovala
13 % vrednosti za u in v
14     B(j,i+1)=u(i);
15 end
16 end
17
18 for l=1:length(u)
19 for p=1:length(u)
20
21     m=B(l,2*p-1); % na vsakem koraku for zank izberemo vrednost
22 % za u
23     n=B(l,p+1); % na vsakem koraku for zank izberemo vrednost
24 % za v
25
26     c=[1-m m];
27     d=[1-n n];
28
29     v=[c(1)*A(1,1:3)+c(2)*A(1,4:6) c(1)*A(2,1:3)+c(2)*A(2,4:6)];
30 % implementiramo definicijo hiperbolicnega paraboloida
31     v2=v(1:3)*d(1)+v(4:6)*d(2);
32     v2=v2';
33     s=s+1;
34     Q(1:3,s)=v2; % vsak rezultat (točka na hiperbolicnem parabo-
35 % loidu) zabeležimo v s-ti stolpec matrike Q
36 end
37 end
38
39 end

```

Priloga Č

Funkcija `Bezier_ploskev` izračuna Bézierove točke in nariše Bézierovo ploskev.

```

1
2
3 function Q=Bezier_ploskev(B,stTock) % matrika B bo velikosti
4 % (m+1) x 3(n+1), kjer je (m,n) stopnja Bezierove krivulje.
5 % V matriko napisemo kontrolne točke vodoravno po naslednjem
6 % vrstnem redu:
7 % b_00, b_01,..., b_0n
8 % b_10, b_11,..., b_1n
9 % ... ..
10 % b_m0, b_m1,...,b_mn
11 % stTock pa je stevilo, ki nam pove, kako na gosto bomo razdelili
12 % interval [0,1], po katerem bosta tekla parametra u in v
13
14 m=size(B)(1)-1;
15 n=(size(B)(2)/3)-1;
16
17 for k=1:m+1
18 for l=1:n+1
19
20     A(k,l,:)=B(k,3*l-2:3*l); % s for zankama zapisemo točke v
21     % matriki B v tenzor A
22
23 end
24 end
25
26
27 Q=zeros(stTock,stTock,3); % pripravimo tenzor Q, v katerega
28 % bomo zapisovali dobljene Bezierove točke
29
30 x=1; % parameter x bo tekkel po vrsticah tenzorja Q
31 for u=linspace(0,1,stTock) % definiramo parameter u, ki bo tekkel
32 % po intervalu [0,1]
33 y=1; % parameter y bo tekkel po stolpcih tenzorja Q
34 for v=linspace(0,1,stTock) % definiramo parameter v, ki bo tekkel
35 % po intervalu [0,1]
36
37 for i=0:m
38 for j=0:n
39
40 Q(x,y,:)=Q(x,y,:)+Bernstein(m,i,u)*Bernstein(n,j,v)*A(i+1,j+1,:);

```

```

41 % Bernsteinova oblika Bezierove ploskve ( po definiciji )
42
43 end
44 end
45
46 y=y+1;
47 end
48 x=x+1;
49 end
50
51 clf;
52 mesh(A(:,:,1),A(:,:,2),A(:,:,3)); % narisemo kontrolne tocke oz.
53 % kontrolno mrezo
54 hold on;
55 surf(Q(:,:,1),Q(:,:,2),Q(:,:,3)); % na isto sliko narisemo se
56 % Bezierovo ploskev
57 end

```

Funkcija Bernstein, ki jo uporabimo v prejšnji kodi v vrstici 40, pa izračuna vrednost Bernsteinovega polinoma.

```

1
2 function B=Bernstein(n,i,t)
3
4 B=nchoosek(n,i)*(t^i)*(1-t)^(n-i);
5 % Bernsteinov polinom po definiciji
6
7 end

```

Priloga D

Funkcija `deCasteljau_trik` implementira de Casteljauov algoritem za Bézierov trikotnik in ga narise

```

1
2 function Q=deCasteljau_trik(kontrolne,stTock) % funkcija dobi kot
3 % vhodna podatka matriko kontrolnih tock 'kontrolne', ki je
4 % velikosti (n+1) x (n+1), kjer je n dano naravno stevilo. Matrika
5 % je diagonalna in jo dobimo tako, da de Casteljauovo shemo
6 % trikotne oblike premaknemo "levo". Za n=3 bi recimo dobili tako
7 % matriko:
8 % b_030    0      0      0
9 % b_021 b_120    0      0
10 % b_012 b_b111 b_210    0
11 % b_003 b_102   b_201   b_300
12
13 n=size(kontrolne)(1)-1;
14
15 m = stTock-1; % 'm' je parameter, ki razdeli interval [0,1] s
16 % pomocjo parametra 'stTock'
17
18 for u = 0:1/m:1
19     t = round(u*m+1); % parametra s in t bosta tekla po vrsticah in
20 % stolpcih tenzorja Q
21     s=1;
22     for v = 0:1/m:(1-u)
23         for w = 1-u-v; % for zanke bodo tekle samo po parametrih
24             % u,v,w, za katere velja u+v+w=1
25
26             Q(1:3,s,t)=tocka(kontrolne,u,v,w); % funkcija 'tocka'
27 % izracuna za dano kontrolno mrezo in dane parametre u,v,w tocko
28 % na ploskvi in jo zapise v tenzor Q
29             s=s+1;
30         end
31     end
32 end
33
34 clf;
35
36 for j=1:m
37     for i=1:m-j+1
38         A =[Q(:,i,j) Q(:,i+1,j) Q(:,i,j+1)]'; % definiramo matriko A,
39 % ki bo imela po stolpcih shranjene x,y,z-komponente

```

```

40 % oglisc treh tock enega trikotnika
41     hold on;
42     % funkcija 'patch' narise trikotnik, kjer prvi parameter
43 % predstavlja x-komponente oglisc, drugi predstavlja
44 % y-komponente, tretji pa z-komponente
45     patch(A(:,1),A(:,2),A(:,3),'m');
46     end;
47     for i=1:m-j
48         A=[Q(:,i+1,j) Q(:,i,j+1) Q(:,i+1,j+1)]';
49         hold on;
50
51         patch(A(:,1),A(:,2),A(:,3),'y');
52     end
53 end
54 end

```

Funkcija tocka izračuna točko na Bézierovem trikotniku pri danih parametrih u, v in w

```

1
2 function a=tocka(kontrolne,u,v,w) % izracuna za dano matriko
3 % kontrolnih tock in dane parametre u,v,w tocko na ploskvi
4 % s pomocjo funkcije 'pomozna'
5
6 n=size(kontrolne)(1)-1;
7
8 for i=1:n % for zanka izvede funkcijo 'pomozna' n-krat. Za vec in-
9 % formacij glej funkcijo 'pomozna'
10
11     kontrolne2=pomozna(kontrolne,u,v,w);
12     kontrolne=kontrolne2;
13
14 end
15
16 a=kontrolne';
17 end

```

Funkcija pomozna izračuna natanko en korak de Casteljauevega algoritma

```

1
2 function k2=pomozna(kontrolne,u,v,w) % funkcija dobi kot vhodne po-
3 % datke tocke b_i^r iz de Casteljauevega algoritma in izracuna
4 % naslednje tocke s pomocjo glavnega koraka algoritma
5 n=size(kontrolne)(1)-1;
6
7 for i=1:n % s pomocjo for zanke izpeljemo glavni korak
8 for r=0:(i-1) % de Casteljauevega algoritma (glej podp. 5.1)
9
10     k2(i,3*(i-r)-2:3*(i-r))=u*kontrolne(i+1,3*((i-r)+1)-2:3*
11 ((i-r)+1))+v*kontrolne(i,3*(i-r)-2:3*(i-r))+
12 w*kontrolne(i+1,3*(i-r)-2:3*(i-r));
13 end
14 end
15 end

```