

UNIVERZA V LJUBLJANI,
FAKULTETA ZA MATEMATIKO IN FIZIKO

Liam Mislej

Projekcije Bézierove krivulje v smeri vektorja na ravnino

Projektno delo pri predmetu Matematično modeliranje 20/21

Lucija, februar 2021

Metorja:

Izred. Prof. dr. Emil Žagar

Asis. Dr. Tadej Kanduč

Izvleček:

V projektni nalogi bom predstavil Bézierove krivulje, opisal njihov izum in zgodovino. Opisal bom, kako se točke parametrične krivulje računa in predstavil nekaj njenih bolj poznanih lastnosti. Nato bom razložil projekcijo posameznih točk na ravnino v smeri vektorja, pri tem pa uporabil lastnosti podobnih trikotnikov. Ugotovljeno bom posplošil še na projekcijo točke v smeri vektorja na ravnino v poljubnih dimezijah tako, da bom ustvaril linearno transformacijo F . Za prikazovanje oz. izris krivulj bom uporabil programski jezik Matlab.

Kazalo

Projekcije Bézierove krivulje v smeri danega vektorja na ravnino.....	1
1. Bézierova krivulja	4
1.1. Izum	4
1.2. De Casteljauov algoritem	4
1.3. Bernsteinovi polinomi	4
1.4. Bernsteinova oblika Bézierove krivulje.....	5
1.5. Nekaj pomembnejše lastnosti Bézierove krivulje	5
2. Projiciranje Bézierovih krivulj v prostoru na ravnino, $z = 0$ v smeri vektorja.....	6
2.1 Projiciranje točke na ravnino $z=0$ v smeri vektorja	6
2.2 Posplošitev projekcije na n -dimenzionalen prostor	8
2.3 Numerična analiza na primeru z uporabo programskega jezika Matlab.....	9
3. Projiciranje na poljubno ravnino v \mathbb{R}^3	12
4. Zaključek	14
5. Viri	14

1. Bézierova krivulja

Bézierova krivulja je parametrična krivulja, najpogosteje uporabljena v računalniški grafiki in podobnim področjem. Krivulja, ki je v sorodu z Bernsteinovim polinomom je poimenovana po Pierreu Bézierju, ki jo je v 60-ih letih prejšnjega stoletja uporabljal za izrisovanje oblik avtomobilov pri podjetju Renault. Dandanes jo najpogosteje najdemo pri risanju in oblikovanju raznih slogov digitalnih pisav, animacijah in vektorskih grafikah.

1.1. Izum

Matematična osnova za Bézierovo krivuljo in Bernsteinove polinome je bila postavljena že leta 1912, vendar so bili polinomi aplicirani šele 50 let kasneje, ko je matematik Paul de Casteljau, leta 1959 razvil t.i. de Casteljev algoritem, ki je numerično stabilna metoda za evaluacijo krivulj. Takrat je postal tudi prvi, ki je krivulje uporabljal za dizajn avtodelov s pomočjo računalnika pri francoskem avto-proizvajalcu Citroen. Krivulje so postale bolj poznane v 60-ih letih prejšnjega stoletja, ko je francoski inženir Pierre Bézier pričel pogosteje objavljati članke o njihovi uporabi. Po njem so krivulje tudi poimenovane.

1.2. De Casteljauov algoritem

Bézierove krivulje lahko predstavimo na različne načine, sam pa bom predstavil prej omenjeni numerično stabilen algoritem, ki si ga je zamislil P. de Casteljau.

Naj bodo $b_0, b_1, \dots, b_n \in \mathbb{R}^3$ dane točke in naj bo $t \in \mathbb{R}$ parameter. Točke $b_i, i = 0, 1, \dots, n$, imenujemo *kontrolne točke* ali Bézierove točke, ki skupaj tvorijo *kontrolni poligon* ali *Bezierov poligon* krivulje. Nato izračunamo točke:

$$b_i^r(t) = (1-t)b_i^{r-1}(t) + b_{i+1}^{r-1}(t)$$

kjer je

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

Velja $b_i^0(t) := b_i$. Kar pomeni, da če hočemo izračunati točke $b_i^1(t)$, jih bomo dobili s pomočjo točk $b_i^0(t)$, ki so ravno kontrolne točke. Točko na Bézierovi krivulji b_n pri danem parametru t , t.j. $b_0^n(t)$, bomo dobili tako, da bomo izračunali vse vmesne točke pri istem parametru, dokler ne bomo izračunali še zaželenih točk.

1.3. Bernsteinovi polinomi

V numerični analizi je Bernsteinov polinom, poimenovan po Ruskemu matematiku Sergeiju Natanoviču Bernsteinu. Polinom v Bernsteinovi osnovi je linearna kombinacija Bernsteinovih baznih polinomov.

Definicija:

$$B_{i,n}(t) = (x + a)^n = \binom{n}{i} t^i (1 - t)^{n-i}, i = 0, 1, \dots, n.$$

Linearno kombinacijo Bernsteinovih baznih polinomov $\sum_{i=0}^n \alpha_i b_{n,i}$ imenujemo Bernsteinov polinom stopnje n .

Nekaj prvih polinomov:

$$B_{0,0}(t) = 1$$

$$B_{0,1}(t) = 1 - t$$

$$B_{1,1}(t) = t$$

$$B_{0,2}(t) = (1 - t)^2$$

$$B_{1,2}(t) = 2(1 - t)t$$

$$B_{2,2}(t) = t^2$$

1.4. Bernsteinova oblika Bézierove krivulje

Bézierove krivulje lahko definiramo tudi na drugačen način, brez uporabe de Casteljauovega algoritma. Tako definirano Bézierovo krivuljo je opisal A.R. Forrester, ko se je začel zanimati za Bezierovo delo. De Casteljau je že pred njim prišel do istega rezultata, vendar svojih rezultatov ni objavil.

Če parameter t omejimo na interval $[0, 1]$ lahko Bézierove krivulje v Bernsteinovi obliki zapišemo kot:

$$b^n(t) = \sum_{i=0}^n b_i B_i^n(t), t \in [0, 1]$$

1.5. Nekaj pomembnejših lastnosti Bézierove krivulje

Pogledali si bomo 3 lastnosti Bezeirovih krivulj brez dokazov:

1. Bézierova krivulja leži v konveksni ovojnici svojega kontrolnega poligona.
2. Interval parametra t lahko poljubno spremenimo.

3. Bézierove krivulje so invariantne za afine preslikave.

Naj bo φ afina preslikava (razteg, rotacija, premik, ...). Imamo dve možnosti za izračun afino preslikane točke:

- izračunamo točko na Bézierovi krivulji b_n in jo afino preslikamo: $\varphi(b_n)$;
- afino preslikamo kontrolni poligon, torej izračunamo $\varphi(b_0), \varphi(b_1), \dots, \varphi(b_n)$, potem pa izračunamo točko na tako dobljeni Bézierovi krivulji.

2. Projiciranje Bézierovih krivulj v prostoru na ravnino, $z = 0$ v smeri vektorja

Glavni cilj projektne naloge je ugotoviti ali se krivulja, ki jo najprej generiramo in potem projiciramo na ravnino $z=0$, razlikuje od tiste kjer najprej projiciramo kontrolne točke na ravnino v smeri vektorja in jo nato generiramo. Preden se osredotočimo na sam problem bom predstavil metode s katerimi projiciramo točke na krivuljo. Tako bomo tudi numerično preverili ali sta si prej omenjeni krivulji enaki. Vsako točko na krivulji bomo posebej projicirali in nato gledali razlike posameznih točk.

2.1 Projiciranje točke na ravnino $z=0$ v smeri vektorja

Projekcija točke $T(x, y, z) \in \mathbb{R}^3$ na ravnino $z = 0$ v smeri danega vektorja $\vec{v} = [v_x \ v_y \ v_z] \in \mathbb{R}^3$.

Da si projekcijo nekoliko poenostavimo, bomo problem omejili z naslednjimi predpostavkami:

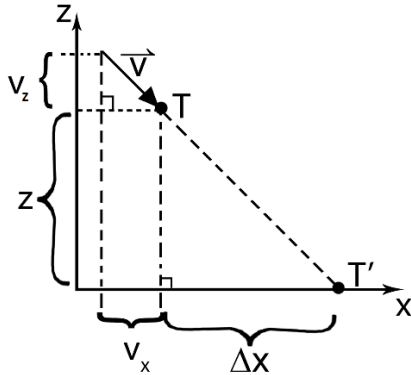
- Točka T leži nad ravnino, $z > 0$
- Vektor \vec{v} kaže navzdol, $v_z < 0$

To lahko naredimo zato, ker v primeru, da bi točka ležala pod ravnino in vektor kazal "na vzdol" projekcija ne bi obstajala (enako velja, če bi vektor kazal navzgor in bi točka ležala nad ravnino). V nasprotnih, ugodnih primerih, ko velja naslednje:

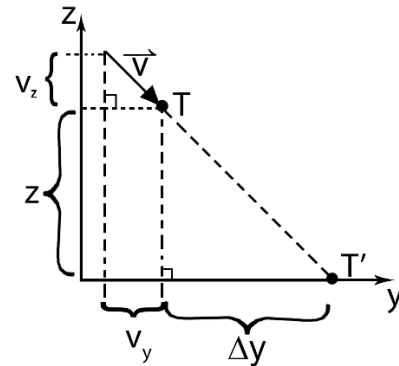
$$z < 0 \text{ in } v_z > 0,$$

lahko problem z raznimi afinimi transformacijami (rotacijami, zrcaljenji, ...) pretvorimo na zgornjega, točko projiciramo in pretvorimo nazaj. Denimo, da bi točko T projicirali na ravnino v

smeri vektorja \vec{v} v točko T' . Oglejmo si projekcijo iz smeri y-osi ter x-osi:



Slika 1: Projekcija točke iz smeri y-osi



Slika 2: Projekcija točke iz smeri x-osi

Na Sliki 1 iz lastnosti podobnih trikotnikov velja enakost:

$$\frac{|v_z|}{|v_x|} = \frac{|z|}{|\Delta x|}$$

od tod sledi

$$|\Delta x| = |z| \frac{|v_x|}{|v_z|}$$

Ker vemo, da je $z > 0$ in $v_z < 0$ se lahko znebimo absolutnih vrednosti tako, da pomnožimo desno stran $z - 1$, prav tako s tem tudi upoštevamo smer v_x (negativno ali pozitivno) in dobimo:

$$\Delta x = z \cdot \frac{v_x}{v_z} \cdot (-1)$$

Enako velja tudi za spremembo y :

$$\Delta y = z \cdot \frac{v_y}{v_z} \cdot (-1)$$

In za spremembo z :

$$\Delta z = z \cdot \frac{v_z}{v_z} \cdot (-1) = -z$$

Projicirano točko T' dobimo tako, da posameznim komponentam prištejemo spremembe:

$$x' = x + \Delta x$$

$$y' = y + \Delta y$$

$$z' = z + \Delta z = z - z = 0$$

Torej:

$$T' = T + (\Delta x, \Delta y, -z)$$

2.2 Posplošitev projekcije na n-dimenzionalen prostor

Definirali bomo tako funkcijo oz. transformacijo $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $n > 1$, $n \in \mathbb{N}$, ki poljubno točko $t \in \mathbb{R}^n$ projicira na ravnino Π z normalo $\vec{n} = [0, 0, \dots, 0, 1] \in \mathbb{R}^n$, ki gre skozi točko $T_n(0, 0, \dots, 0) \in \mathbb{R}^n$ v smeri vektorja $\vec{v} = [v_1 \ v_2 \ \dots \ v_n] \in \mathbb{R}^n$. Točko t bomo zapisali kot vektor $\vec{t} = [t_1 \ t_2 \ \dots \ t_n]$.

Enako kot v prejšnjem sklopu bomo problem omejili z naslednjimi predpostavkami:

- Točka t zapisana kot vektor \vec{t} leži nad ravnino Π , $t_n > 0$
- Vektor \vec{v} kaže navzdol, $v_n < 0$

Ideja je enaka kot v 3-dimenzionalnem prostoru le, da sedaj posplošimo na poljubno dimenzionalen prostor.

Projicirana točka $\vec{t}' = F(\vec{t})$, kjer je F definirana kot:

$$F(\vec{t}) = \vec{t} + (-1) \cdot \frac{\vec{e}_n^T \vec{t}}{v_n} \vec{v}, \quad v_n = \vec{e}_n^T \vec{v},$$

na krajše zapišemo kot

$$F(\vec{t}) = \vec{t} + \overline{\Delta t}, \quad \text{kjer } \overline{\Delta t} = (-1) \cdot \frac{\vec{e}_n^T \vec{t}}{v_n} \vec{v}$$

Poglejmo si transformacijo na primeru v \mathbb{R}^3 , da bomo lažje razumeli. Če bi projicirali točko $t = (1, 1, 1)$ na ravnino Σ kjer je $z = 0$, torej ravnino z normalo $\vec{n} = [0, 0, 1]$, ki gre skozi točko $T(0, 0, 0)$ v smeri vektorja $\vec{v} = [1, 1, -1]$, bi projicirana točka bila enaka:

$$\begin{aligned} \vec{t}' = F([1, 1, 1]) &= [1, 1, 1] + (-1) \cdot \frac{[0, 0, 1]^T \cdot [1, 1, 1]}{[0, 0, 1]^T \cdot [1, 1, -1]} [1, 1, -1] = \\ &= [1, 1, 1] + (-1) \cdot \frac{1}{-1} [1, 1, -1] = \\ &= [1, 1, 1] + [1, 1, -1] = \\ &= [2, 2, 0] \end{aligned}$$

Dobili smo točko $T'(2, 2, 0)$, kar je pravilno. Pravzaprav smo enačbe sprememb iz prejšnjega sklopa posplošili na vektorske enačbe poljubne dimenzije. Seveda pa transformacija velja le pod zgoraj navedenimi pogoji.

Izkaže se, da je taka transformacija F z omejitvami linearna.

Dokaz:

$$\begin{aligned} F(\alpha_1 \vec{t}_1 + \alpha_2 \vec{t}_2 + \dots + \alpha_m \vec{t}_m) &= \\ &= (\alpha_1 \vec{t}_1 + \alpha_2 \vec{t}_2 + \dots + \alpha_m \vec{t}_m) + (-1) \frac{\vec{e}_n^T (\alpha_1 \vec{t}_1 + \alpha_2 \vec{t}_2 + \dots + \alpha_m \vec{t}_m)}{v_n} \cdot \vec{v} \end{aligned}$$

$$\begin{aligned}
 &= \alpha_1 \vec{t}_1 + \alpha_2 \vec{t}_2 + \dots + \alpha_m \vec{t}_m + (-1) \frac{\vec{e}_n^T \alpha_1 \vec{t}_1 + \vec{e}_n^T \alpha_2 \vec{t}_2 + \dots + \vec{e}_n^T \alpha_m \vec{t}_m}{v_n} \cdot \vec{v} \\
 &= \alpha_1 \vec{t}_1 + (-1) \frac{\alpha_1 \vec{e}_n^T \vec{t}_1}{v_n} \cdot \vec{v} + \alpha_2 \vec{t}_2 + (-1) \frac{\alpha_2 \vec{e}_n^T \vec{t}_2}{v_n} \cdot \vec{v} + \dots + \alpha_m \vec{t}_m + (-1) \frac{\alpha_m \vec{e}_n^T \vec{t}_m}{v_n} \cdot \vec{v} \\
 &= \alpha_1 (\vec{t}_1 + \vec{\Delta t}_1) + \alpha_2 (\vec{t}_2 + \vec{\Delta t}_2) + \dots + \alpha_m (\vec{t}_m + \vec{\Delta t}_m) \\
 &= \alpha_1 F(\vec{t}_1) + \alpha_2 F(\vec{t}_2) + \dots + \alpha_m F(\vec{t}_m)
 \end{aligned}$$

□

Ker je F linearna je tudi afina.

2.3 Numerična analiza na primeru z uporabo Matlaba

Glavno vprašanje projektnega dela je, če je Bézierova krivulja v prostoru projicirana na ravnino $z=0$ enaka Bézierovi krivulji, kjer najprej projiciramo kontrolne točke na ravnino in nato ustvarimo ravninsko Bézierjevo krivuljo.

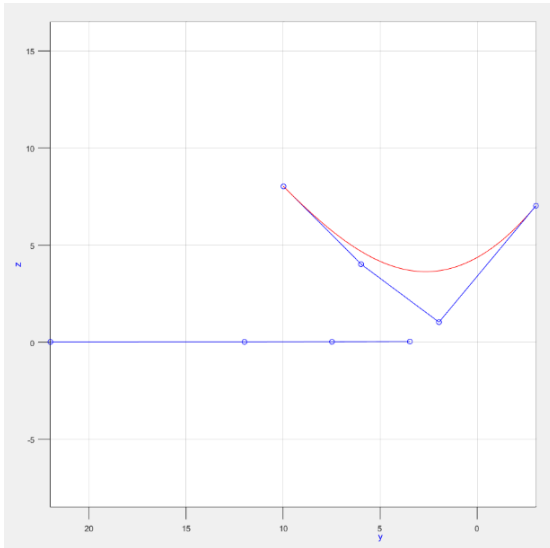
Kot smo povedali je Bézierova krivulja v Bernsteinovi obliki afino invariantna, kar na kratko pomeni, da ob raznih afinih transformacijah »ohranja obliko«. Ker smo pokazali, da je F linearna in s tem tudi afina, je projicirana Bézierjeva krivulja na ravnino enaka tisti, kjer najprej projiciramo kontrolne točke. To bi pokazali še numerično oziroma z uporabo programskega jezika Matlab.

Pri računanju sem uporabil dve funkciji, ki jih je napisal Prof. dr. Emil Žagar, ki izračunajo točke na Bézierovi krivulji glede na podane kontrolne točke z uporabo Bernsteinovih polinomov in še dve funkciji za risanje le teh in transformacijo F , ki posamezno točko projicira na ravnino $z=0$ v smeri danega vektorja.

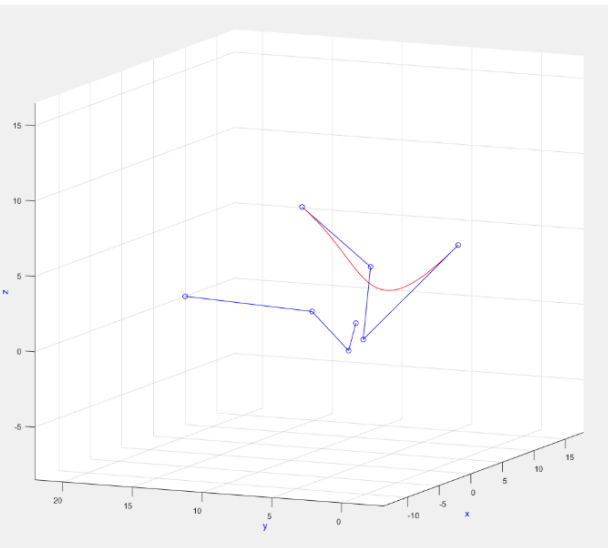
Primer: Določimo vektor $\vec{v} = [2 \ 3 \ -2]$ in kontrolne točke zapisane v matriki, kjer stolpec predstavlja posamezno točko:

$$t = \begin{bmatrix} 2 & 4 & -6 & -2 \\ 10 & 6 & 2 & -3 \\ 8 & 4 & 1 & 7 \end{bmatrix}$$

Če narišemo Bézierovo krivuljo v prostoru. Z modro bravo narišemo kontrolne točke in kontrolne točke projicirane na ravnino $z = 0$ v smeri vektorja v , z rdečo bravo pa Bézierovo krivuljo:

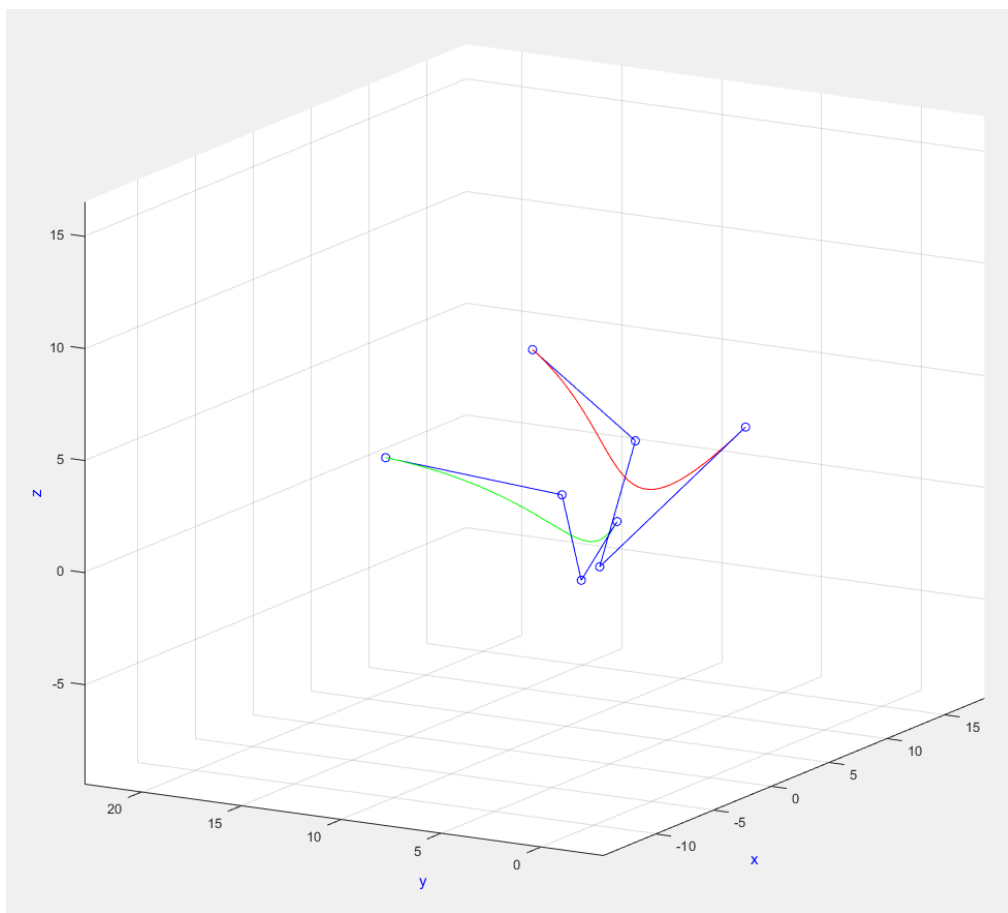


Slika 3: Bézierova krivulja pogled 1



Slika 4: Bézierova krivulja pogled 2

Če sedaj narisan krivuljo projiciramo na ravnino(označimo z Pr_B , pobarvano rdeče) hkrati pa iz kontrolnih točk na ravnini ustvarimo novo Bezierovo krivuljo(označimo z Pr_K , pobarvano zeleno):



Slika 5: Bézierova krivulja pred in po projekciji na ravnino v smeri vektorja

Na ravnini $z = 0$ opazimo le eno zeleno obarvano krivuljo, to pa zato ker je bila zelena krivulja narisana zadnja in sta si Pr_B in Pr_K tako podobni, da se rdeče krivulje ne vidi. To preverimo še numerično tako, da izračunamo 2. normo njunih razlik:

$$\|Pr_B - Pr_K\|_2 = 1.326512769794804 \cdot 10^{-14}$$

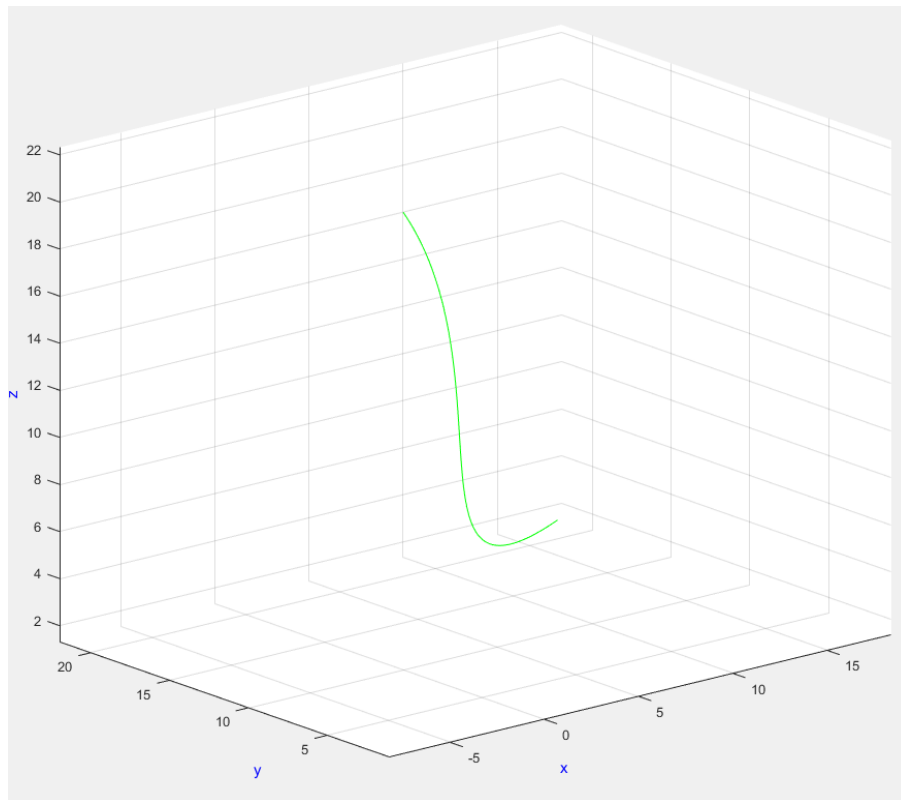
Razlika krivulji je toliko majhna, da lahko trdimo, da sta si krivulji enaki saj je pri izračunu verjetno prišlo do zaokroževanja in s tem posledično, do te majhne razlike.

Ker smo F posplošili na poljubno dimenzijo, bi kot zanimivost lahko poskusili projicirati 4-dimenzionalni krivulji v 3-dimenzionalen prostor. Vzamemo podobne kontrolne točke in vektor kot pri prejšnjem primeru le, da dodamo novi stolpec. oz vrstico:

$$\vec{v} = [2 \ 3 \ 2 \ -3]$$

$$t = \begin{bmatrix} 2 & 4 & -6 & -2 \\ 10 & 6 & 2 & -3 \\ 8 & 4 & 1 & 7 \\ 12 & 10 & 5 & 4 \end{bmatrix}$$

Sedaj iz kontrolnih točk naredimo Bézierovo krivuljo, ter jo projiciramo v smeri vektorja \vec{v} na ravnino z normalo $\vec{n} = [0 \ 0 \ 0 \ 1]$ in jo pobarvamo rdeče. Enako ponovimo le, da najprej projiciramo kontrolne točke ter nato naredimo Bézierovo krivuljo, ki jo pobarvamo zeleno. Ter dobimo:



Slika 6: Projicirani Bézierovi krivulji prikazani v 3-dimenzionalnem prostoru

Ponovno se vidi le ena krivulja, ker sta si krivulji enaki. Norma njunih razlik je tudi v tem primeru zelo majhna in sicer:

$$1.433052172060520 \cdot 10^{14}.$$

3. Projiciranje na poljubno ravnino v \mathbb{R}^3

Do sedaj je vse temeljilo na projekciji na ravnino $z = 0$, to bi radi posplošili na poljubno ravnino v prostoru.

Omejili se bomo na ravnine, ki potekajo skozi izhodišče $(0, 0, 0)$ saj lahko sisteme linearno premaknemo tako, da je ravnina v izhodišču. Problem bomo prevedli na prejšnjega tako, da bomo vse točke, vektorje in ravnino zarotirali za določen vektor.

Točko T , ki jo pišemo kot vektor \vec{t} projiciramo na neko ravnino Σ z normalo \vec{n} tako, da ustvarimo rotacijsko matriko R_0 , ki $\frac{\vec{n}}{\|\vec{n}\|}$ zarotira v enotski vektor $\vec{e}_z = [0 \ 0 \ 1]$ ter rotacijsko matriko R_1 , ki \vec{e}_z zarotira v vektor $\frac{\vec{n}}{\|\vec{n}\|}$.

Točko nato projiciramo v smeri vektorja \vec{v} tako, da točko, vektor in ravnino pomnožimo z R_0 iz leve:

$$\vec{t}_r = R_0 \vec{t}$$

$$\vec{v}_r = R_0 \vec{v}$$

Samo projekcijo točke nato opravimo z uporabo zgoraj prikazane funkcije F v smeri vektorja \vec{v}_r saj velja:

$$\vec{n}_r = R_0 \vec{n} = \vec{e}_z,$$

kar je ravno ravnina $z = 0$. Dobimo projicirano točko:

$$\vec{t}'_r = F(\vec{t}_r),$$

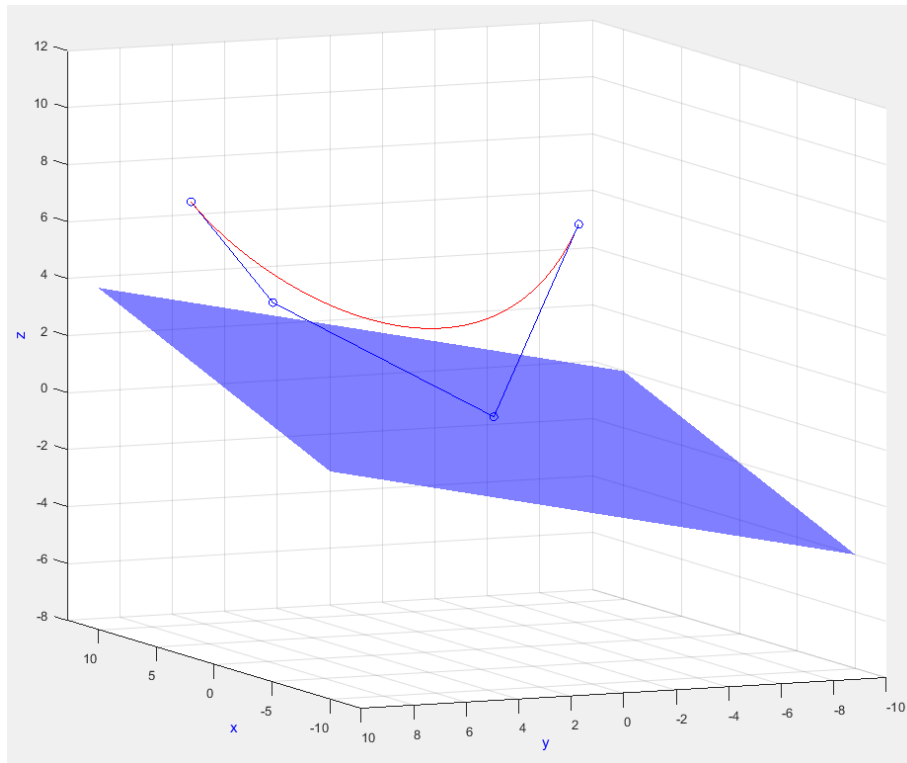
ki jo nato pomnožimo z rotacijsko matriko R_1 , da dobimo točko, ki leži na ravnini Σ :

$$\vec{t}_p = R_1 \vec{t}'_r$$

Preizkusimo to še numerično. Rotacijske matrike ustvarimo z uporabo funkcije `vecRotMat(f, t)` (Koblick, 2021) v programskem jeziku Matlab. Za primer bomo vzeli enake podatke, kakor za primer iz prejšnjega poglavja:

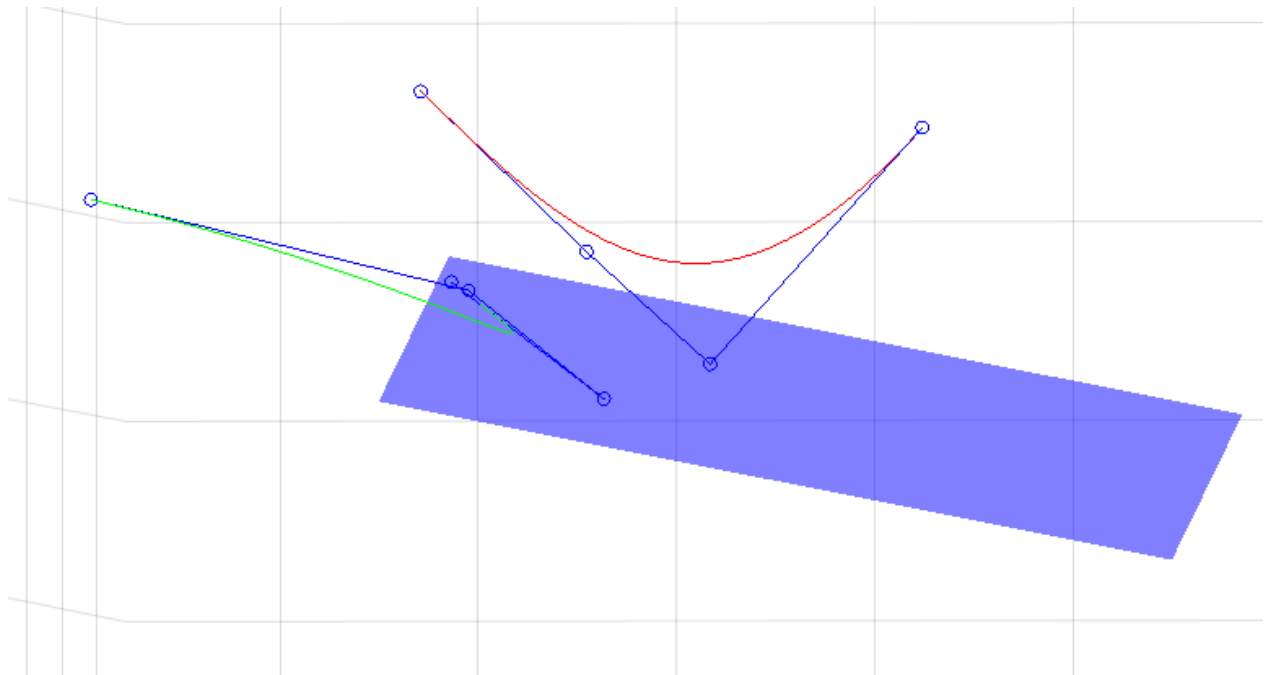
- Kontrolne točke $t = \begin{bmatrix} 2 & 4 & -6 & -2 \\ 10 & 6 & 2 & -3 \\ 8 & 4 & 1 & 7 \end{bmatrix}$
- Vektor $\vec{v} = [2 \ 3 \ -2]$
- Ravnino z normalo $\vec{n} = [-1 \ -1 \ 5]$

Najprej narišemo ravnino, kontrolne točke ter pripadajočo Bézierovo krivuljo:



Slika 7: Kontrolne točke in pripadajoča Bézierova krivulja nad ravnino

Če sedaj izrišemo še projicirano krivuljo na ravnino v smeri vektorja in krivuljo, kjer smo najprej projicirali kontrolne točke:



Slika 8: Projicirani krivulji na omenjeno ravnino

Ponovno opazimo, da sta si projicirani krivulji zelo podobni saj je vidna le zeleno pobarvana. Če izračunamo 2. normo njunih razlik dobimo rezultat:

$$1.712687626561988 \cdot 10^{-14},$$

kar je zanemarljivo malo in krivulji smatramo kot enaki.

4. Zaključek

Soočili smo se z enim od načinov projiciranja točk na ravnino in pokazali posplošitev na vektorsko in matrično računanje z omejitvami. S tem smo pokazali, da je Bézierova krivulja, ki jo projiciramo na ravnino enaka tisti kjer najprej projiciramo kontrolne točke. Afina invariantnost je zelo lepa lastnost saj, če želimo krivuljo raztegovati, rotirati ali pa opraviti katerokoli afino transformacijo si lahko prihranimo računske operacije tako, da transformacije opravljamo na kontrolnih točkah in kasneje generiramo krivuljo.

V priloženih datotekah je še nekaj funkcij, ki niso bile uporabljene pri projektu, s katerimi lahko rišemo poljubne krivulje, točke in projiciramo iz smeri izven določene točke.

5. Viri

TUTA, Marko, 2012, Bézierove krivulje in ploskve : od teoretičnega ozadja do računalniških aplikacij [na spletu]. Univerza na Primorskem, Fakulteta za matematiko, naravoslovje in informacijske tehnologije. [Dostopano 21 februar 2021]. Pridobljeno od:

http://www.famnit.upr.si/sl/izobrazevanje/zakljucna_dela/view/61

Darin Koblick (2021). Efficient Vector Rotation Matrix Routine [Dostopano 21 februar 2021]. Pridobljeno od: <https://www.mathworks.com/matlabcentral/fileexchange/37597-efficient-vector-rotation-matrix-routine>