

LEČ PRÁVĚ NAŠE STOLETÍ,  
KTERÉ BEZ SLOVA VĚDA ANI ZÍVNOUT NEUMÍ,  
MĚ PŘESVĚDČILO,  
ŽE Z NIČEHO SE NEDÁ A NEMÁ DĚLAT VĚDA.  
ANI Z VĚDY.

JAN WERICH

### **Poděkování:**

Ráda bych poděkovala svému školiteli doc. PaedDr. Daliboru Martiškovi, Ph.D. za vedení, podporu a precizní kontrolu mojí práce a doc. RNDr. Josefu Janyškovi, CSc. za možnosti konzultace. V neposlední řadě také svojí rodině, bez jejichž podpory bych už asi  $10^6$ -krát skončila . . .



# Obsah

<b>Obsah</b>	<b>1</b>
<b>Úvod</b>	<b>3</b>
<b>1 Teoretická východiska</b>	<b>5</b>
1.1 Základní definice . . . . .	5
1.1.1 Vektorové prostory . . . . .	5
1.1.2 Afinní a projektivní prostory . . . . .	7
1.1.3 Zobrazení, invariantnost . . . . .	8
1.1.4 Tenzorový počet . . . . .	10
1.2 Pojem křivky a plochy . . . . .	15
1.2.1 Křivky . . . . .	15
1.2.2 Plochy . . . . .	16
<b>2 Současný stav problematiky</b>	<b>19</b>
2.1 Předchůdci B-spline . . . . .	20
2.1.1 Fergusonovy křivky a plochy . . . . .	20
2.1.2 Bézierovy křivky a plochy . . . . .	20
2.1.3 Coonsovy křivky a plochy . . . . .	22
2.2 B-spline, NURBS křivky . . . . .	24
2.2.1 B-spline funkce . . . . .	24
2.2.2 B-spline křivky . . . . .	34
2.2.3 NURBS křivky . . . . .	45
2.3 NURBS plochy . . . . .	55
2.3.1 Definice NURBS plochy . . . . .	55
2.3.2 Vlastnosti NURBS ploch . . . . .	55
2.4 T-spline . . . . .	57
2.4.1 Polární forma – blossoms . . . . .	58
2.4.2 PB-splines . . . . .	60
2.4.3 T-splines . . . . .	61
<b>3 Výsledky práce</b>	<b>69</b>
3.1 Derivace NURBS křivek . . . . .	69
3.1.1 Věta o derivaci B-spline funkce . . . . .	70
3.1.2 Programování derivací NURBS křivek . . . . .	72

3.2	Vliv uzlového vektoru na NURBS plochy . . . . .	77
3.2.1	Ekvidistantní uzlový vektor (0, 1, 2, 3, 4, 5, 6, 7) . . . . .	78
3.2.2	Neekvidistantní uzlový vektor . . . . .	80
3.2.3	Ovládání NURBS ploch . . . . .	86
3.3	NURBS plochy jako tenzorový součin . . . . .	92
3.3.1	Tenzorový součin mezi Grassmannovými prostory . . . . .	93
3.3.2	Odvození tenzorového součinu pomocí vektorových prostorů a přidružených lineárních forem . . . . .	95
3.3.3	Kvazitenzorový součin . . . . .	98
3.4	Programové zpracování NURBS objektů . . . . .	106
3.4.1	NURBS křivky . . . . .	106
3.4.2	Obecná NURBS plocha . . . . .	107
3.4.3	Implementace NURBS těles . . . . .	111
3.5	Spojitě navazování NURBS ploch . . . . .	133
3.5.1	Podmínky hladkého napojení NURBS ploch . . . . .	136
3.6	Využití interpolace pomocí NURBS křivek . . . . .	136
3.6.1	Obecný postup . . . . .	136
3.6.2	Matematický aparát . . . . .	137
3.6.3	Interpolace na válci . . . . .	139
3.7	Nové metody modelování ploch použité v GIS . . . . .	140
3.7.1	Detekce objektů - střechy budov, objekty . . . . .	142
3.7.2	Uspořádání mraku bodů . . . . .	142
3.7.3	Výpočet uzlových vektorů . . . . .	142
3.7.4	Zjednodušení plochy metodou T-spline simplification . . . . .	143
3.7.5	Výpočet výsledné T-spline plochy a zobrazení . . . . .	144
3.8	NURBS programy . . . . .	147
3.8.1	Open Source knihovny . . . . .	148
3.8.2	Form•Z . . . . .	149
3.8.3	Blender . . . . .	150
3.8.4	Autodesk Maya – learning edition . . . . .	151
3.8.5	RFEM 3D . . . . .	153

<b>4 Závěr</b>	<b>157</b>
----------------	------------

<b>Literatura</b>	<b>162</b>
-------------------	------------

<b>Použité značení</b>	<b>163</b>
------------------------	------------



# Úvod

Vývoj novodobých matematických křivek a ploch započal v 60. letech minulého století. S rozvojem nových průmyslových odvětví vznikla potřeba speciálních druhů křivek a ploch, které by sloužily designérům k návrhu hladkých obecných tvarů (automobilový průmysl, strojírenství). Vedle křivek vyjádřených explicitními či parametrickými rovnicemi vznikají křivky a plochy určené řídicími body a dalšími parametry. Mezi jejich průkopníky patří jména francouzských inženýrů - Carl de Boor, James C. Ferguson, Pierre Bézier, Paul F. de Casteljau. Tak vznikl obor, který nese název CAD – Computer Aided Design.

Současný vývoj aplikací dospěl až k NURBS objektům, bez nichž se v dnešní době neobejde většina grafických softwarů (Maya, Rhinoceros, Blender, atd.). Otázkou je, proč používat NURBS? Výhod existuje celá řada. Patří mezi ně lokální kontrolovatelnost (změna jednoho parametru ovlivní objekt pouze lokálně), rychlý a stabilní algoritmus výpočtu, možnost kreslení kuželoseček, zachovaná spojitost při změnách a samozřejmě takřka neomezené konstrukční možnosti.

Z těchto důvodů je NURBS křivkám a plochám v oblasti Computer science věnována velká pozornost. Vývoj samozřejmě pokračuje dál a NURBS objekty jsou zlepšovány. Novým trendem se staly T-splines. Jedná se o zobecnění NURBS plochy, jejíž body nemusí ležet v pravidelné obdélníkové mřížce. V roce 2005 byly implementovány do softwaru Maya a v roce 2006 do softwaru Rhinoceros.

Při vývoji těchto objektů je však často kladen důraz jen na technickou stránku věci, aniž by se věnovala pozornost matematickým souvislostem. Cílem této práce je tedy přispět k vývoji těchto objektů, a to jak po stránce matematické, tak po stránce implementační.

První kapitola práce je věnována teoretickým východiskům, která jsou nutná k pochopení dalšího textu. obsahuje základy vektorových prostorů, afinní a projektivní geometrie. Druhá kapitola je věnována obecné teorii křivek a ploch z pohledu diferenciální geometrie. Třetí kapitola obsahuje vlastní výsledky práce, které lze shrnout do těchto bodů:

1. **Analytické vyjádření derivace NURBS křivek.** Derivace NURBS křivek hraje důležitou roli při výpočtech, např. statiky objektů. Práce dává nový důkaz pro vzorec výpočtu, jeho srovnání se současně používanou numerickou derivací a také způsob vlastní implementace. Navržený způsob výpočtu derivace byl v r. 2006 implementován do komerčního softwaru RFEM 3D.
2. **Modelace vlivu uzlového vektoru na tvar NURBS.** NURBS objekty patří mezi nové křivky a plochy, které se zadávají řídicími body a dalšími parametry.

Těmi jsou uzlové vektory a váhy jednotlivých bodů. Většina softwaru nabízí možnost práce s uzlovými vektory, ta však není intuitivní. V dostupné literatuře neexistuje přehledný geometrický pohled na vliv uzlového vektoru na tvar křivek a ploch.

3. **Využití interpolace pomocí NURBS křivek v lékařství.** V rámci spolupráce s ústavem strojírenské technologie naší fakulty a nemocnicí u sv. Anny v Brně vyvíjíme program založený na interpolaci pomocí NURBS, který bude v budoucnu sloužit při výrobě kolenních kloubních náhrad. Výzkum je v současné době v testovací fázi.
4. **Matematický popis jednoduchých NURBS těles z netradičního tenzorového pohledu.** Pro usnadnění práce s NURBS plochami jsou ve většině kvalitních aplikací předdefinována základní NURBS tělesa – koule, kužel, válec, hranol, obecná rovina a anuloid. Lze je elegantně popsat pomocí tenzorového součinu. Tento přístup je v literatuře silně opomíjen.
5. **Matematická formulace nových trendů a jejich uplatnění v geografických informačních systémech.** Novým trendem se staly T-spline a T-NURCCs plochy, které obsahují uzly T-junctions. Jedná se o zobecnění NURBS ploch, jejichž řídicí body nemusí ležet v pravidelné obdélníkové mřížce. Tyto objekty postupně nacházejí široké použití v grafických systémech, mohou však být použity i v jiných oblastech. V práci je demonstrováno jejich možné použití v geografických informačních systémech při vykreslování ploch ze zadaných dat.
6. **Návrh efektivních způsobů implementace NURBS křivek, ploch a základních těles s ohledem na jejich stabilitu a rychlost.**
7. **Ověření navržených metod implementací do německého komerčního CAD softwaru RFEM 3D.** V rámci doktorského studia jsem spolupracovala s firmou Fem Consulting, do jejichž komerčního softwaru RFEM 3D implementuji svoje teoretické poznatky a testuji výsledky. Systém RFEM 3D slouží k navrhování stavebních konstrukcí či strojních součástí, na které je následně aplikována metoda konečných prvků pro výpočet statiky a dalších důležitých konstrukčních vlastností.
8. **Srovnání implementace a stability NURBS v grafických programech, kde jsou již implementovány, s mými metodami implementovanými v programu RFEM 3D.**

# Kapitola 1

## Teoretická východiska

### 1.1 Základní definice

V následující kapitole jsou uvedeny základní definice struktur a vlastností, které se v textu vyskytují. Více podrobností o vektorových a afinních prostorech lze najít v Horák (1991), Martišek (2002), Sekanina (1988). Projektivní vektorový prostor a projektivní invariantnost je podrobně popsána v Janyška – Sekaninová (2001). Tenzorovým součinem a tenzory se zabývá například Motl – Zahradník (Naposledy navštíveno 10. 3. 2007), Čadek (Naposledy navštíveno 12. 4. 2007).

#### 1.1.1 Vektorové prostory

*Definice 1.1.* Nechť  $(V, \oplus)$  je komutativní grupa, jejíž prvky nazýváme vektory a  $(T, +, \cdot)$  je číselné těleso. Nechť pro každé číslo  $t \in T$  a každý vektor  $\mathbf{u} \in V$  je definován vektor  $t \odot \mathbf{u} \in V$  tak, že platí:

1.  $t \odot (\mathbf{u} \oplus \mathbf{v}) = t \odot \mathbf{u} \oplus t \odot \mathbf{v}$
2.  $(t + s) \odot \mathbf{u} = t \odot \mathbf{u} \oplus s \odot \mathbf{u}$
3.  $t \odot (s \odot \mathbf{u}) = (t \cdot s) \odot \mathbf{u}$
4.  $1 \odot \mathbf{u} = \mathbf{u}$

kde  $t, s \in T$  a  $\mathbf{u}, \mathbf{v} \in V$ . Potom  $V$  se nazývá **vektorový** nebo **lineární prostor nad tělesem  $T$** .

*Poznámka 1.* V dalším textu budeme používat znaménko  $+$  místo  $\oplus$  a znaménko  $\cdot$  místo  $\odot$ . Je nutné si uvědomit, že se obecně jedná o různé operace.

*Definice 1.2.* Nechť  $V$  je vektorový prostor nad tělesem  $T$ . Neprázdna podmnožina  $W$  množiny  $V$  se nazývá **podprostor vektorového prostoru  $V$** , jestliže platí:

1.  $\mathbf{u}, \mathbf{v} \in W$  libovolné  $\Rightarrow \mathbf{u} + \mathbf{v} \in W$
2.  $t \in T, \mathbf{u} \in W$  libovolné  $\Rightarrow t \cdot \mathbf{u} \in W$ .

Nechť  $M$  je libovolná podmnožina vektorového prostoru  $V$ . Pak existuje alespoň jeden podprostor, obsahující množinu  $M$  (např. celý prostor  $V$ ). Můžeme utvořit průnik všech podprostorů ve  $V$ , které obsahují množinu  $M$ , který označíme  $\langle M \rangle$ .

**Věta 1.1.** *Nechť  $M$  je libovolná podmnožina ve vektorovém prostoru  $V$ . Potom*

1.  $\langle M \rangle$  je podprostor,
2.  $\langle M \rangle$  je nejmenší (vzhledem k  $\subseteq$ ) podprostor ve  $V$ , obsahující množinu  $M$ .

Důkaz: Viz Horák (1991).

**Definice 1.3.** Nechť  $M$  je podmnožina ve vektorovém prostoru  $V$  a nechť  $W = [M]$ . Pak podprostor  $W$  se nazývá **podprostor generovaný množinou  $M$** .

Je-li speciálně  $M = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$ , pak  $W$  se nazývá **podprostor generovaný vektory  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$**  a vektory  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$  se nazývají **generátory podprostoru  $W$** .

**Definice 1.4.** Nechť  $V$  je vektorový prostor nad tělesem  $T$  a nechť  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$  je konečná posloupnost vektorů z  $V$ . Jestliže pro všechna  $t_1, \dots, t_k \in T$  platí, že

$$t_1 \cdot \mathbf{u}_1 + \dots + t_k \cdot \mathbf{u}_k = \mathbf{0} \Rightarrow t_1 = t_2 = \dots = t_k = 0,$$

pak říkáme, že vektory  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$  jsou **lineárně nezávislé**.

**Definice 1.5.** Nechť  $V$  je vektorový prostor nad tělesem  $T$ . Potom jakýkoliv vektor  $\mathbf{w}$  tvaru součtu násobků konečného počtu vektorů  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$  tohoto prostoru nazveme **lineární kombinací vektorů  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$** . Píšeme:

$$\mathbf{w} = \sum_{i=1}^n t_i \mathbf{u}_i$$

**Definice 1.6.** Nechť  $V$  je vektorový prostor nad  $T$ . Konečná posloupnost  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$  vektorů z  $V$  se nazývá **báze vektorového prostoru  $V$** , jestliže platí:

1. Vektory  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$  jsou lineárně nezávislé.
2. Vektory  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$  generují vektorový prostor  $V$ .

Číslo  $n$  se potom nazývá **konečnou dimenzí** vektorového prostoru  $V$ .

**Definice 1.7.** Nechť

$$\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\} \tag{1.1}$$

je báze vektorového prostoru  $V$  a nechť vektor  $\mathbf{w} \in V$  je vyjádřen ve tvaru

$$\mathbf{w} = t_1 \mathbf{u}_1 + \dots + t_n \mathbf{u}_n, \quad \text{kde } t_1, \dots, t_n \in T. \tag{1.2}$$

Pak číslo  $t_i$  nazýváme  $i$ -tou souřadnicí vektoru  $\mathbf{w}$  v bázi (1.1). **Souřadnicemi vektoru v bázi (1.1)** nazýváme uspořádanou  $n$ -tici  $(t_1, \dots, t_n)$   $\mathbf{w}$ .

*Definice 1.8.* Necht  $V$  je vektorový prostor (nad  $\mathbb{R}$ ) a necht každé dvojici vektoru  $\mathbf{u}, \mathbf{v} \in V$  je přiřazeno reálné číslo  $\mathbf{u} \cdot \mathbf{v}$  tak, že pro libovolné  $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V, r \in \mathbb{R}$  platí:

$$\begin{aligned}\mathbf{u} \cdot \mathbf{v} &= \mathbf{v} \cdot \mathbf{u} \\ (\mathbf{u} + \mathbf{v}) \cdot \mathbf{w} &= (\mathbf{u} \cdot \mathbf{w}) + (\mathbf{v} \cdot \mathbf{w}) \\ (r \cdot \mathbf{u}) \cdot \mathbf{v} &= r \cdot (\mathbf{u} \cdot \mathbf{v}) \\ \text{je-li } \mathbf{u} &\neq \mathbf{0}, \text{ pak } \mathbf{u} \cdot \mathbf{u} > 0\end{aligned}$$

Potom reálné číslo  $\mathbf{u} \cdot \mathbf{v}$  se nazývá **skalární součin vektorů  $\mathbf{u}, \mathbf{v}$** .

*Definice 1.9.* Konečnědimenzionální vektorový prostor se skalárním součinem se nazývá euklidovský vektorový prostor nebo krátce **euklidovský prostor**.

*Poznámka 2.* V dalších částech budeme k výpočtům používat klasický skalární součin vektorů  $\mathbf{u}, \mathbf{v}$  daný předpisem:  $\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^n u_i v_i$ . Pro zápis tohoto součinu použijeme  $(\mathbf{u}, \mathbf{v})$ .

*Poznámka 3.* Vektorový prostor se skalárním součinem bývá označován jako unitární metrický prostor. Norma prvku  $\mathbf{u} \in V$  je určena jako  $|\mathbf{u}| = (\mathbf{u}, \mathbf{u})$  a vzdálenost mezi dvěma prvky  $\mathbf{u}, \mathbf{v} \in V$  je rovna  $d(\mathbf{u}, \mathbf{v}) = |\mathbf{v} - \mathbf{u}|$ .

Skalární součin budeme používat pouze v praktické části při geometrických výpočtech (vzájemná poloha bodu a přímky, odchylka přímek). Z tohoto důvodu nebudeme zavádět celou definici normy a omezíme se na jeho praktické použití.

## 1.1.2 Afinní a projektivní prostory

*Definice 1.10.* Mějme neprázdnou množinu  $\mathcal{A}_n$ , pro kterou existuje lineární prostor  $V_n$  dimenze  $n$ ,  $n \in \mathbb{N}$  nad tělesem  $T$  a dále zobrazení  $\varphi : \mathcal{A}_n \times \mathcal{A}_n \rightarrow V_n$  takové, že platí:

- a. pro každé  $A \in \mathcal{A}_n$  a každé  $\mathbf{x} \in V_n$  existuje právě jedno  $B \in \mathcal{A}_n$  takové, že  $\varphi(A; B) = \mathbf{x}$
- b. pro každé  $A, B, C \in \mathcal{A}$  je  $\varphi(A, C) = \varphi(A, B) + \varphi(B, C)$ .

Pak množinu  $\mathcal{A}_n$  nazýváme **afinním prostorem**. Vektorový prostor  $V_n$  nazýváme **zaměřením prostoru  $\mathcal{A}_n$** , dimenzi prostoru  $V_n$  nazýváme **dimenzí  $\mathcal{A}_n$** . Prvky  $A \in \mathcal{A}_n$  nazýváme **body** afinního prostoru. Uspořádanou dvojici  $(A, B)$  nazýváme **umístěním vektoru  $\mathbf{x}$  v prostoru  $\mathcal{A}_n$** . Místo  $\mathbf{x} = \varphi(A, B)$  budeme psát  $\mathbf{x} = \overrightarrow{AB}$ .

*Definice 1.11.* Necht  $O \in \mathcal{A}_n$  je libovolný bod afinního prostoru,  $\langle \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n \rangle$  libovolná báze jeho zaměření. Pak uspořádanou  $(n + 1)$ -tici  $\langle O, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n \rangle$  nazýváme **afinním repérem**. Afinní repér euklidovského prostoru nazýváme **kartézskou souřadnou soustavou**.

**Věta 1.2.** Necht  $\langle O, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n \rangle$  je libovolný afinní repér prostoru  $\mathcal{A}_n$ ,  $A \in \mathcal{A}_n$  libovolný bod. Pak existují  $a_i \in T$ ;  $i = 1, 2, \dots, n$  taková, že  $\overrightarrow{OA} = a_1 \mathbf{e}_1 + \dots + a_n \mathbf{e}_n$ .

Důkaz: Janyška – Sekaninová (2001).

*Definice 1.12.* Čísla  $a_i \in T$  z předchozí věty nazýváme **souřadnicemi bodu  $A \in \mathcal{A}_n$** . Píšeme  $A = [a_1, \dots, a_n]$ .

*Definice 1.13.* Afinní prostor  $\mathcal{B}_m$  je **podprostorem afinního prostoru**  $\mathcal{A}_n$  právě tehdy, když  $\mathcal{B}_m \subseteq \mathcal{A}_n$  a  $Z(\mathcal{B}_m)$  je podprostorem lineárního prostoru  $Z(\mathcal{A}_n)$ . Je-li  $B \in \mathcal{B}_m$  libovolný bod a  $\langle \mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n \rangle$  báze zaměření  $Z(\mathcal{B}_m)$ , píšeme  $\mathcal{B}_m = \langle B, \mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n \rangle$ .

Speciálně pro  $n = 3$ :  $\langle B, \mathbf{u} \rangle$  – přímka;  $\langle B, \mathbf{u}, \mathbf{v} \rangle$  – rovina.

*Definice 1.14.* Nechť  $V_{n+1}$  je vektorový prostor dimenze  $n + 1$  nad tělesem reálných čísel. Množinu  $P_n$  všech jednodimezionálních podprostorů prostoru  $V_{n+1}$  nazýváme **projektivním prostorem** dimenze  $n$  nad tělesem  $R$ . Jeho prvky nazýváme **body**.  $V_{n+1}$  nazýváme aritmetickým základem prostoru  $P_n$ . Vektor  $\mathbf{x} \neq \mathbf{o}$ , který generuje bod  $X = \langle \mathbf{x} \rangle = \{\alpha \mathbf{x}, \alpha \in T\} \in P_n$  nazýváme aritmetickým zástupcem bodu  $X$ .

*Definice 1.15.* **Aritmetickou bází** projektivního prostoru  $P_n$  rozumíme libovolnou bázi  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n+1}$  vektorového prostoru  $V_{n+1}$ . **Geometrickou bází** prostoru  $P_n$  rozumíme  $(n + 2)$ -tici bodů  $\langle O_1, \dots, O_{n+1}, E \rangle$  takových, že libovolných  $(n + 1)$  z nich je lineárně nezávislých. Body  $O_1, \dots, O_{n+1}$  nazýváme **základní body** a bod  $E$  **jednotkový bod** geometrické báze.

*Definice 1.16.* Nechť  $X \in P_n$  je bod,  $\langle O_1, \dots, O_{n+1}, E \rangle$  je geometrická báze  $P_n$  taková, že  $O_1 = \langle \mathbf{u}_1 \rangle, \dots, O_{n+1} = \langle \mathbf{u}_{n+1} \rangle, E = \langle \mathbf{u}_1 + \dots + \mathbf{u}_{n+1} \rangle$ . Nechť  $X = \langle \mathbf{x} \rangle$ , kde

$$\mathbf{x} = x_1 \mathbf{u}_1 + \dots + x_{n+1} \mathbf{u}_{n+1}, \quad x_i \in T.$$

Potom uspořádanou  $(n + 1)$ -tici  $(x_1, \dots, x_n)$  prvků z  $T$  nazveme **projektivními homogenními souřadnicemi** bodu  $X$  vzhledem ke geometrické bázi  $\langle O_1, \dots, O_{n+1}, E \rangle$ .

*Definice 1.17.* Projektivní prostor  $Q_m$  nazýváme **projektivním podprostorem** prostoru  $P_n$ , jestliže aritmetický základ prostoru  $Q_m$  je podprostorem aritmetického základu  $P_n$ .

### 1.1.3 Zobrazení, invariantnost

*Definice 1.18.* Nechť  $A, B$  jsou libovolné neprázdné množiny. Množinu  $f \subseteq A \times B$  nazveme **zobrazením množiny**  $A$  do množiny  $B$  právě tehdy, když:

$$\forall a \in A \quad \exists! b \in B : [a, b] \in f \quad (1.3)$$

Píšeme  $f(A) = B$ , popř.  $f : A \rightarrow B$ ; místo  $[a, b] \in f$  píšeme obvykle  $f(a) = b$ , popř.  $f : a \rightarrow b$ . Množina  $A$  se nazývá **definiční obor zobrazení**  $f$  a množina  $B$  se nazývá **obor hodnot zobrazení**  $f$ .

*Definice 1.19.* Nechť  $f$  je zobrazení z množiny  $A$  do množiny  $B$ . Pak se zobrazení nazývá **surjektivní**, jestliže pro každý prvek  $y \in B$  existuje  $x \in A$  tak, že  $f(x) = y$  neboli množina hodnot zobrazení  $f$  se rovná oboru hodnot zobrazení  $f$ .

Jestliže má každý prvek z  $B$  nejvýše jeden vzor, nazývá se  $f$  **prosté**, neboli **injektivní**. Je-li  $f$  surjektivní i injektivní, nazývá se **bijekce**.

Morfismy prostorů – jsou zobrazení, která přeneseně řečeno zachovávají operace mezi prostory. Morfismy lineárních (afinních, projektivních) prostorů budeme nazývat lineární (afinní, projektivní) zobrazení.

*Definice 1.20.* Nechť  $V, V'$  jsou vektorové prostory nad stejným číselným tělesem. Zobrazení  $\varphi : V \rightarrow V'$  splňující:

1.  $\varphi(\mathbf{u} + \mathbf{v}) = \varphi(\mathbf{u}) + \varphi(\mathbf{v})$
2.  $\varphi(t \cdot \mathbf{u}) = t \cdot \varphi(\mathbf{u})$

se nazývá **lineární zobrazení** vektorového prostoru  $V$  do  $V'$ . Je-li navíc  $\varphi$  bijektivní, pak se nazývá **izomorfismus vektorového prostoru**  $V$  na  $V'$ .

*Definice 1.21.* Necht  $\mathcal{A}, \mathcal{A}'$  jsou afinní prostory,  $\varphi : Z(\mathcal{A}) \rightarrow Z(\mathcal{A}')$  lineární zobrazení jejich zaměření a  $\psi : \mathcal{A} \rightarrow \mathcal{A}'$  zobrazení takové, že

$$\forall A, B \in \mathcal{A} : \overrightarrow{AB} = \mathbf{x} \implies \overrightarrow{\psi(A)\psi(B)} = \varphi(\mathbf{x}), \quad (1.4)$$

pak  $\psi$  je **afinní zobrazení**.

*Definice 1.22.* Necht  $A, B$  jsou dva různé body přímky  $p$ , necht  $C \in p$ ,  $C \neq B$ . Pak číslo

$$\lambda = \frac{|\overrightarrow{AC}|}{|\overrightarrow{BC}|}$$

se nazývá **dělicí poměr** bodu  $C$  vzhledem k bodům  $A, B$ .

**Věta 1.3.** *Afinní zobrazení zachovává dělicí poměr bodů.*

*Definice 1.23.* Necht  $P_n, P_m$  jsou projektivní prostory,  $V_{n+1}, V_{m+1}$  jejich aritmetické základy. Lineární zobrazení  $\varphi : V_{n+1} \rightarrow V_{m+1}$  nazýváme **projektivním zobrazením**.

*Definice 1.24.* Necht  $V$  je vektorový prostor nad tělesem  $T$ . Lineární zobrazení  $\varphi : V \rightarrow V$  se nazývá **lineární transformace** vektorového prostoru  $V$ . Je-li navíc  $\varphi$  bijektivní, pak se nazývá **automorfismus** vektorového prostoru  $V$ .

*Definice 1.25.* Necht  $\mathcal{A}_n$  je afinní (resp. projektivní prostor). Lineární zobrazení odpovídajícího zaměření (resp. aritmetickému základu) se nazývá **afinní** (resp. **projektivní**) **transformace**. Je-li toto zobrazení bijektivní, jedná se o afinní (resp. projektivní) **automorfismus**.

### Afinní a projektivní invariantnost

*Definice 1.26.* Necht  $\tau : \mathcal{S} \rightarrow \mathcal{S}$  je lineární (resp. afinní, resp. projektivní) transformace. Lineární (afinní, projektivní) podprostor  $\mathcal{W}$  prostoru  $\mathcal{S}$  se nazývá **invariantní** vzhledem k  $\tau$  právě tehdy, když  $\tau(\mathcal{W}) \subseteq \mathcal{W}$ .

*Poznámka 4.* B-spline křivky (resp. plochy) jsou afinně invariantní, NURBS jsou projektivně invariantní, což bude vysvětleno v dalším textu. V praxi to znamená, že není nutné transformovat celou křivku či plochu bodově, ale stačí v dané transformaci přepočítat pouze řídicí body.

Obecně řekneme, že objekt je afinně invariantní, pokud je invariantní vzhledem k afinním transformacím. Tj. složení otáčení, posunutí, změna měřítka a zkosení (více v Foley et al. (2005), Martišek (2002)). Afinní transformace zachovávají incidenci a rovnoběžnost, ale nezachovávají délky a úhly.

**Matice**

*Definice 1.27.* Zobrazení

$$\mathbf{A} : \{0, 1, \dots, m\} \times \{0, 1, \dots, n\} \rightarrow A$$

nazýváme **maticí** typu  $(m + 1) \times (n + 1)$  nad množinou  $A$ . Speciálně:

$A = \mathbb{R} \Rightarrow$  reálná matice,

$A = \mathcal{A}_n \Rightarrow$  matice  $n$ -rozměrných bodů,

$A = \mathcal{A}_3 \Rightarrow$  matice trojrozměrných bodů.

Obvykle zapisujeme ve tvaru:

$$\mathbf{A} = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0n} \\ a_{10} & a_{11} & \dots & a_{1n} \\ \vdots & \ddots & & \vdots \\ a_{m0} & a_{m1} & \dots & a_{mn} \end{pmatrix}. \quad (1.5)$$

**1.1.4 Tenzorový počet**

V následující části jsou uvedeny základní pojmy týkající se tenzorového součinu. Tenzorový součin lze definovat několika způsoby. Jeden z nich je založen na lineárním obalu vektorových prostorů, další využívá faktorprostorů a jedna z možných definic je uvedena níže. Podrobné definice i jejich vzájemné propojení jsou uvedeny např. v Motl – Zahradník (Naposledy navštíveno 10. 3. 2007), Čadek (Naposledy navštíveno 12. 4. 2007).

*Definice 1.28.* Nechť  $(V, +, \cdot)$  je nenulový vektorový prostor konečné dimenze  $n$  nad tělesem  $(T, +, \cdot)$ . Lineární zobrazení  $\phi : V \rightarrow T$  se nazývá **lineární forma** nad  $(V, +, \cdot)$  právě tehdy, když pro každé  $\mathbf{u}, \mathbf{v} \in V$  a každé  $c \in T$  platí:

**a.**  $\phi(\mathbf{u} + \mathbf{v}) = \phi(\mathbf{u}) + \phi(\mathbf{v})$

**b.**  $\phi(c\mathbf{u}) = c\phi(\mathbf{u})$

Je nutné si uvědomit, že na levých stranách uvedených rovností je vektorové sčítání, resp. násobení vektoru skalárem, kdežto na pravé straně se jedná o binární operace v tělese  $T$ . Tyto operace jsme podle poznámky 1 (str. 5) přestali z důvodů formálního zjednodušení rozlišovat.

**Věta 1.4.** *Nechť  $V^*$  je množina všech lineárních forem nad  $V$ . Pro každé  $\phi_1, \phi_2 \in V^*$  definujeme:*

**a.**  $\phi_1(\mathbf{u}) \oplus \phi_2(\mathbf{u}) = \phi_1(\mathbf{u}) + \phi_2(\mathbf{u})$

**b.**  $c \odot \phi_1(\mathbf{u}) = c \cdot \phi_1(\mathbf{u})$

*Pak  $(V^*, \oplus, \odot)$  je vektorový prostor nad  $T$ .*

*Definice 1.29.* Vektorový prostor  $V^*$  z předchozí věty nazýváme **duálním prostorem** k prostoru  $V$ .



*Definice 1.30.* Necht'  $U, V$  jsou vektorové prostory konečných dimenzí  $m, n$  nad týmž tělesem  $T$ . Zobrazení  $\omega : U \times V \rightarrow T$  se nazývá **bilineární formou** mezi prostory  $U, V$  právě tehdy, když pro libovolné  $\mathbf{u}_1, \mathbf{u}_2 \in U, \mathbf{v}_1, \mathbf{v}_2 \in V$  a  $a, b \in T$  platí:

$$\begin{aligned}\omega(a\mathbf{u}_1 + b\mathbf{u}_2, \mathbf{v}_1) &= a\omega(\mathbf{u}_1, \mathbf{v}_1) + b\omega(\mathbf{u}_2, \mathbf{v}_1) \\ \omega(\mathbf{u}_1, a\mathbf{v}_1 + b\mathbf{v}_2) &= a\omega(\mathbf{u}_1, \mathbf{v}_1) + b\omega(\mathbf{u}_1, \mathbf{v}_2)\end{aligned}$$

Tyto operace jsou podle poznámky 1 (str. 5) opět na každé straně rovnic jiné.

*Definice 1.31.* Zobrazení  $\omega : U \times V \rightarrow T$  nazýváme **součtem bilineárních forem**  $\omega_1 : U \times V \rightarrow T, \omega_2 : U \times V \rightarrow T$  právě tehdy, když pro každé  $\mathbf{u} \in U, \mathbf{v} \in V$  platí:

$$\omega(\mathbf{u}, \mathbf{v}) = \omega_1(\mathbf{u}, \mathbf{v}) + \omega_2(\mathbf{u}, \mathbf{v}) \quad (1.6)$$

*Definice 1.32.* Zobrazení  $\omega : U \times V \rightarrow T$  nazýváme **skalárním násobkem bilineární formy**  $\omega : U \times V \rightarrow T$  právě tehdy, když existuje  $c \in T$  takové, že pro každé  $\mathbf{u} \in U, \mathbf{v} \in V$  platí:

$$\omega(c\mathbf{u}, c\mathbf{v}) = c\omega(\mathbf{u}, \mathbf{v}) \quad (1.7)$$

**Věta 1.5.** Množina  $U \otimes V$  všech bilineárních forem mezi prostory  $U, V$  spolu se součtem z definice 1.31 a skalárním násobkem z definice 1.32 je vektorový prostor nad  $T$ .

*Důkaz:* Lze provést ověřením podmínek z definice vektorového prostoru.

*Definice 1.33.* Vektorový prostor  $U \otimes V$  daný větou 1.5 nazýváme **tenzorovým součinem** prostorů  $U, V$ . Speciálně tenzorové součiny  $U \otimes U, U \otimes U^*, U^* \otimes U^*$  nazýváme prostory tenzorů, jejich prvky nazýváme **tenzory**, přičemž prvky prostoru  $U$  nazýváme kontravariantní vektory, značíme  $\mathbf{u} = (u_1, u_2, \dots, u_m)$ . Prvky duálního prostoru  $U^*$  nazýváme kovariantní vektory, značíme  $\mathbf{u}^* = (u^1, u^2, \dots, u^m)$ . Podrobněji pak nazýváme:

$$\begin{aligned}U \otimes U &\quad \text{prostor kontravariantních tenzorů} \\ \mathbf{u} \otimes \mathbf{v} &= (u_1, u_2, \dots, u_m) \otimes (v_1, v_2, \dots, v_m) \\ U \otimes U^* &\quad \text{prostor smíšených tenzorů} \\ \mathbf{u} \otimes \mathbf{v}^* &= (u_1, u_2, \dots, u_m) \otimes (v^1, v^2, \dots, v^m) \\ U^* \otimes U^* &\quad \text{prostor kovariantních tenzorů} \\ \mathbf{u}^* \otimes \mathbf{v}^* &= (u^1, u^2, \dots, u^m) \otimes (v^1, v^2, \dots, v^m)\end{aligned}$$

*Poznámka 5.* Vzhledem k tomu, že tenzorový součin lineárních prostorů je opět lineární prostor, lze celou konstrukci opakovat a sestrojovat tenzorové součiny např.  $(U \otimes U) \otimes U, (U \otimes U) \otimes U^*, (U \otimes U^*) \otimes U^*$ , atd. Rovněž tyto prostory nazýváme prostory tenzorů a jejich prvky tenzory. Lze pak hovořit například o tenzorech dvakrát kontravariantních a jednou kovariantních (což jsou prvky prostoru  $(U \otimes U) \otimes U^*$ ) apod.

Takto opakované konstrukce však v naší práci nebudeme potřebovat a vystačíme s tenzorovými součiny, resp. tenzory sestrojenými v definici 1.33. Rovněž nebudeme potřebovat součiny duálních prostorů, tj.  $U \otimes U^*, U^* \otimes U^*$  a omezíme se pouze na součiny  $U \otimes V$ , resp.  $U \otimes U$ .

*Poznámka 6.* V dalších úvahách budeme používat číslování od nuly a to vzhledem k tomu, že budeme tenzorový součin aplikovat na NURBS plochy, kde se toto číslování používá.

V další práci (především v kapitole 3, části 3.3) bude mít zásadní důležitost jednoduchý důsledek definice 1.30.

**Věta 1.6.** *Nechť  $\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_m \in U$ ,  $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n \in V$  jsou libovolné vektory,  $a_0, a_1, \dots, a_m, b_0, b_1, \dots, b_n$  libovolné skaláry,  $\omega : U \times V \rightarrow T$  bilineární forma mezi  $U, V$ . Pak*

$$\omega \left( \sum_{i=0}^m a_i \cdot \mathbf{u}_i \sum_{j=0}^n b_j \cdot \mathbf{v}_j \right) = \sum_{i=0}^m \sum_{j=0}^n a_i b_j \cdot \omega(\mathbf{u}_i, \mathbf{v}_j). \quad (1.8)$$

Tento zápis připomíná obdobnou vlastnost skalárního součinu dvou vektorů. Skalární součin je speciálním případem bilineární formy. Máme-li totiž ve větě 1.6 speciálně:

$$\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_m \in U, \quad \mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_n \in V$$

báze prostorů  $U$ , resp.  $V$ , pak pro každý vektor  $u \in U$  o souřadnicích  $\mathbf{u} = (u_0, u_1, \dots, u_m)$ , resp.  $v \in V$  o souřadnicích  $\mathbf{v} = (v_0, v_1, \dots, v_n)$  je

$$\omega(\mathbf{u}, \mathbf{v}) = \omega \left( \sum_{i=0}^m u_i \cdot \mathbf{e}_i \sum_{j=0}^n v_j \cdot \mathbf{f}_j \right) = \sum_{i=0}^m \sum_{j=0}^n u_i v_j \cdot \omega(\mathbf{e}_i, \mathbf{f}_j) = \sum_{i=0}^m \sum_{j=0}^n u_i v_j \cdot a_{ij}. \quad (1.9)$$

kde hodnotu bilineární formy  $\omega$  pro uspořádanou dvojici bázevých vektorů  $\mathbf{e}_i \in U, \mathbf{f}_j \in V$  jsme označili

$$\omega(\mathbf{e}_i, \mathbf{f}_j) = a_{ij} \in T.$$

Je-li v bilineární formě (1.9)  $V = U$  nebo  $V = U^*$ , je tato forma dle definice 1.33 tenzorem.

Další specializací dostáváme:

Je-li  $V = U$ , tenzor  $\omega : U \times U \rightarrow T$  je symetrický a kvadratická forma příslušná k tomuto tenzoru je pozitivně definitní, tj. platí pro  $\forall \mathbf{x}, \mathbf{y} \in U$ :

$$\omega(\mathbf{x}, \mathbf{y}) = \omega(\mathbf{y}, \mathbf{x})$$

$$\mathbf{x} \neq \mathbf{0} \Rightarrow \omega(\mathbf{x}, \mathbf{x}) > 0$$

pak tenzor  $\omega$  splňuje axiomy skalárního součinu a je tedy skalárním součinem na prostoru  $U$ .

Odmocnina  $\sqrt{\omega(\mathbf{u}, \mathbf{u})}$  z pozitivně definitní formy splňuje všechny axiomy normy a je tedy normou vektoru indukovanou skalárním součinem  $\omega(\mathbf{u}, \mathbf{v})$ .

Je-li zcela speciálně:

$$\omega(\mathbf{u}, \mathbf{v}) = \sum_{i=0}^m \sum_{j=0}^m u_i v_j \omega(\mathbf{e}_i, \mathbf{f}_j) = \sum_{i=0}^m \sum_{j=0}^m u_i v_j a_{ij}, \quad (1.10)$$

kde

$$\omega(\mathbf{e}_i, \mathbf{f}_i) = a_{ii} = 1; i \neq j \Rightarrow \omega(\mathbf{e}_i, \mathbf{f}_j) = a_{ij} = 0.$$

je příslušná kvadratická forma tvaru

$$\omega(\mathbf{u}, \mathbf{u}) = \sum_{i=0}^m \sum_{j=0}^m u_i u_j \omega(\mathbf{e}_i, \mathbf{f}_j) = \sum_{i=0}^m \sum_{j=0}^m u_i u_j a_{ij} = \sum_{i=0}^m u_i^2 > 0 \quad (1.11)$$

a tenzor  $\omega(\mathbf{u}, \mathbf{v})$  přejde v tzv. bodový skalární součin tvaru:

$$\omega(\mathbf{u}, \mathbf{v}) = \sum_{i=0}^m \sum_{j=0}^m u_i v_j \omega(\mathbf{e}_i, \mathbf{f}_j) = \sum_{i=0}^m \sum_{j=0}^m u_i v_j a_{ij} = \sum_{i=0}^m u_i v_i = u_0 v_0 + u_1 v_1 + \dots + u_m v_m,$$

kde  $\mathbf{u} = (u_0, u_1, \dots, u_m)$ ,  $\mathbf{v} = (v_0, v_1, \dots, v_m)$ .

*Poznámka 7.* Výraz (1.9) – bilineární formu lze samozřejmě napsat maticově jako:

$$\omega(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{A} \mathbf{v}, \quad (1.12)$$

a je to:

- a) bodový skalární součin  $\Leftrightarrow \mathbf{A}$  je jednotková
- b) obecný skalární součin  $\Leftrightarrow \mathbf{A}$  je čtvercová, symetrická, pozitivně definitní
- c) tenzor  $\Leftrightarrow \mathbf{A}$  je čtvercová
- d) bilineární forma  $\Leftrightarrow \mathbf{A}$  je typu  $m \times n$

Ve všech případech je množina  $\omega(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{A} \mathbf{v}$  tenzorový součin vektorových prostorů a je to vektorový prostor.

Myšlenka využít tenzorového součinu se objevuje i v současné literatuře z oblasti týkající se počítačové grafiky. Plocha je popisována jako vyčíslitelná funkce dvou parametrů  $u, v$ , která popisuje zobrazení rovinné oblasti  $\Omega$  do euklidovského trojrozměrného prostoru, které je formálně zapsáno jako:

$$S(u, v) = (x(u, v), y(u, v), z(u, v)) = \sum_{i=0}^m \sum_{j=0}^n f_i(u) g_j(v) P_{ij}, \quad (1.13)$$

kde

$$\begin{cases} P_{ij} = (x_{ij}, y_{ij}, z_{ij}) \\ 0 \leq u, v \leq 1 \end{cases}$$

Funkce  $f, g$  jsou zcela běžně nazývány bázovými funkcemi. Takto sestrojená plocha bývá označována jako *tensor product surface* tj. povrch vytvořený tenzorovým součinem – viz např. Shene (Naposledy navštíveno 11. 12. 2006), Piegl – Tiller (2002), Piegl (1991).

Zápis (1.13) formálně skutečně připomíná vztah (1.9), kterým jsme definovali bilineární formu a tenzorový součin. Tento tenzor se však zobrazuje do tělesa, nad kterým jsou sestrojeny lineární prostory, ze kterých je tenzorový součin sestrojen. V jeho zápisu

$$\omega(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^m \sum_{j=1}^m u_i v_j \omega(\mathbf{e}_i, \mathbf{f}_j) = \sum_{i=1}^m \sum_{j=1}^m u_i v_j a_{ij} \quad (1.14)$$

jsou  $\omega(\mathbf{u}, \mathbf{v})$  a  $(\mathbf{e}_i, \mathbf{f}_j) = a_{ij}$  hodnoty bilineárního zobrazení,  $u_i, v_j$  jsou složky vektorů  $\mathbf{u}, \mathbf{v}$ , tj. všechno prvky výše uvedeného tělesa. Zápis (1.13) oproti tomu definuje zobrazení

oblasti do prostoru (pravděpodobně euklidovského).  $S(u, v)$  je tedy bod tohoto prostoru,  $f_i, g_j$  funkce báze blíže nespecifikovaného prostoru funkcionálního a  $P_{ij}$  jsou buď přímo řídicí body plochy, tj. opět prvky (pravděpodobně) euklidovského prostoru, anebo jejich polohové vektory, tj. prvky zaměření euklidovského prostoru, do kterého má tato funkce zobrazovat.

Toto přinejmenším velmi volné zacházení s matematickou symbolikou značně ztěžuje pochopení podobných textů a komplikuje výzkum v této oblasti.

Mějme danu rovnici (1.14), kterou lze zapsat jako:

$$\omega(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{A}_{m+1, n+1} \mathbf{v} \quad (1.15)$$

Skalární násobek definujeme jako:

$$c \cdot \omega(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T c \mathbf{A}_{m+1, n+1} \mathbf{v} \quad (1.16)$$

a součet forem  $\omega_1(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{A}_{m+1, n+1} \mathbf{v}$  a  $\omega_2(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{B}_{m+1, n+1} \mathbf{v}$  je dán:

$$\omega_1(\mathbf{u}, \mathbf{v}) + \omega_2(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T (\mathbf{A}_{m+1, n+1} + \mathbf{B}_{m+1, n+1}) \mathbf{v} \quad (1.17)$$

Tenzorový součin vektorových prostorů  $U, V$  je tak zřejmě izomorfní s vektorovým prostorem všech matic typu  $(m+1) \times (n+1)$ .

Mějme nyní  $N_i(s)$ ,  $i = 0, 1, \dots, m$ , resp.  $\bar{N}_j(t)$ ,  $j = 0, 1, \dots, n$  báze B-spline funkce libovolného stupně. Množiny

$$N = \langle N_0(s), N_1(s), \dots, N_m(s) \rangle$$

$$\bar{N} = \langle \bar{N}_0(t), \bar{N}_1(t), \dots, \bar{N}_n(t) \rangle$$

jsou funkcionální vektorové prostory nad  $\mathbb{R}$ , jejich prvky

$$u(s) = u_0 N_0(s) + u_1 N_1(s) + \dots + u_m N_m(s)$$

$$v(t) = v_0 \bar{N}_0(t) + v_1 \bar{N}_1(t) + \dots + v_n \bar{N}_n(t)$$

jsou B-spline křivky.

Tenzorový součin  $M \otimes N$  prostorů  $M, N$  je prostor bilineárních forem

$$\omega(\mathbf{u}, \mathbf{v}) = (u_0, \dots, u_m) \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0,n} \\ a_{10} & a_{11} & \dots & a_{1,n} \\ \vdots & & \ddots & \\ a_{m0} & a_{m1} & \dots & a_{m,n} \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \end{pmatrix}. \quad (1.18)$$

Bází tohoto prostoru je  $(m+1)(n+1)$  zobrazení

$$\omega_{ij}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{C}_{ij} \mathbf{v} \quad (1.19)$$

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

kde  $C_{ij}$  je matice typu  $(m+1)(n+1)$ , která je nulová s výjimkou členu  $c_{ij}$ , který je roven jedné.

Nechť  $\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_m$  je báze prostoru  $N$  a  $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n$  je báze prostoru  $\bar{N}$ . Vektory  $\mathbf{u}_i = (u_{i0}, u_{i1}, \dots, u_{im})$  a  $\mathbf{v}_j = (v_{j0}, v_{j1}, \dots, v_{jm})$  se zobrazením  $\omega(\mathbf{u}, \mathbf{v})$  se zobrazí na prvek  $c_{i,j}$  tělesa  $T$ . Báze prostoru  $N \times \bar{N}$  se zobrazí na uspořádanou  $(m+1)(n+1)$ -tici  $c_{i,j} \in T^{n+1, m+1}$ . Množina těchto  $(m+1)(n+1)$ -tic je opět vektorovým prostorem. Máme tedy zobrazení:

$$\omega(\mathbf{u}, \mathbf{v}) = \sum_{i=0}^m \sum_{j=0}^n c_{ij} \omega_{ij}(\mathbf{u}, \mathbf{v}).$$

Je-li tedy

$$\begin{aligned} \mathbf{u} &= (u_0, \dots, u_m) = u_0 N_0(s) + \dots + u_m N_m(s) \in N, \\ \mathbf{v} &= (v_0, \dots, v_n) = v_0 \bar{N}_0(t) + \dots + v_n \bar{N}_n(t) \in \bar{N}, \end{aligned}$$

pak jsou bázové vektory tvaru:

$$\begin{aligned} \mathbf{u}_i(s) &= (0, 0, \dots, 0, u_i = 1, 0, \dots, 0, 0) = 0 \cdot N_0(s) + \dots + 1 \cdot N_i(s) + \dots + 0 \cdot N_m(s) = \\ &= N_i(s) \in N \end{aligned}$$

$$\begin{aligned} \mathbf{v}_j(t) &= (0, 0, \dots, 0, v_j = 1, 0, \dots, 0, 0) = 0 \cdot \bar{N}_0(t) + \dots + 1 \cdot \bar{N}_j(t) + \dots + 0 \cdot \bar{N}_n(t) = \\ &= \bar{N}_j(t) \in \bar{N} \end{aligned}$$

Bázi prostoru  $T^{n+1, m+1}$  je potom množina zobrazení:

$$\omega_{ij}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{C}_{ij} \mathbf{v}, \quad \text{kde } i = 0, \dots, m, \quad j = 0, \dots, n.$$

Každé z těchto zobrazení zobrazí bázové vektory  $\mathbf{u}_i(s), \mathbf{v}_j(t)$  na vektor

$$c_{ij}(s, t) = C_{ij}(s, t) = \mathbf{u}_i \mathbf{C}_{ij} \mathbf{v}_j = \mathbf{u}_i \mathbf{v}_j = N_i(s) \bar{N}_j(t).$$

Každý prvek  $\mathbf{k} \in T^{n+1, m+1}$  je tedy lineární kombinací

$$\sum_{i=0}^m \sum_{j=0}^n k_{ij} \omega_{ij}(\mathbf{u}, \mathbf{v}) = \sum_{i=0}^m \sum_{j=0}^n k_{ij} N_i(s) \bar{N}_j(t).$$

V této reprezentaci značí prvky  $k_{ij}$  pouze jednu ze souřadnic bodu. Pro použití trojrozměrných bodů je nutné vytvořit speciální matici trojrozměrných bodů ve smyslu definice 1.27.

## 1.2 Pojem křivky a plochy

### 1.2.1 Křivky

V literatuře (např. Doupovec (1999), Lomtatidze (2005)) odvozují autoři základní definice z vektorové funkce. Nevýhodou tohoto pojetí je to, že dvě parametrická vyjádření mohou určovat tutéž křivku. Proto je výhodnější, používat definici pomocí třídy ekvivalentních cest.

*Definice 1.34.* Nechť je dáno zobrazení  $c(a, b) \rightarrow \mathcal{E}_3$ ,  $c(t) = [x_1(t), x_2(t), x_3(t)]$  tak, že pro každé  $t$  jsou funkce  $x_1(t), x_2(t), x_3(t)$  spojité, pak obor hodnot  $[c]$  zobrazení  $c$  nazýváme (neorientovanou) **křivkou**, zobrazení  $c$  její **parametrizací – cestou**.

Platí-li navíc

a) pro každé  $t$  mají funkce  $x_1(t), x_2(t), x_3(t)$  spojité derivace alespoň 1. řádu,

b) pro každé  $t$  je  $(\frac{dx_1}{dt})^2 + (\frac{dx_2}{dt})^2 + (\frac{dx_3}{dt})^2 \neq 0$ ,

pak nazýváme křivku  $[c]$  **regulární**.

*Poznámka 8.* V literatuře se za regulární křivku označuje většinou křivka, jejíž parametrizace je navíc injektivní (tj. křivka sama sebe neprotíná). V naší práci není třeba tyto body vylučovat, proto tento požadavek vynecháme.

*Definice 1.35.* Bod  $c(t)$  parametrizace  $c$  se nazývá **násobný**, pokud existují dvě různá reálná čísla  $t_1, t_2 \in \langle a, b \rangle$  taková, že  $c(t) = c(t_1) = c(t_2)$  a přitom  $\{t_1, t_2\} \neq \{a, b\}$ . V opačném případě se bod nazývá **jednoduchý**. Cesta se nazývá **jednoduchá**, je-li každý její bod jednoduchý.

*Definice 1.36.* Nechť  $[c]$  je regulární křivka,  $c(t) = [x_1(t), x_2(t), x_3(t)]$  její parametrizace,  $T \in [c]$  její libovolný bod. Přímkou  $t = \langle T, \mathbf{t} \rangle$ , kde

$$\mathbf{t} = \left( \frac{dx_1}{dt}, \frac{dx_2}{dt}, \frac{dx_3}{dt} \right) \in \mathcal{E}_3,$$

nazýváme **tečnou křivku** v bodě  $T$ .

*Definice 1.37.* Dvě cesty  $c_1 : \langle a_1, b_1 \rangle \rightarrow \mathbb{R}^2$  a  $c_2 : \langle a_2, b_2 \rangle \rightarrow \mathbb{R}^2$  se nazývají **ekvivalentní**, pokud existuje zobrazení  $\tau : \langle a_1, b_1 \rangle \rightarrow \langle a_2, b_2 \rangle$ , které je spojitě a rostoucí a platí:  $\tau(a_1) = a_2$ ,  $\tau(b_1) = b_2$  a pro každé  $t \in \langle a_1, b_1 \rangle$

$$c_1(t) = c_2(\tau(t)). \quad (1.20)$$

*Označení:* cesty  $c_1, c_2$  jsou ekvivalentní:  $c_1 \sim c_2$

**Věta 1.7.**  $\sim$  je relací ekvivalence na množině všech cest.

Podle relace ekvivalence se všechny cesty rozdělí do tříd ekvivalence. Z toho vyplývá následující definice:

*Definice 1.38.* **Orientovaná křivka** je třídou všech ekvivalentních cest.

## 1.2.2 Plochy

K určení polohy bodu plochy je potřeba dvou parametrů. Proto je nutné rozšířit předchozí úvahu.

*Definice 1.39.* Nechť  $D \subset \mathbb{R}$  je interval,  $\Omega \subset \mathbb{R}^2$  je souvislá otevřená množina, dále nechť  $p : \Omega \rightarrow \mathcal{E}_3$ ,  $\varphi : D \rightarrow \Omega$ ;  $\varphi(t) = (u(t), v(t)) \in \Omega$  jsou spojitá zobrazení. Pak obor hodnot  $[p]$  zobrazení  $p : \Omega \rightarrow \mathcal{E}_3$  nazýváme (neorientovanou) **plochou**, zobrazení  $p : \Omega \rightarrow \mathcal{E}_3$  její parametrizací, zobrazení  $f = p \circ \varphi : D \rightarrow \mathcal{E}_3$  nazveme **parametrizací křivky na ploše**.

Platí-li navíc pro každé  $(u, v) \in \Omega$

a) polohový vektor  $\mathbf{p}(u, v)$  bodu  $p(u, v)$  má spojité parciální derivace alespoň prvního řádu,

b) vektory  $\frac{\partial \mathbf{p}}{\partial u}, \frac{\partial \mathbf{p}}{\partial v}$  jsou lineárně nezávislé,

nazýváme plochu **regulární**.

*Poznámka 9.* Po regulární ploše je většinou opět vyžadována injektivnost (nemožnost samoprotínání), ze stejného důvodu jako u křivek ji však nepožadujeme. Některé dále studované plochy (povrch krychle, válce, kužele, apod.) nejsou regulární z důvodu existence bodů, ve kterých není definována derivace – vrcholy, hrany.

*Poznámka 10.* Křivky i plochy definujeme v afinním euklidovském prostoru daném repérem  $\langle O, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \rangle$ . Každému bodu  $X \in \mathcal{E}_3$  je tedy přiřazen polohový vektor  $\overrightarrow{OX}$ .

*Definice 1.40.* Rovinu  $\tau = \langle T, \mathbf{t}_1, \mathbf{t}_2 \rangle$ , kde

$$\mathbf{t}_1 = \frac{\partial \mathbf{p}}{\partial u}, \quad \mathbf{t}_2 = \frac{\partial \mathbf{p}}{\partial v}$$

nazýváme **tečnou rovinou** plochy  $[p]$  v bodě  $T \in [p]$ . Každou přímku tečné roviny, která prochází bodem  $T \in [p]$  nazýváme **tečnou plochy**  $[p]$  v bodě  $T \in [p]$ .

*Definice 1.41.* Křivku na ploše s parametrizací  $f = p \circ \varphi : D \rightarrow E_3$  kde  $p(u, v) = p(u, v_0)$ , resp.  $p(u, v) = p(u_0, v)$  nazveme  $u$  křivka, resp.  $v$  křivka na ploše.

**Věta 1.8.** Každým bodem  $p(u_0, v_0)$  regulární plochy prochází právě jedna  $u$  křivka, resp.  $v$  křivka na ploše. Tečné vektory těchto křivek v bodě  $p(u_0, v_0)$  jsou:

$$\begin{aligned} \frac{df(u_0)}{dt} &= \frac{\partial p}{\partial u}(u_0, v_0) \\ \frac{dg(v_0)}{dt} &= \frac{\partial p}{\partial v}(u_0, v_0) \end{aligned}$$

Je-li dán systém křivek na ploše tak, že každým bodem  $p(u_0, v_0)$  procházejí právě dvě křivky, které v něm mají různé tečny (tj. nedotýkají se), dostáváme síť křivek na ploše. Tato síť je obrazem jisté části kartézské souřadnicové sítě při zobrazení  $p$ , proto se nazývá **křivočaré souřadnice**.

Pro všechny křivky na ploše procházející bodem  $p(u, v)$  platí, že jejich tečny leží v tečné rovině. Naopak každá přímka tečné roviny, která prochází bodem  $p(u, v)$ , je tečnou nějaké křivky na ploše.

**Normálou plochy** v bodě  $p(u, v)$  nazveme přímku procházející tímto bodem kolmo k tečné rovině v tomto bodě. Normálový vektor je dán jako:

$$\mathbf{n} = \frac{\partial \mathbf{p}}{\partial u} \times \frac{\partial \mathbf{p}}{\partial v}$$

Orientace křivky byla řešena spojitým rostoucím zobrazením. U ploch je nutné pracovat s jakobiánem.

*Definice 1.42.* Necht'  $p_1(u(t), v(t)) : D \rightarrow \mathcal{E}_3$ ,  $p_2(\bar{u}(t), \bar{v}(t)) : D \rightarrow \mathcal{E}_3$  jsou dvě libovolné parametrizace téže regulární plochy  $[p]$ . Dále necht'  $\psi_1(u, v) : \Omega \rightarrow \mathbb{R}$ ,  $\psi_2(u, v) : \Omega \rightarrow \mathbb{R}$  jsou zobrazení taková, že pro každé  $[u, v] \in \Omega$  je

$$\bar{u} = \psi_1(u, v), \quad \bar{v} = \psi_2(u, v), \quad [\bar{u}, \bar{v}] \in \bar{\Omega}$$

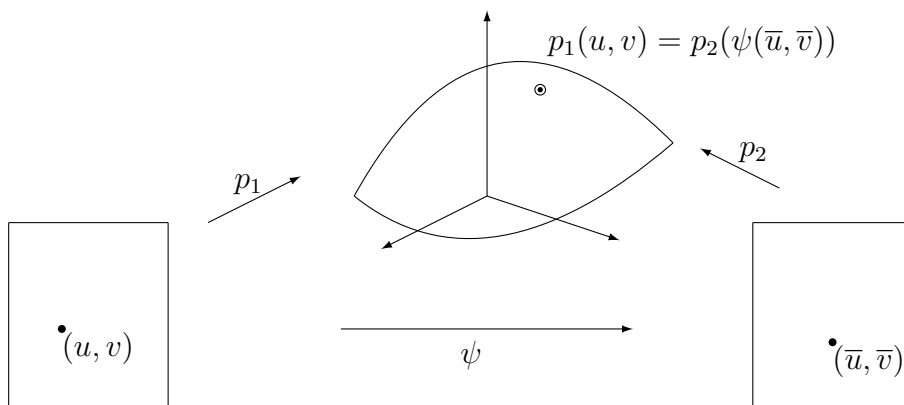
a

$$J = \begin{vmatrix} \frac{\partial \psi_1}{\partial u} & \frac{\partial \psi_1}{\partial v} \\ \frac{\partial \psi_2}{\partial u} & \frac{\partial \psi_2}{\partial v} \end{vmatrix} \neq 0. \quad (1.21)$$

Pak plochu nazýváme **orientovatelnou**.

**Věta 1.9.** Na množině  $\langle p \rangle$  všech parametrizací orientovatelné plochy  $[p]$  definujeme relaci  $p_1 \sim p_2$  právě tehdy, když  $J > 0$ . Pak relace  $\sim$  je ekvivalence na  $[p]$ .

*Definice 1.43.* Třídy rozkladu indukovaného ekvivalencí z předchozí věty nazveme **orientovanými plochami**.



Obrázek 1.1: Reparametrizace plochy

*Poznámka 11.* S orientovanými křivkami a plochami nebudeme dále pracovat. Jejich definice jsou zde uvedeny pro úplnost. Křivky a plochy jsou zobrazeny bodově, proto na závěr nadefinujeme graf pro funkci více proměnných.

*Definice 1.44.* Nechť je dána funkce  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . **Grafem**  $G$  funkce nazveme množinu všech uspořádaných  $n + 1$ -tic tvaru:

$$G = (x_0, \dots, x_n, f(x_0, \dots, x_n)).$$



# Kapitola 2

## Současný stav problematiky

V šedesátých letech minulého století se v technické praxi objevila potřeba speciálních křivek a ploch. Významní geometři a matematici (Carl de Boor, James C. Ferguson, Pierre Bézier, Paul F. de Casteljau) začali vývoj nových typů křivek, které se zadávají řídicími body a dalšími parametry. První z nich byly Fergusonovy, Coonsovy a Bézierovy křivky a plochy. Jejich nevýhodou byla pouze lokální kontrolovatelnost (tj. po změně jednoho bodu se musí přepočítat celá křivka) a špatná ovladatelnost. Podrobnější informace včetně programového zpracování lze nalézt v Martišek (2002), Žára et al. (2004).

Vývoj pokračoval dál a již v sedmdesátých letech se objevují B-spline křivky. Jejich zobecněním jsou NURBS, tedy Neuniformní Racionální B-spline křivky a plochy. NURBS objekty jsou tedy B-spline objekty, jejichž uzlový vektor může být neekvidistantní a ke každému bodu je přidána váha.

Mezi průkopníky B-spline křivek patří Carl de Boor, který položil teoretické základy v de Boor (1976). Podrobný popis vlastností a některých algoritmů lze nalézt v Shene (Naposledy navštíveno 11. 12. 2006), Piegl (1991), Tiller (1983) Zkoumání NURBS křivek a ploch se věnovala celá řada vědců a za shrnutí velké části znalostí je považována kniha Piegl – Tiller (2002).

Vzhledem k již poměrně dlouhému vývoji existuje celá řada vědeckých článků zabývajících se studiem týkajícím se NURBS objektů či vlastností. Základní změny tvaru racionálních B - spline křivky jsou popsány v Piegl (1989). Tvorba kruhových oblouků pomocí NURBS křivek je podrobně probrána v Tiller – Piegl (1989). Ovládání kubických B-spline křivek pomocí uzlového vektoru najdeme v práci Juhasz – Hoffmann (2004). Při posunu jednoho parametru z uzlového vektoru se vytváří třída křivek a existuje také jejich obálka, což je podrobně popsáno v Juhasz – Hoffmann (2001). Implementaci NURBS křivek v Javě najdeme v práci Alexandr (1999).

Pomocí NURBS křivek a ploch lze samozřejmě interpolovat množinu bodů. V současné době se používá několik způsobů, první z nich je nalezení nových řídicích bodů, aby křivka či plocha původními procházela. Uzlový vektor se volí podle daného předpisu automaticky (více v Aszódi et al. (2004)). Další možností je generování uzlového vektoru pomocí genetických algoritmů (viz Bercovier – Goldenthal (2003)). V současné době se předchozí postupy doplňují ještě o optimalizační metody (více v Bercovier – Goldenthal (2004)).

Z fyzikálního pohledu jsou nejzajímavější tzv. D-NURBS (Dynamic NURBS), které

v sobě dokáží zachytit další fyzikální vlastnosti a staly se součástí při měření fyzikálních jevů (více viz Terzopoulos – Qin (1994), Terzopoulos – Qin (1996)).

Jak již bylo zmíněno, za nástupce NURBS jsou považovány T-spline, neboli NURBS plochy s T-Junctions. Hlavní výhodou T-spline ploch je úspora paměti a jednodušší a rychlejší způsob vykreslení bez přebytečných bodů. V současné době se používají pouze v softwaru Autodesk Maya a Rhinoceros, kde slouží k hladkému napojování plátů a k jednoduššímu a dokonalejšímu vykreslení složitých tvarů.

Teoretický základ podává Sederberg et al. (2003a), Sederberg et al. (2004). Obecně jsou T-Junctions hrany, z kterých vychází další hrany i v bodech, které nejsou koncové. T-spline plocha je tedy definována nad nepravidelnou sítí.

## 2.1 Předchůdci B-spline

### 2.1.1 Fergusonovy křivky a plochy

Základní oblouk je určen čtyřmi body  $P_0, P_1, P_2, P_3$ . Křivka má krajní body  $P_0, P_3$ . Tečné vektory v těchto bodech jsou  $\overrightarrow{P_0P_1}, \overrightarrow{P_3P_2}$ . Velikost těchto vektorů ovlivňuje tvar křivky – čím je vektor větší, tím více se křivka k vektoru přibližuje.

Křivka je definována parametrickými rovnicemi:

$$Q(t) = \sum_{i=0}^3 P_i F_i(t), \quad t \in \langle 0, 1 \rangle \quad (2.1)$$

$$\begin{aligned} F_0(t) &= t^3 - t^2 - t + 1 \\ F_1(t) &= t^3 - 2t^2 + t \\ F_2(t) &= -3t^3 + 4t^2 \\ F_3(t) &= t^3 - t^2 \end{aligned} \quad (2.2)$$

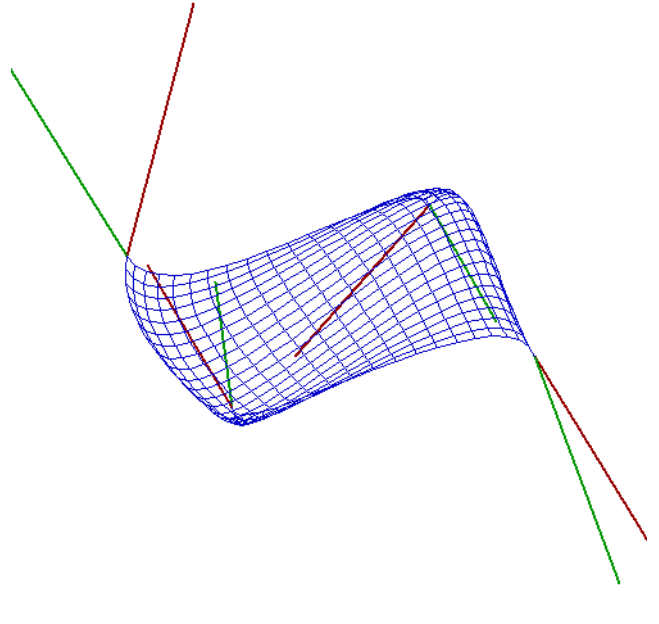
Dvojměrným zobecněním Fergusonových křivek dostaneme Fergusonovy plochy. Jsou dány sítí reálných bodů  $P_{ij}$  (dle definice 1.27) a jejich rovnice je:

$$Q(r, s) = \sum_{i=0}^3 \sum_{j=0}^3 P_{ij} F_i(r) F_j(s), \quad (2.3)$$

kde  $F_i$  jsou polynomy definované v rovnici (2.2) a  $(r, s) \in \langle 0, 1 \rangle \times \langle 0, 1 \rangle$ .

### 2.1.2 Bézierovy křivky a plochy

Bézierovy křivky a plochy jsou velmi často používány v technické praxi. Za jejich zakladatele je považován P. E. Bézier a nezávisle na něm také P. de Casteljaeu. Křivka je určena čtyřmi body a prochází prvním a posledním z nich. Vektory  $\overrightarrow{P_0P_1}, \overrightarrow{P_3P_2}$  určují tečny v krajních bodech. Jejich směrnice jsou rovny třetině délky těchto úseček.



Obrázek 2.1: Fergusonova plocha

Křivka je definována parametrickými rovnicemi:

$$Q(t) = \sum_{i=0}^3 P_i B_i(t), \quad t \in \langle 0, 1 \rangle \quad (2.4)$$

$$\begin{aligned} B_0(t) &= (1-t)^3 \\ B_1(t) &= 3t(1-t)^2 \\ B_2(t) &= 3t^2(1-t) \\ B_3(t) &= t^3 \end{aligned} \quad (2.5)$$

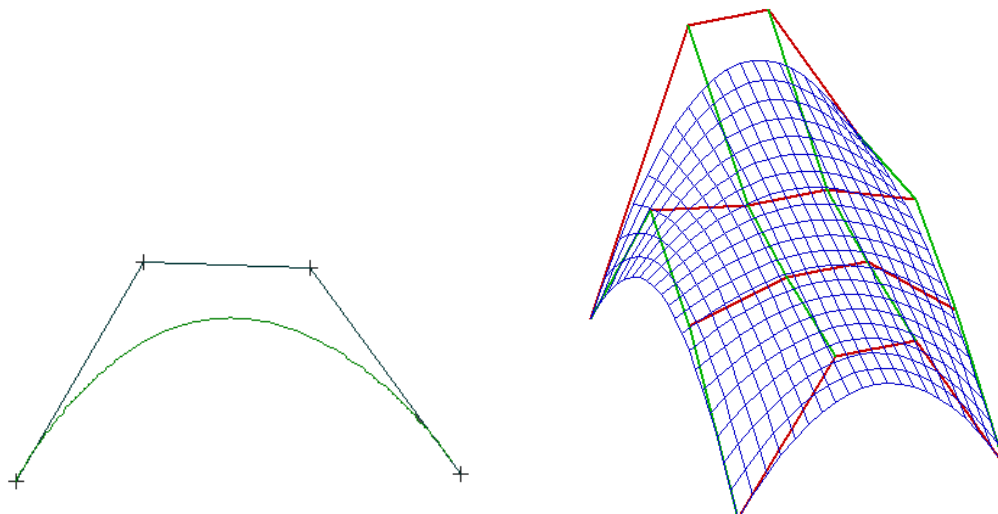
Obecná Bézierova křivka vznikne zobecněním předchozího případu. Bázové polynomy tvoří binomický rozvoj.

Mějme dáno  $n + 1$  řídicích bodů, potom je Bézierova křivka tvaru:

$$\begin{aligned} Q(t) &= \sum_{i=0}^n P_i B_i^n(t), \quad t \in \langle 0, 1 \rangle, \\ B_i^n(t) &= \binom{n}{i} t^i (1-t)^{n-i}. \end{aligned} \quad (2.6)$$

Dvojrozměrným zobecněním Bézierových křivek dostaneme Bézierovy plochy. Jsou dány sítí reálných bodů  $P_{ij}$  (dle definice 1.27) a rovnice plochy je:

$$Q(r, s) = \sum_{i=0}^3 \sum_{j=0}^3 P_{ij} B_i(r) B_j(s), \quad (2.7)$$



Obrázek 2.2: Bézierova křivka a plocha

kde  $B_i$  jsou polynomy definované v rovnici (2.5) a  $(r, s) \in \langle 0, 1 \rangle \times \langle 0, 1 \rangle$ .

### 2.1.3 Coonsovy křivky a plochy

Podobné Fergusonovým křivkám jsou Coonsovy křivky, pojmenované po S. A. Coonsovi. Jediným rozdílem je tvar bazových polynomů. Základní oblouk je opět určen čtyřmi body  $P_0, P_1, P_2, P_3$ . Křivka však nezačíná ani nekončí v žádném z těchto bodů. Počáteční a koncový bod leží v antitěžištích trojúhelníků  $P_0P_1P_2$  a  $P_1P_2P_3$ .

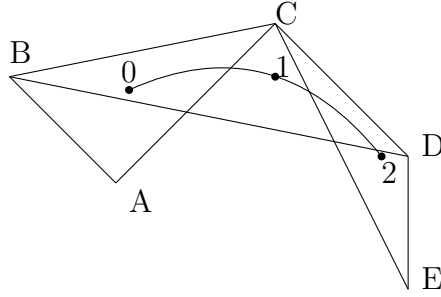
Křivka je definována parametrickými rovnicemi:

$$Q(t) = \frac{1}{6} \sum_{i=0}^3 P_i C_i(t), \quad t \in \langle 0, 1 \rangle \quad (2.8)$$

$$\begin{aligned} C_0(t) &= (1-t)^3 \\ C_1(t) &= 3t^3 - 6t^2 + 4 \\ C_2(t) &= 3t^3 + 3t^2 + 3t + 1 \\ C_3(t) &= t^3 \end{aligned} \quad (2.9)$$

Pro další vývoj je důležitá násobnost řídicích bodů Coonsových kubik. Položíme-li  $P_0 = P_1$ , stává se z prvního trojúhelníku úsečka a počáteční bod Consovy kubiky leží v jedné šestině této úsečky. Je-li  $P_0 = P_1 = P_2$ , zdegeneruje trojúhelník v bod a křivka začíná v tomto bodě.

Jednotlivé Coonsovy kubiky je možné hladce napojit. Pokud má navazující oblouk první tři body shodné s posledními třemi předchozího oblouku, dostáváme po částech polynomiální křivku, která se nazývá Coonsov kubický B-spline. Body styku se nazývají



Obrázek 2.3: Napojování Coonsových oblouků

uzly a označujeme je  $t$  (viz obr. 2.3). Coonsův B-spline je určen  $n \geq 4$  body a je tvořen  $n - 3$  segmenty. Parametr  $t$  definuje tzv. uzlový vektor. Jeho hodnoty jsou ekvidistantní, tj. rozdíl sousedních hodnot je konstanta. Zobecněním Coonsovy kubické B-spline křivky dostáváme B-spline křivky a plochy s libovolným stupněm. Dalším zobecněním je volba neekvidistantním uzlového vektoru – více v části 2.2. Přidáním vah se již hovoří o NURBS.

Dvojrozměrným zobecněním Coonsových křivek dostaneme Coonsovy plochy. Jsou dány sítí reálných bodů  $P_{ij}$  (dle definice 1.27) a rovnice plochy je:

$$Q(r, s) = \sum_{i=0}^3 \sum_{j=0}^3 P_{ij} C_i(r) C_j(s), \quad (2.10)$$

kde  $C_i$  jsou polynomy definované v rovnici (2.9) a  $(r, s) \in \langle 0, 1 \rangle \times \langle 0, 1 \rangle$

## 2.2 B-spline, NURBS křivky

### 2.2.1 B-spline funkce

Jako bázové funkce se pro B-spline a NURBS používají B-spline funkce. Jsou definovány rekurentně. Jejich definice je v současné době uvedena ve většině literatury týkající se grafiky – například Žára et al. (2004), Piegl – Tiller (2002).

*Definice 2.1.* Necht'  $\mathbf{t} = (t_0, t_1, \dots, t_n)$  je uzlový vektor. **B-spline funkce** je definována jako:

$$N_i^0(t) = \begin{cases} 1 & \text{pro } t \in \langle t_i, t_{i+1} \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_i^k(t) = \frac{t - t_i}{t_{i+k} - t_i} N_i^{k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1}^{k-1}(t), \quad (2.11)$$

$$\text{kde } 0 \leq i \leq n - k - 1, 1 \leq k \leq n - 1, \frac{0}{0} := 0.$$

Uzlový vektor je tvořen neklesající posloupností kladných reálných čísel. Obvykle se zadává v intervalu  $\langle 0, 1 \rangle$ , obecně lze volit libovolný interval.

*Příklad 2.1.* Spočítejme B-spline funkce stupně 0, 1, 2, 3 pro uzlový vektor  $(t_0, t_1, \dots, t_6) = (0, 1, 2, 3, 4, 5, 6, 7)$ .

#### B-spline funkce stupně nula

$$N_i^0(t) = \begin{cases} 1 & \text{pro } t \in \langle t_i, t_{i+1} \rangle \\ 0 & \text{jinak,} \end{cases}$$

$$\text{kde } i = 0, \dots, 6$$

#### B-spline funkce stupně jedna

$$N_0^1(t) = \begin{cases} t & \text{pro } t \in \langle 0, 1 \rangle \\ 2 - t & \text{pro } t \in \langle 1, 2 \rangle \\ 0 & \text{jinak} \end{cases}$$

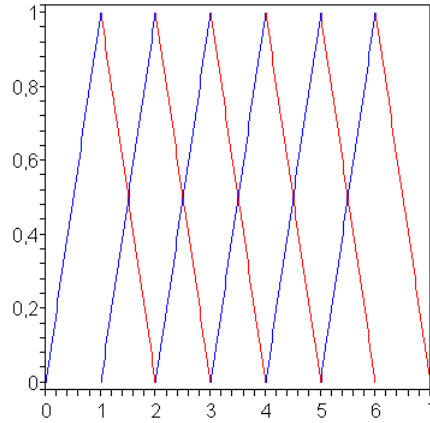
$$N_1^1(t) = \begin{cases} t - 1 & \text{pro } t \in \langle 1, 2 \rangle \\ 3 - t & \text{pro } t \in \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_2^1(t) = \begin{cases} t - 2 & \text{pro } t \in \langle 2, 3 \rangle \\ 4 - t & \text{pro } t \in \langle 3, 4 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^1(t) = \begin{cases} t-3 & \text{pro } t \in \langle 3, 4 \rangle \\ 5-t & \text{pro } t \in \langle 4, 5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^1(t) = \begin{cases} t-4 & \text{pro } t \in \langle 4, 5 \rangle \\ 6-t & \text{pro } t \in \langle 5, 6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_5^1(t) = \begin{cases} t-5 & \text{pro } t \in \langle 5, 6 \rangle \\ 7-t & \text{pro } t \in \langle 6, 7 \rangle \\ 0 & \text{jinak} \end{cases}$$

Obrázek 2.4: B-spline funkce stupně jedna pro vektor  $(0, 1, 2, 3, 4, 5, 6, 7)$ **B-spline funkce stupně dva**

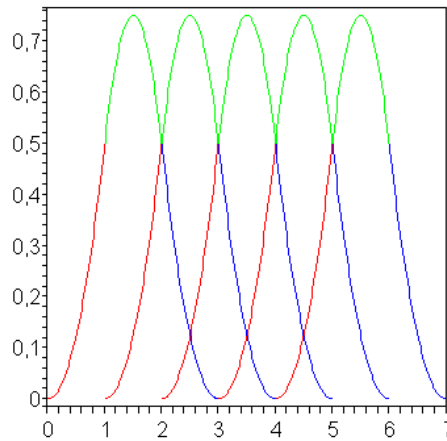
$$N_0^2(t) = \begin{cases} t^2/2 & \text{pro } t \in \langle 0, 1 \rangle \\ t(2-t)/2 + (t-1)(3-t)/2 & \text{pro } t \in \langle 1, 2 \rangle \\ (3-t)^2/2 & \text{pro } t \in \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_1^2(t) = \begin{cases} (t-1)^2/2 & \text{pro } t \in \langle 1, 2 \rangle \\ (t-1)(3-t)/2 + (t-2)(4-t)/2 & \text{pro } t \in \langle 2, 3 \rangle \\ (4-t)^2/2 & \text{pro } t \in \langle 3, 4 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_2^2(t) = \begin{cases} (t-2)^2/2 & \text{pro } t \in \langle 2, 3 \rangle \\ (t-2)(4-t)/2 + (t-3)(5-t)/2 & \text{pro } t \in \langle 3, 4 \rangle \\ (5-t)^2/2 & \text{pro } t \in \langle 4, 5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^2(t) = \begin{cases} (t-3)^2/2 & \text{pro } t \in \langle 3, 4 \rangle \\ (t-3)(5-t)/2 + (t-4)(6-t)/2 & \text{pro } t \in \langle 4, 5 \rangle \\ (6-t)^2/2 & \text{pro } t \in \langle 5, 6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^2(t) = \begin{cases} (t-4)^2/2 & \text{pro } t \in \langle 4, 5 \rangle \\ (t-4)(6-t)/2 + (t-5)(7-t)/2 & \text{pro } t \in \langle 5, 6 \rangle \\ (7-t)^2/2 & \text{pro } t \in \langle 6, 7 \rangle \\ 0 & \text{jinak} \end{cases}$$



Obrázek 2.5: B-spline funkce stupně dva pro vektor  $(0, 1, 2, 3, 4, 5, 6, 7)$

### B-spline funkce stupně tři

$$N_0^3(t) = \begin{cases} t^3/6 & \text{pro } t \in \langle 0, 1 \rangle \\ [t^2(2-t) + t(3-t)(t-1) + (4-t)(t-1)^2]/6 & \text{pro } t \in \langle 1, 2 \rangle \\ [t(3-t)^2 + (4-t)(t-1)(3-t) + (4-t)^2(t-2)]/6 & \text{pro } t \in \langle 2, 3 \rangle \\ (4-t)^3/6 & \text{pro } t \in \langle 3, 4 \rangle \\ 0 & \text{jinak} \end{cases}$$

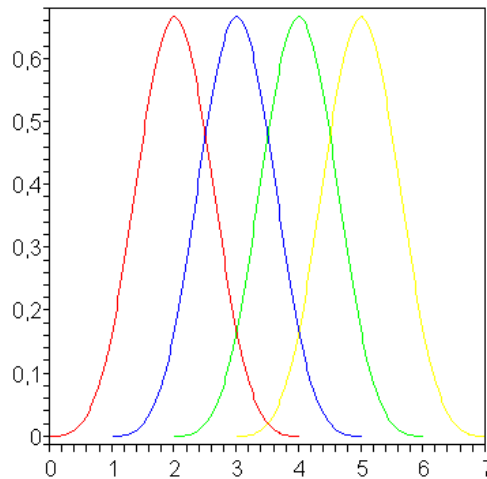


$$N_1^3(t) = \begin{cases} (t-1)^3/6 & t \in \langle 1, 2 \rangle \\ [(t-1)^2(3-t) + (t-1)(4-t)(t-2) + (5-t)(t-2)^2]/6 & t \in \langle 2, 3 \rangle \\ [(t-1)(4-t)^2 + (5-t)(t-2)(4-t) + (5-t)^2(t-3)]/6 & t \in \langle 3, 4 \rangle \\ (5-t)^3/6 & t \in \langle 4, 5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_2^3(t) = \begin{cases} (t-2)^3/6 & t \in \langle 2, 3 \rangle \\ [(t-2)^2(4-t) + (t-2)(5-t)(t-3) + (6-t)(t-3)^2]/6 & t \in \langle 3, 4 \rangle \\ [(t-2)(5-t)^2 + (6-t)(t-3)(5-t) + (6-t)^2(t-4)]/6 & t \in \langle 4, 5 \rangle \\ (6-t)^3/6 & t \in \langle 5, 6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^3(t) = \begin{cases} (t-3)^3/6 & t \in \langle 3, 4 \rangle \\ [(t-3)^2(5-t) + (t-3)(6-t)(t-4) + (7-t)(t-4)^2]/6 & t \in \langle 4, 5 \rangle \\ [(t-3)(6-t)^2 + (7-t)(t-4)(6-t) + (7-t)^2(t-5)]/6 & t \in \langle 5, 6 \rangle \\ (7-t)^3/6 & t \in \langle 6, 7 \rangle \\ 0 & \text{jinak} \end{cases}$$

Jedná se o ekvidistantní uzly, proto jsou všechny funkce totožné, liší se pouze o posunutí. Výsledné funkce vykresleny v Maple jsou na obr. 2.4, 2.5, 2.6.



Obrázek 2.6: B-spline funkce stupně tři pro vektor  $(0, 1, 2, 3, 4, 5, 6, 7)$

### Vliv uzlového vektoru

Uzlový vektor je, jak již bylo řečeno, neklesající posloupnost kladných reálných čísel. Pokud je průběh plynulý, jedná se o ekvidistantní (uniformní) uzlový vektor, v opačném případě se nazývá neekvidistantní, jak udává definice 2.2.

V této části bude ukázán výpočet B-spline funkcí pro různé tvary uzlového vektoru. V části 2.2.2 týkající se B-spline křivek bude vysvětleno, jak různé tvary uzlových vektorů působí na tvar výsledné křivky.

*Definice 2.2.* Uzlový vektor  $\mathbf{t} = (t_0, t_1, \dots, t_r)$  se nazývá **ekvidistantní** – uniformní, jestliže platí

$$t_{i+1} - t_i = t_{i+2} - t_{i+1}, \text{ kde } i = 0, 1, \dots, t_{r-2}. \quad (2.12)$$

Pokud není rovnost (2.12) splněna, nazývá se uzlový vektor **neekvidistantní** – neuniformní.

*Definice 2.3.* Nechť  $\mathbf{t} = (t_0, t_1, \dots, t_r)$  je uzlový vektor, potom počet jeho prvků  $r + 1$  nazýváme **délkou uzlového vektoru**.

*Příklad 2.2.* Spočítejme B-spline funkce pro neekvidistantní uzlový vektor  $(0, 1, 2, 3.8, 4, 4.2, 6, 7)$ .

### B-spline funkce stupně nula

$$N_i^0(t) = \begin{cases} 1 & \text{pro } t \in \langle t_i, t_{i+1} \rangle \\ 0 & \text{jinak,} \end{cases}$$

$$\text{kde } i = 0, 1, \dots, 6.$$

### B-spline funkce stupně jedna

$$N_0^1(t) = \begin{cases} t & \text{pro } t \in \langle 0, 1 \rangle \\ 2 - t & \text{pro } t \in \langle 1, 2 \rangle \\ 0 & \text{jinak} \end{cases}$$

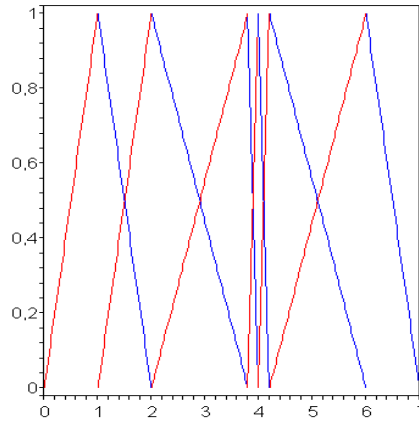
$$N_1^1(t) = \begin{cases} t - 1 & \text{pro } t \in \langle 1, 2 \rangle \\ (3.8 - t)/1.8 & \text{pro } t \in \langle 2, 3.8 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_2^1(t) = \begin{cases} (t - 2)/1.8 & \text{pro } t \in \langle 2, 3.8 \rangle \\ (4 - t)/0.2 & \text{pro } t \in \langle 3.8, 4 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^1(t) = \begin{cases} (t - 3.8)/0.2 & \text{pro } t \in \langle 3.8, 4 \rangle \\ (4.2 - t)/0.2 & \text{pro } t \in \langle 4, 4.2 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^1(t) = \begin{cases} (t-4)/0.2 & \text{pro } t \in \langle 4, 4.2 \rangle \\ (6-t)/1.8 & \text{pro } t \in \langle 4.2, 6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_5^1(t) = \begin{cases} (t-4.2)/1.8 & \text{pro } t \in \langle 4.2, 6 \rangle \\ 7-t & \text{pro } t \in \langle 6, 7 \rangle \\ 0 & \text{jinak} \end{cases}$$

Obrázek 2.7: B-spline funkce stupně jedna pro vektor  $(0, 1, 2, 3.8, 4, 4.2, 6, 7)$ **B-spline funkce stupně dva**

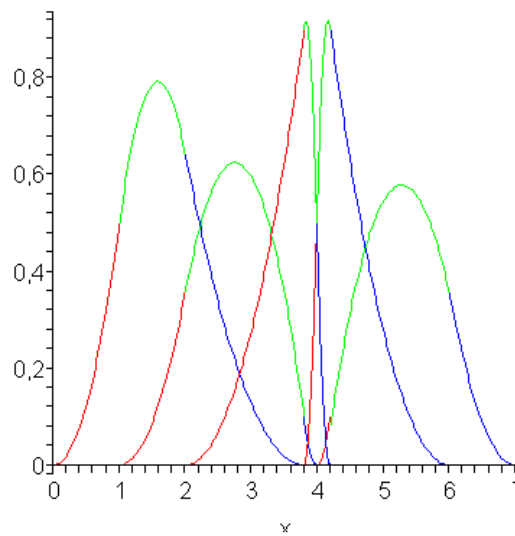
$$N_0^2(t) = \begin{cases} t^2/2 & \text{pro } t \in \langle 0, 1 \rangle \\ t(2-t)/2 + (t-1)(3.8-t)/2.8 & \text{pro } t \in \langle 1, 2 \rangle \\ (3.8-t)^2/5.04 & \text{pro } t \in \langle 2, 3.8 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_1^2(t) = \begin{cases} (t-1)^2/2.8 & \text{pro } t \in \langle 1, 2 \rangle \\ (t-1)(3.8-t)/5.04 + (t-2)(4-t)/3.6 & \text{pro } t \in \langle 2, 3.8 \rangle \\ (4-t)^2/0.4 & \text{pro } t \in \langle 3.8, 4 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_2^2(t) = \begin{cases} (t-2)^2/3.6 & \text{pro } t \in \langle 2, 3.8 \rangle \\ (t-2)(4-t)/0.4 + (t-3.8)(4.2-t)/0.08 & \text{pro } t \in \langle 3.8, 4 \rangle \\ (4.2-t)^2/0.08 & \text{pro } t \in \langle 4, 4.2 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^2(t) = \begin{cases} (t-3.8)^2/0.08 & \text{pro } t \in \langle 3.8, 4 \rangle \\ (t-3.8)(4.2-t)/0.08 + (t-4)(6-t)/0.4 & \text{pro } t \in \langle 4, 4.2 \rangle \\ (6-t)^2/3.6 & \text{pro } t \in \langle 4.2, 6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^2(t) = \begin{cases} (t-4)^2/0.4 & \text{pro } t \in \langle 4, 4.2 \rangle \\ (t-4)(6-t)/4.4 + (t-4.2)(7-t)/5.04 & \text{pro } t \in \langle 4.2, 6 \rangle \\ (7-t)^2/2.8 & \text{pro } t \in \langle 6, 7 \rangle \\ 0 & \text{jinak} \end{cases}$$



Obrázek 2.8: B-spline funkce stupně dva pro vektor  $(0, 1, 2, 3.8, 4, 4.2, 6, 7)$

Výsledné funkce jsou vykresleny na obr. 2.7, 2.8.

*Příklad 2.3.* Vezmeme-li uzlový vektor  $(0, 0, 0, 1, 2, 3, 3, 3)$  a vypočteme pro něj B-spline funkci stupně jedna a dva, dostaneme následující polynomy, které jsou zobrazeny na obr. 2.9, 2.10.

### B-spline funkce stupně nula

$$N_0^0(t) = 0$$

$$N_1^0(t) = 0$$

$$N_2^0(t) = \begin{cases} 1 & \text{pro } t \in \langle 0, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^0(t) = \begin{cases} 1 & \text{pro } t \in \langle 1, 2 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^0(t) = \begin{cases} 1 & \text{pro } t \in \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_5^0(t) = \begin{cases} 1 & \text{pro } t = 3 \\ 0 & \text{jinak} \end{cases}$$

$$N_6^0(t) = 0$$

**B-spline funkce stupně jedna**

$$N_0^1(t) = 0$$

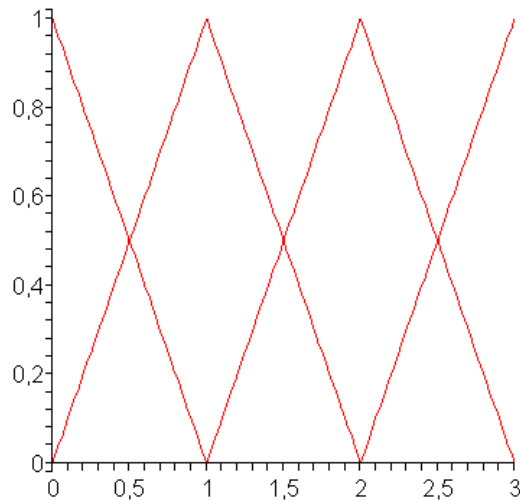
$$N_1^1(t) = \begin{cases} 1 - t & \text{pro } t \in \langle 0, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_2^1(t) = \begin{cases} t & \text{pro } t \in \langle 0, 1 \rangle \\ 2 - t & \text{pro } t \in \langle 1, 2 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^1(t) = \begin{cases} t - 1 & \text{pro } t \in \langle 1, 2 \rangle \\ 3 - t & \text{pro } t \in \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^1(t) = \begin{cases} t - 2 & \text{pro } t \in \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_5^1(t) = 0$$

Obrázek 2.9: B-spline funkce stupně jedna pro vektor  $(0, 0, 0, 1, 2, 3, 3, 3)$ **B-spline funkce stupně dva**

$$\begin{aligned}
 N_0^2(t) &= \begin{cases} (1-t)^2 & \text{pro } t \in \langle 0, 1 \rangle \\ 0 & \text{jinak} \end{cases} \\
 N_1^2(t) &= \begin{cases} t(1-t) + t(2-t)/2 & \text{pro } t \in \langle 0, 1 \rangle \\ (2-t)^2/2 & \text{pro } t \in \langle 1, 2 \rangle \\ 0 & \text{jinak} \end{cases} \\
 N_2^2(t) &= \begin{cases} t^2/2 & \text{pro } t \in \langle 0, 1 \rangle \\ t(2-t)/2 + (t-1)(3-t)/2 & \text{pro } t \in \langle 1, 2 \rangle \\ (3-t)^2/2 & \text{pro } t \in \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases} \\
 N_3^2(t) &= \begin{cases} (t-1)^2/2 & \text{pro } t \in \langle 1, 2 \rangle \\ (t-1)(3-t)/2 + (t-2)(3-t) & \text{pro } t \in \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases} \\
 N_4^2(t) &= \begin{cases} (t-2)^2 & \text{pro } t \in \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases}
 \end{aligned}$$

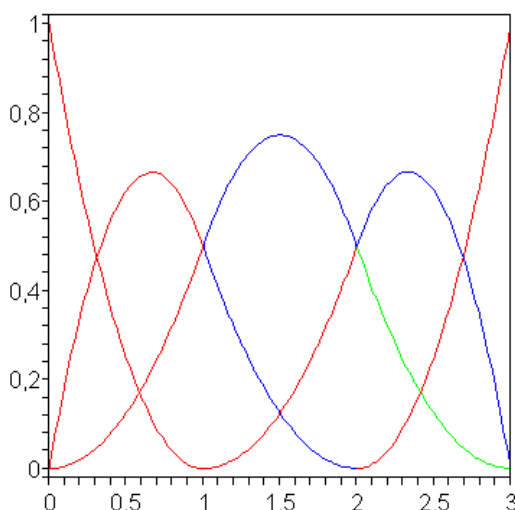
**Vlastnosti B-spline funkce**

## 1. Nezápornost

Pro libovolné  $i, k, t$  je  $N_i^k(t) \geq 0$ .

## 2. Lokální podpůrné intervaly

 $N_i^k(t) \neq 0$  pro  $t \in \langle t_i, t_{i+n+1} \rangle$ .

Obrázek 2.10: B-spline funkce stupně dva pro vektor  $(0, 0, 0, 1, 2, 3, 3, 3)$ 

## 3. Vlastnost intervalu

Na libovolném intervalu  $\langle t_i, t_{i+1} \rangle$  je nejvýše  $k + 1$  bázových funkcí stupně  $k$  nenulových, jmenovitě  $N_{i-k}^k, N_{i-k+1}^k, \dots, N_i^k$ .

## 4. Rozklad jednotky

Součet B-spline funkcí stupně  $k$  na intervalu  $\langle t_i, t_{i+1} \rangle$  je jedna.

$$\sum_{j=0}^k N_{i-k+j}^k = 1$$

## 5. Ekvidistantní uzlový vektor

Pro ekvidistantní uzlový vektor jsou funkce  $N_i^k$  až na posun podél osy parametrů identické.

$$N_i^k(t) = N_{i+1}^k(t+1)$$

## 6. Spojitost

V uzlu násobnosti  $s$  má bázová funkce  $N_i^k$  spojitost  $C^{k-s}$

**Algoritmus výpočtu B-spline funkcí**

B-spline funkce jsou definovány rekurentně (viz def. 2.1 na str. 24). Rekurzivní výpočet je velmi časově náročný. Proto se bere v úvahu důležitá vlastnost B-spline funkce (str. 33, bod 3) o nenulových hodnotách na intervalu. Při výpočtu můžeme postupovat podle stromu uvedeného na obr. 2.11. Popis algoritmu najdeme například v Piegl – Tiller (2002), Shene (Naposledy navštíveno 11. 12. 2006).

**Algoritmus:**

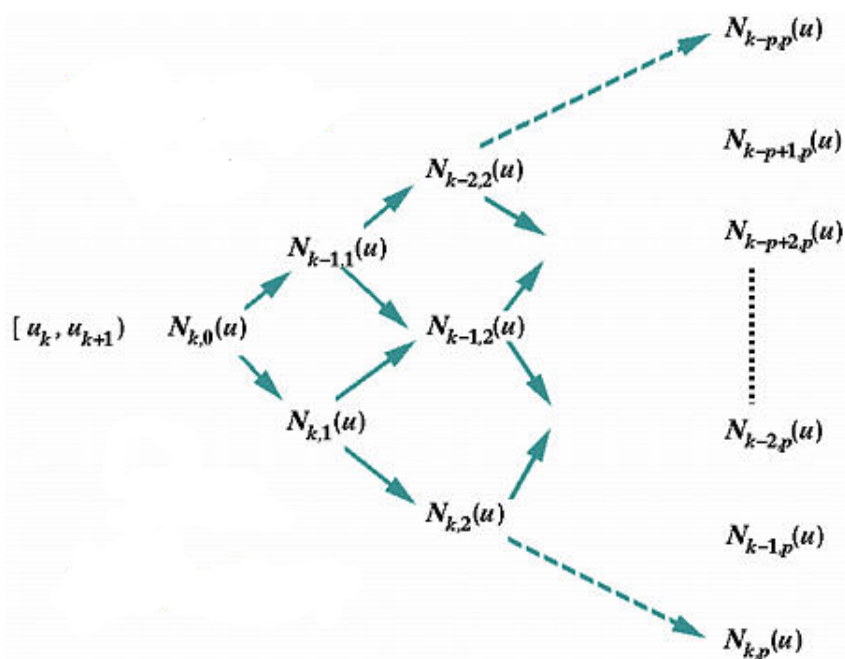
**Vstup:** stupeň  $n$ , index  $i$ , parametr  $t$ , posloupnost uzlů  $T = (t_0, \dots, t_m)$

**Výstup:** koeficienty  $N_0^n(t), N_1^n(t), \dots, N_i^n(t)$  v poli  $N[0], N[1], \dots, N[i]$

```

N[0]=1.0;
for (j=1; j<=n; j++){
  left[j]=t-T[i+1-j]
  right[j]=T[i+j]-t;
  saved = 0.0;
  for (r = 0; r<j; r++){
    temp = N[r]/(right[r+1]+left[j-r]);
    N[r] = saved+right[r+1]*temp;
    saved = left[j-r]*temp;}
  N[j]=saved;}

```



Obrázek 2.11: Výpočet B-spline koeficientů, zdroj Shene (Naposledy navštíveno 11. 12. 2006)

### 2.2.2 B-spline křivky

Termín B-spline se odvozuje od dlouhých ohebných proužků kovu, které v minulosti při své práci používali projektanti při navrhování trupu letadel, lodí a karosérií aut. *Ducks* – olověná závaží – se užívala k upravení tvaru v různých směrech. Kovové proužky byly druhého stupně spojitosti. Matematickým ekvivalentem těchto proužků se staly přirozené kubické spline křivky. Ty ještě nebyly lokálně kontrolovatelné, tj. po jakékoliv změně bylo nutné přepočítat celou křivku. Problém vyřešily B-spline, po částech polynomiální křivky s lokální kontrolou. Bázovými funkcemi pro B-spline se staly B-spline funkce definované v části 2.2.1.



**Definice a vlastnosti B-spline křivek**

*Definice 2.4.* Nechť je dáno  $m+1$  řídicích (kontrolních) bodů  $P_0, P_1, \dots, P_m$ , kde  $P_i \in \mathbb{R}^d$ , uzlový vektor  $\mathbf{t} = (t_0, t_1, \dots, t_{m+n+1})$ . B-spline křivka stupně  $n$  pro řídicí body  $P_i$  a uzlový vektor  $\mathbf{t}$  je definována jako:

$$C(t) = \sum_{i=0}^m P_i N_i^n(t), \quad (2.13)$$

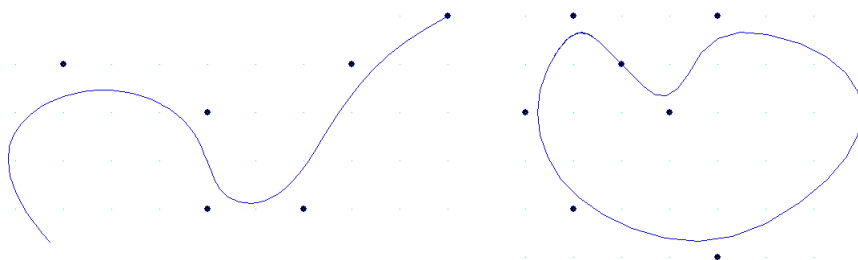
kde  $N_i^n$  jsou báze B-spline funkce podle definice 2.1 v části 2.2.1.

*Definice 2.5.* B-spline křivka se nazývá **uzavřená**, je-li prvních a posledních  $n$  kontrolních bodů shodných, kde  $n$  je stupeň křivky. V ostatních případech se křivka nazývá **otevřená**.

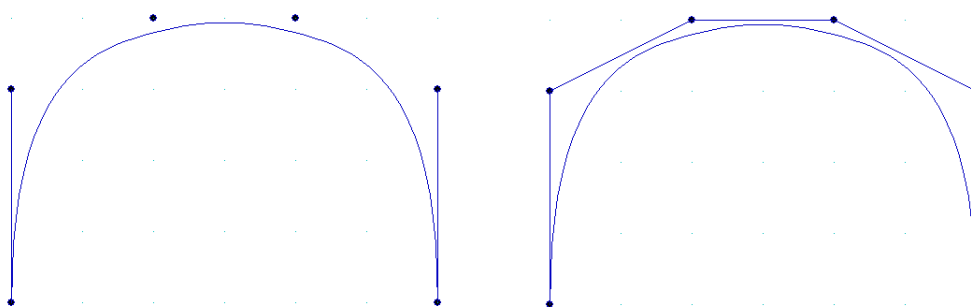
Speciálním případem otevřené křivky je tzv. *sevřená křivka*<sup>1</sup>.

*Definice 2.6.* B-spline křivka se nazývá **sevřená**, jestliže spojnice prvních a posledních dvou řídicích bodů jsou tečnami v krajních bodech křivky.

*Definice 2.7.* Řídicí body  $P_0, P_1, \dots, P_m$  se nazývají **deBoorovy body** a jejich spojení se nazývá **deBoorův<sup>2</sup> polygonální tah**.



Obrázek 2.12: Otevřená a uzavřená B-spline křivka



Obrázek 2.13: Sevřená křivka a její polygonální tah

<sup>1</sup>angl. *clamped curve*

<sup>2</sup>Carl de Boor – emeritní profesor na University of Wisconsin – Madison

## Vlastnosti

### 1. Sevrěná křivka

Pokud je v uzlovém vektoru prvních a posledních  $n + 1$  čísel shodných, prochází křivka prvním a posledním řídicím bodem a jedná se o *sevrěnou* křivku (viz definice 2.6).

### 2. Konvexní obal

B-spline křivka leží celá v konvexním obalu deBoorova polygonálního tahu (viz obr. 2.13).

### 3. Lokální vliv změny uzlového bodu

Změna kontrolního bodu  $P_i$  ovlivní křivku pouze v části odpovídající intervalu  $\langle t_i, t_{i+n+1} \rangle$ .

### 4. Variation diminishing property

Každá přímka protíná deBoorův polygonální tah častěji, než odpovídající B-spline křivku.

### 5. Počet uzlů

Délka uzlového vektoru (ve smyslu definice 2.3) se musí rovnat součtu počtu bodů a stupně křivky plus jedna – řádu křivky. (V definici 2.4 jsou již správně zadané indexy.) Vnitřní uzly uzlového vektoru se mohou opakovat nejvýše  $n$ -krát, kde  $n$  je stupeň křivky.

### 6. Symetrie

Platí:

$$\sum_{i=0}^m N_i^n(t) P_i = \sum_{i=0}^m \bar{N}_{m-i}^n(1-t) P_{m-i},$$

kde funkce  $N_i^n(t)$  je definována pro uzlový vektor  $(t_0, t_1, \dots, t_{n+m+1})$  a funkce  $\bar{N}_{m-i}^n(1-t)$  pro uzlový vektor  $(1-t_{n+m+1}, \dots, 1-t_1, 1-t_0)$ . Tato vlastnost je společná pro všechny bázové funkce, například i pro Bernsteinovy polynomy.

### 7. Spojitost

Křivka  $C(t)$  je  $C^{n-s}$  spojitá v bodě  $C(t_i)$  odpovídajícímu uzlu násobnosti  $s$ , kde  $n$  je stupeň křivky.

### 8. Definiční obor

Definiční obor B-spline křivek je interval  $\langle t_n, t_m \rangle$ .

### 9. Speciální případ B-spline křivek jsou Bézierovy křivky

Pokud je stupeň B-spline křivky roven počtu bodů minus jedna, uzlový vektor má délku  $2(n+1)$  a má tvar

$$\begin{aligned} t_0 &= t_1 = \dots = t_n = 0, \\ t_{n+1} &= t_{n+2} = \dots = t_{2n+1} = 1, \end{aligned}$$

pak B-spline křivka odpovídá Bézierově křivce.

## 10. Afinní invariantnost

B-spline křivky jsou afinně invariantní – stačí zobrazit řídicí body afinní transformací a znovu určit křivku, nemusí se tedy transformovat všechny body na křivce (afinní transformace viz část 1.1.3, str. 8).

**deBoorův algoritmus****Knot insertion – vkládání uzlů**

Stěžejní částí v deBoorově algoritmu je právě vkládání uzlů. Hlavním problémem je vložit uzel tak, aby se nezměnil tvar křivky. Aby byla splněna rovnost z vlastnosti 5 na straně 36 musí se zvýšit stupeň křivky nebo přidat další kontrolní bod. Stupeň křivky by změnil tvar celé křivky, proto je nutné přidat nový řídicí bod.

**Jednoduché přidání uzlu**

Chceme-li vložit uzel  $t$  do uzlového vektoru, musíme udělat několik kroků:

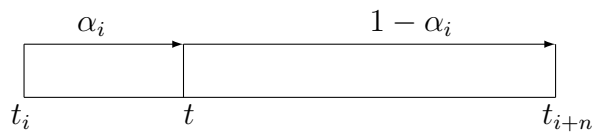
1. Najít interval  $\langle t_k, t_{k+1} \rangle$ , kde zadaný parametr  $t$  leží.
2. Vytvořit novou posloupnost bodů  $Q_k, Q_{k-1}, \dots, Q_{k-n+1}$ , pro kterou platí:

$$\begin{aligned} Q_k &\in P_{k-1}P_k \\ Q_{k-1} &\in P_{k-2}P_{k-1} \\ &\vdots \\ Q_{k-n+1} &\in P_{k-n}P_{k-n+1}. \end{aligned}$$

Body  $Q_i$  se počítají podle vztahů:

$$\begin{aligned} Q_i &= (1 - \alpha_i)P_{i-1} + \alpha_i P_i \\ \alpha_i &= \frac{t - t_{i-1}}{t_i - t_{i-1}}. \end{aligned} \tag{2.14}$$

Geometricky určuje číslo  $\alpha_i$  podíl intervalu  $\langle t_{i-1}, t_i \rangle$ , jak vidíme na schématu 2.14.

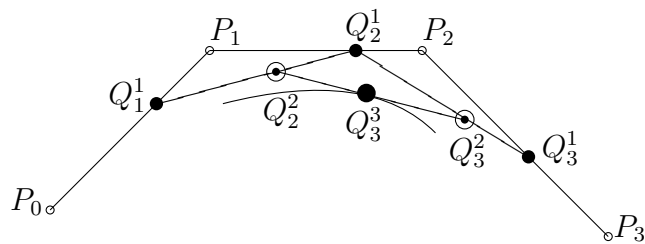


Obrázek 2.14: Geometrický význam vkládání uzlů

3. Vykreslit křivku pro nové kontrolní body  $P_0, P_1, \dots, P_{k-n}, Q_{k-n+1}, \dots, Q_{k-1}, P_k, \dots, P_m$ . (Jejich počet je  $m + 2$ .)

**Násobné vložení uzlu**

Situace je analogií předchozího případu. Změníme pouze výpočetní formule. Uzel  $t$  vložíme do uzlového vektoru  $h$ -krát.



Obrázek 2.15: Knot insertion

Body  $Q_i$  lze vyjádřit jako:

$$\begin{aligned} Q_{i,h} &= (1 - \alpha_{i,h})P_{i-1,h-1} + \alpha_{i,h}P_{i,h-1}, \\ \alpha_{i,h} &= \frac{t - t_i}{t_{i+n-(h-1)} - t_i}. \end{aligned} \quad (2.15)$$

### Algoritmus výpočtu

**Věta 2.4.** *Je-li násobnost uzlu  $t$  rovna stupni křivky, pak bod  $C(t)$  je bodem křivky.*

Z následující věty vyplývá algoritmus výpočtu. Vezmeme-li libovolnou hodnotu uzlu a provedeme algoritmus *knot insertion* tolikrát, kolik je stupeň křivky, dostáváme bod na křivce. Je nutné určit, zda se zvolený uzel již v uzlovém vektoru nevyskytuje a o tento počet snížit počet opakování *knot insertion*. Algoritmus výpočtu můžeme zapsat následovně:

**Vstup:** stupeň křivky  $n$ , řídicí body, uzlový vektor, parametr  $t$

**Výstup:** bod křivky  $C(t)$

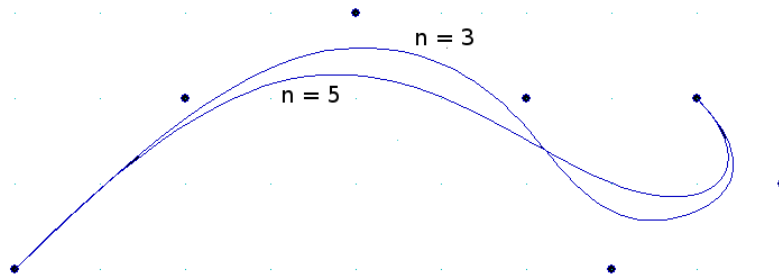
1. Ověřit podmínky pro vstupní data.
2. Nalézt interval  $\langle t_k, t_{k+1} \rangle$ , kde zadané  $t$  leží.
3. Pokud se  $t = t_k$ , je nutné určit, s jakou násobností  $s$  se zde uzel vyskytuje.
4. Opakované vložení uzlu – *knot insertion*  $(n - s)$ -krát.

### Jak pracovat s B-spline křivkami

Pro práci s B-spline křivkami máme několik možností, jak měnit jejich tvar. Lze změnit stupeň křivky (viz část 1), upravit polohu řídicích bodů (viz část 2) a dále změnit tvar uzlového vektoru (viz část 3).

#### 1. Vliv stupně křivky

Změna stupně křivky je globální – mění tvar celé křivky. Obvykle se používá stupeň dva nebo tři. Při stupni jedna se jedná o lomenou čáru. Na obr. 2.16 vidíme B-spline křivku pro stupeň  $n = 3$  a  $n = 5$ . Čím nižší je stupeň, tím více se křivka přimyká k řídicím bodům. Pro stupeň jedna se jedná o lomenou čáru procházející zadanými body.



Obrázek 2.16: Vliv stupně na tvar křivky

## 2. Vliv kontrolních bodů

Při výpočtu bodu křivky se používá výrazu

$$C(t) = \sum_{i=0}^m P_i N_i^n(t),$$

kde  $N_i^n(t)$  jsou bazové B-spline funkce z definice 2.1, str. 24.

Funkce  $N_i^n(t)$  určuje, jak silně bude kontrolní bod  $P_i$  ovlivňovat křivku v čase  $t$ . Slovo čas se někdy používá jako výraz pro uzel, neboť procházením uzlového vektoru se postupně – jakoby v čase – dostávají hodnoty křivky. Po dosazení parametru  $t$  do odpovídající bazové funkce dostáváme, jak silně bude působit řídicí bod na tvar křivky.

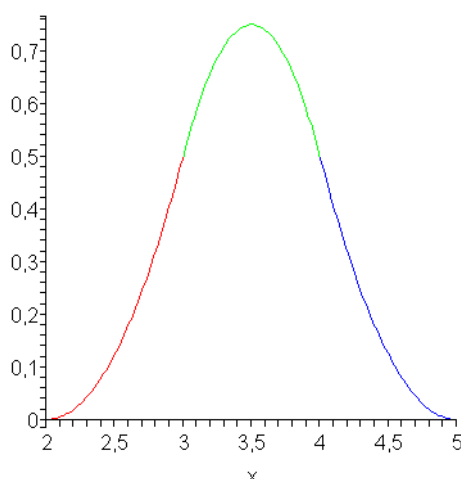
*Příklad 2.5.* Na obr. 2.17 je vykreslena bazová funkce stupně dva na intervalu  $\langle 2, 5 \rangle$ . Pro čas  $t = 3.5$  je vidět, že hodnota  $N_2^2(3.5)$  pro bod  $P_2$  je přibližně 0.75, tzn. že bod  $P_2$  má 75-ti procentní vliv na výsledný bod.

Vliv polohy bodu je pouze lokální. Bod  $P_i$  je v součinu s bazovou funkcí  $N_i^n(t)$ , tedy funkcí, která ovlivňuje křivku pro parametr  $t$  z intervalu  $\langle t_i, t_{i+n+1} \rangle$  podle vlastnosti 3 (str. 36). Na tomto intervalu nalezneme nenulové funkce  $N_{i-n}^n, N_{i-n+1}^n, \dots, N_{i+n}^n$ , které ovlivní tvar křivky mezi body  $P_{i-n}, \dots, P_{i+n}$ .

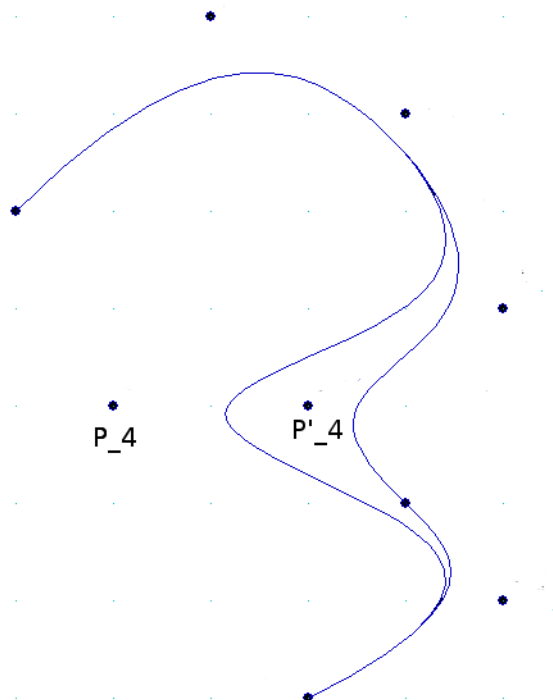
*Příklad 2.6.* Je dána B-spline křivka stupně  $n = 3$  body  $P_0, P_1, \dots, P_7$  a ekvidistantním uzlovým vektorem  $(0, 1, \dots, 11)$ .

Na obr. 2.18 je vidět změna křivky po posunutí bodu  $P_4$  do bodu  $P'_4$ . Změní se hodnota funkce  $N_4^3(t)$  a tvar křivky bude jiný pro hodnoty parametru  $t$  z intervalu  $\langle 4, 8 \rangle$ . Na tomto intervalu působí také funkce  $N_1^3, N_2^3, \dots, N_7^3$  a tvar křivky se změní mezi body  $P_1, P_2, \dots, P_7$ .

Pro sestrojení uzavřené křivky stupně  $n$  je nutné a stačí, aby se přidalo  $n + 1$  prvních bodů posloupnosti za poslední kontrolní bod  $P_m$ . Tedy dostáváme novou



Obrázek 2.17: Bázová funkce  $N_2^2(t)$  pro ekvidistantní uzlový vektor  $(0, 1, \dots, 7)$



Obrázek 2.18: Vliv polohy bodu na tvar křivky

posloupnost řídicích bodů

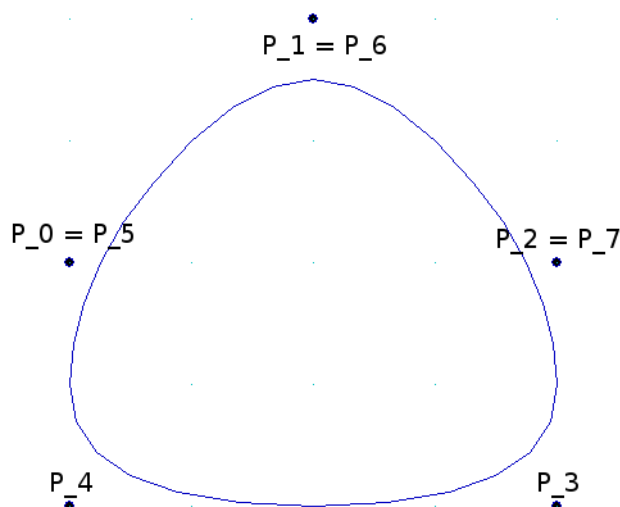
$$P_0, P_1, \dots, P_m, P_{m+1} = P_0, P_{m+2} = P_1, \dots, P_{m+n+1} = P_{n+1}.$$

Uzlový vektor musí být ekvidistantní bez násobných uzlů.

*Příklad 2.7.* Uzavřená křivka stupně  $n = 2$  pro řídicí body

$$P_0, P_1, \dots, P_5, P_6 = P_0, P_7 = P_1, P_8 = P_2$$

a ekvidistantní uzlový vektor  $(0, 1, \dots, 10)$ . Výsledek je zobrazen na obr. 2.19.



Obrázek 2.19: Uzavřená B-spline křivka

### 3. Vliv uzlového vektoru

B-spline křivky se dělí na uniformní a neuniformní, podle toho, zda je uzlový vektor ekvidistantní ve smyslu definice 2.2. K pochopení vlivu uzlového vektoru je nutné si uvědomit, že každá bázeová funkce, která se počítá se vstupním uzlovým vektorem, je spjata s odpovídajícím řídicím bodem. Obecně pro bod  $P_i$  je to bázeová funkce  $N_i^n(t)$ . Je nutné se podívat na grafy bázeových funkcí  $N_i^n$  vykreslených v části 2.2.1.

Vezměme ekvidistantní uzlový vektor  $(0, 1, 2, 3, 4, 5, 6, 7)$  se stupněm křivky rovným dvěma. Pro parametr  $t = 4.1$  dostáváme hodnoty:

$$N_2^2 = 0.405$$

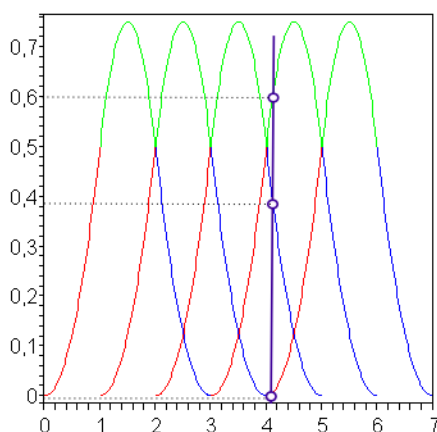
$$N_3^2 = 0.59$$

$$N_4^2 = 0.005$$

Znamená to, že v čase  $t = 4.1$  má bod  $P_2$  vliv 40,5% na tvar křivky, bod  $P_3$  má vliv 59% a bod  $P_4$  0,5%.

Pro ekvidistantní uzlový vektor  $(0, 1, 2, 3, 4, 5, 6, 7)$  jsou výsledné funkce pro stupeň 2 vykresleny na obr. 2.20. Úsečka ukazuje hodnoty bázeových funkcí pro parametr  $t = 4.1$ .

Pro ekvidistantní uzlový vektor vidíme, že všechny křivky jednotlivých funkcí jsou totožné, až na posunutí o jednotku. Proto se vliv mění plynule a rovnoměrně. Pokud chceme upravit tvar křivky, je nutné použít neekvidistantní uzlový vektor.



Obrázek 2.20: B-spline funkce stupně dva pro vektor  $(0, 1, 2, 3, 4, 5, 6, 7)$ .

Vezměme uzlový vektor  $(0, 1, 2, 3.8, 4, 4.2, 6, 7)$  a báze funkce opět druhého stupně. Položíme parametr  $t = 4.1$  a dostáváme hodnoty:

$$N_2^2 = 0.125$$

$$N_3^2 = 0.85$$

$$N_4^2 = 0.025$$

Znamená to, že v čase  $t = 4.1$  má bod  $P_2$  vliv 12,5% na tvar křivky, bod  $P_3$  má vliv 85% a bod  $P_4$  2,5%. Je zřejmé, že se výrazně zvýšil vliv bodu  $P_3$ . Na obr. 2.21 jsou zobrazeny báze funkce s vyznačenými hodnotami pro parametr  $t = 4.1$ .

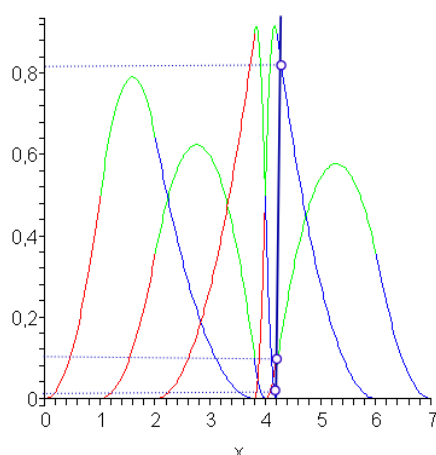
*Poznámka 12.* Součet výše vypočítaných hodnot je roven jedné podle vlastnosti 4, str. 33.

Výsledné křivky pro oba výše uvedené uzlové vektory jsou vykresleny na obr. 2.22 a 2.23.

Zmenšení intervalu tedy způsobí větší vliv daného bodu, ale na menší části křivky. Křivka se k bodu  $P_3$  přiblíží, ale na okolní body bude mít tento bod už menší vliv. Je to patrné z obr. 2.8, kde má graf funkce strmý vzestup i sestup na intervalu  $\langle 3.8, 4.2 \rangle$ .

Dalším případem je požadavek, aby křivka procházela prvním a posledním řídicím bodem. Jedná se o tzv. *sevřenou* křivku. (viz definice 2.6). Podle vlastnosti 1 v části 2.2.2 stačí, aby se prvních a posledních  $n+1$  hodnot uzlového vektoru opakovalo. Pro uzlový vektor  $(0, 0, 0, 1, 2, 3, 3, 3)$  jsou báze funkce druhého stupně opět vypočítány a zobrazeny v části 2.2.1 na obr. 2.9 a 2.10.

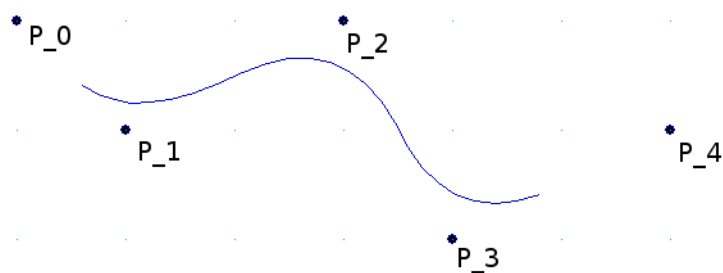




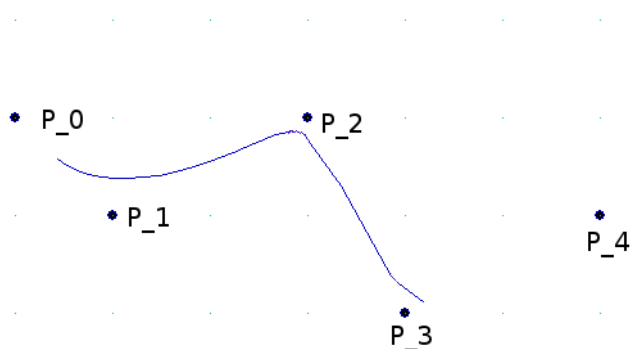
Obrázek 2.21: B-spline funkce stupně dva pro vektor  $(0, 1, 2, 3.8, 4, 4.2, 6, 7)$ .

Pro  $t = 0$  dostáváme  $N_0^2 = 1$ , tedy bod  $P_0$  má vliv 100%. Pro  $t = 1$  dostáváme  $N_4^2 = 1$ , tedy bod  $P_4$  má vliv 100%.

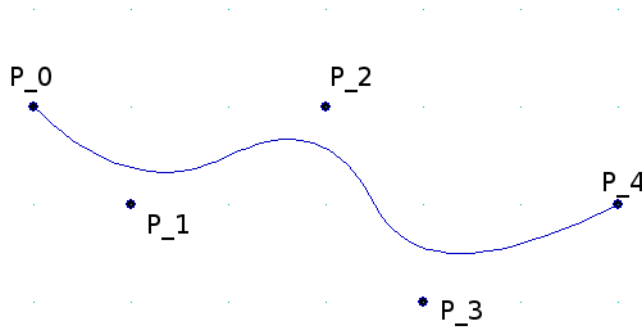
Výslednou křivku vidíme na obr. 2.24.



Obrázek 2.22: Křivka pro ekvidistantní uzlový vektor  $(0, 1, 2, 3, 4, 5, 6, 7)$ .



Obrázek 2.23: Křivka pro neekvidistantní uzlový vektor  $(0, 1, 2, 3.8, 4, 4.2, 6, 7)$ .



Obrázek 2.24: Křivka stupně dva pro uzlový vektor  $(0, 0, 0, 1, 2, 3, 3, 3)$ .

### Obálka B-spline křivek při změně uzlového vektoru

Mějme danu B-spline křivku definovanou v části 2.2.2 (str. 35). Pokud budeme jeden z vnitřních uzlů spojitě měnit mezi jeho okolními uzly, dostaneme jednoparametrickou soustavu křivek stupně o jedno nižší. Obecně:

**Věta 2.8.** *Měníme-li hodnotu uzlu  $u_i \in \langle u_{i-1}, u_{i+1} \rangle$  B-spline křivky  $c(t)$  stupně  $n$  ( $n > 1$ ), dostáváme jednoparametrickou soustavu B-spline křivek danou rovnicí:*

$$c(u, u_i) = \sum_{j=0}^m P_j N_j^n(u, u_i), \quad \text{kde } u \in \langle u_n, u_{m+1} \rangle. \quad (2.16)$$

Tato soustava má obálku, která je B-spline křivkou stupně  $n - 1$  a lze ji psát jako:

$$h(v) = \sum_{j=i-n+2}^{i-1} P_j N_j^{n-1}(v), \quad \text{kde } v \in \langle v_{i-1}, v_i \rangle, \quad (2.17)$$

kde uzlové hodnoty jsou:

$$v_r = \begin{cases} u_r & \text{pro } r < i \\ u_{r+1} & \text{jinak,} \end{cases}$$

Omezíme-li se na kubické křivky, je obálkou dané soustavy křivka stupně druhého, tedy parabolický oblouk. Měníme-li uzel  $u_{j+1} \in \langle u_j, u_{j+2} \rangle$  dostáváme soustavu křivek

$$c(u, u_{j+1}) = \sum_{l=0}^m P_l N_l^3(u, u_{j+1}), \quad \text{kde } u \in \langle u_3, u_{m+1} \rangle \quad (2.18)$$

s uzly  $u_0, u_1, \dots, u_{m+4}$ . Její obálkou je parabolický oblouk

$$h_j(v) = \sum_{l=j-2}^j P_l N_l^2(v), \quad \text{kde } v \in \langle v_{j-1}, v_j \rangle \quad (2.19)$$

s uzly  $v_{j-2} = u_{j-2}$ ,  $v_{j-1} = u_{j-1}$ ,  $v_j = u_j$ ,  $v_{j+2} = u_{j+3}$ ,  $v_{j+3} = u_{j+4}$ .

Smyslem této věty je, že je možné zjednodušit práci s uzlovým vektorem pro B-spline křivku na práci s parabolickými oblouky, tedy zredukovat problém stupně tři na problém stupně dva.

Další možností úpravy křivky je určit bod, kterým má daná křivka procházet. K tomu se opět využije obálka soustavy definovaná výše. Obálka se změní tak, aby procházela daným bodem, poté se určí uzly křivky a na závěr zmodifikujeme původní křivku tak, aby se dotýkala obálky v námi požadovaném bodě.

Tato úloha nemá řešení vždy, je nutné stanovit oblast, ve které zvolený bod může ležet. Přepisem parabolického oblouku z rovnice 2.19 pomocí Bernsteinových polynomů (parabolický oblouk je možné přepsat jako Bézierovu křivku) dostáváme dvou parametrickou soustavu parabol, které ohraničují oblast, kde můžeme umístit nový bod (podrobněji v Juhasz – Hoffmann (2004), Juhasz – Hoffmann (2001)).

Další možností pro úpravu tvaru je stanovit pro danou křivku tečnu s bodem dotyku. Aby úloha byla řešitelná, musí zvolená tečna protínat hrany  $d_{j-2}, d_{j-1}$  a  $d_{j-1}, d_j$  kontrolního polygonu soustavy parabol 2.19.

*Poznámka 13.* Tato věta a její aplikace mají pouze teoretický význam, v praxi se pomocí křivek s nižším stupněm napracuje.

#### 4. Napojování B-spline křivek

Je dána B-spline křivka a chceme na ni hladce navázat další křivku, začínající v posledním bodě této křivky.

Křivka je dána  $m+1$  řídicími body, stupněm  $n$  a dále váhovým a uzlovým vektorem. Uzlový vektor má prvních a posledních  $n+1$  hodnot stejných, křivka tedy začíná v bodě  $P_0$  a končí v bodě  $P_m$ . Podle vlastnosti 1 v části 2.2.2 na straně 35 jsou spojnice prvních a posledních dvou bodů tečnami křivky. Pokud chceme spojitě navázat další křivku (řídicí body  $Q_i$ ), musí splňovat následující podmínky:

- (a) jedná se o *sevrženou* křivku (začíná v prvním řídicím bodě a končí v posledním řídicím bodě),
- (b)  $P_m = Q_0$ ,
- (c) body  $P_{m-1}, P_m, Q_1$  leží na jedné přímce.

V bodě  $P_m = Q_0$  mají tyto křivky společnou tečnu  $P_{m-1}Q_1$  a jsou hladce navázané.

### 2.2.3 NURBS křivky

Jak již bylo řečeno, jsou NURBS (Non Uniform Rational B-Spline) zobecněním B-spline křivek. V části týkající se B-spline křivek jsme uvažovali neuniformní (Non-Uniform) uzlový vektor. Racionalita (Rational) znamená, že ke každému bodu je přidána jeho váha, tj. s jakou *silou* bude bod působit na tvar křivky. Aby bylo možné pracovat se stejným algoritmem výpočtu jako pro B-spline křivky, je nutné zavést projektivní rozšíření prostoru a homogenní souřadnice.

Přidáním vah dostáváme další způsob ovládání tvaru křivky, ale také možnost zadat kuželosečkové oblouky – eliptický, parabolický a hyperbolický. Dosud známe křivky nejsou schopny je zadat přesně. Vzhledem k tomu jsou NURBS křivky velmi užitečné v technické praxi, kde jsou kuželosečkové oblouky požadovány.

V následující části je vysvětleno odvození homogenních souřadnic a podrobně popsáno odvození jednotlivých kuželosečkových oblouků. Na závěr jsou uvedeny příklady zadání kuželoseček.

## Definice a vlastnosti

*Definice 2.8.* Mějme dáno  $m + 1$  kontrolních bodů  $P_i$ ,  $m + 1$  kladných reálných čísel  $w_i$  nazývaných váhy, dále stupeň křivky  $n$  a uzlový vektor  $\mathbf{t} = (t_0, t_1, \dots, t_{n+m+1})$ . NURBS křivka je definována jako:

$$C(t) = \frac{\sum_{i=0}^m w_i P_i N_i^n(t)}{\sum_{i=0}^m w_i N_i^n(t)}, \quad (2.20)$$

kde  $t \in \langle t_n, t_{m+1} \rangle$ .

Označíme-li

$$R_i^n(t) = \frac{w_i N_i^n(t)}{\sum_{i=0}^m w_i N_i^n(t)}, \quad (2.21)$$

pak lze NURBS křivku zapisovat jako

$$C(t) = \sum_{i=0}^m P_i R_i^n(t). \quad (2.22)$$

## Vlastnosti

Vlastnosti B-spline křivek uvedené v části 2.2.2 na straně 35 platí i pro polynomy  $R_i^n(t)$ . Jediným rozdílem je invariantnost zobrazení. NURBS křivky jsou oproti B-spline křivkám projektivně invariantní (viz teorie uvedená v části 1.1.3, str. 8).

## Racionalita

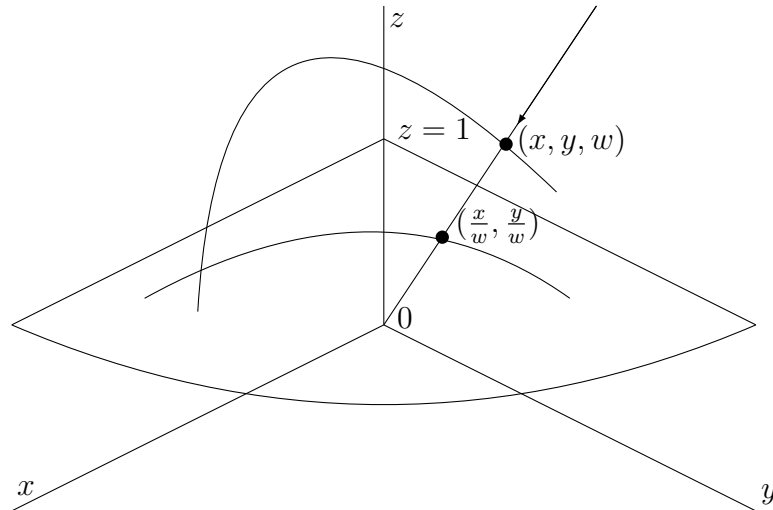
Přidání vah k bodům se řeší projektivním rozšířením prostoru. Definice projektivního rozšíření je uvedena v kapitole 1.

Každému řídicímu bodu  $P_i$  je přiřazena váha  $w_i$ . Ta určuje s jakou *silou* se křivka bude přibližovat k danému bodu. Pokud mají všechny body váhu konstantní, jedná se o B-spline křivku. V opačném případě je nutné pracovat v projektivním rozšíření prostoru.

Projektivní rozšíření roviny do prostoru je názorné (viz obr. 2.25). Body NURBS křivky v rovině jsou dány dvěma souřadnicemi a váhou. Homogenizace se provede vynásobením souřadnic bodů vahou a přidáním třetí souřadnice, která bude rovna váze samotné. Například bod  $(6, 3, 3)$  určuje rovinný bod  $(2, 1)$  s váhou 3. Výpočet bodu křivky se provede pro zhomogenizované body a výsledné body se promítnou zpět do roviny  $z = 1$ .

Matematicky lze toto promítnutí zapsat podílem:

$$C(t) = \frac{\sum_{i=0}^n w_i P_i N_i^n(t)}{\sum_{i=0}^n w_i N_i^n(t)}. \quad (2.23)$$



Obrázek 2.25: Projektivní rozšíření roviny

Vysvětleme, jak z rovnice B-spline křivky uvedené v definici 2.4 dostaneme vyjádření (2.23) pro NURBS křivku.

Máme dáno  $m+1$  řídicích bodů  $P_i$ ,  $m+1$  vah  $w_i$ , dále stupeň křivky  $n$  a uzlový vektor  $\mathbf{t} = (t_0, t_1, \dots, t_{n+m+1})$ . Provedeme projektivní rozšíření prostoru, ve kterém pracujeme podle předchozí úvahy. Každou souřadnici bodu vynásobíme příslušnou vahou a jako čtvrtou souřadnici přidáme váhu. Dostáváme tedy:

$$(x_i, y_i, z_i) \longrightarrow (w_i x_i, w_i y_i, w_i z_i, w_i)$$

Nyní můžeme v rozšířeném prostoru vypočítat bod neuniformní B-spline křivky pro libovolný parametr  $t$ :

$$C(t) = \sum_{i=0}^m (w_i x_i, w_i y_i, w_i z_i, w_i) N_i^n(t) = \sum_{i=0}^m w_i P_i N_i^n(t) = (x^W, y^W, z^W, w^W) = W. \quad (2.24)$$

Bod  $W$  je obecným bodem NURBS křivky v rozšířeném prostoru, má tedy čtyři souřadnice. Nyní uděláme zpětnou projekci do trojrozměrného prostoru. Pro případ roviné křivky je to promítnutí bodů do roviny  $z = 1$ . V případě prostorové křivky to můžeme popsat jako promítnutí do příslušné nadroviny. Matematicky to znamená, že poslední souřadnice musí být rovna jedné, tedy všechny čtyři souřadnice výsledného bodu  $W$  vydělíme poslední souřadnicí.

$$(x^W, y^W, z^W, w^W) \longrightarrow \left( \frac{x^W}{w^W}, \frac{y^W}{w^W}, \frac{z^W}{w^W}, 1 \right)$$

kde hodnotu  $w^W$  dostaneme jako:

$$w^W = \sum_{i=0}^m w_i N_i^n(t), \quad (2.25)$$

Pokud vyjádříme rovnici obecného bodu NURBS křivky pomocí rovnic (2.24) a (2.25) dostáváme známé vyjádření z definice 2.20. Součin  $w_i P_i$  v čitateli znamená zhomogenizované první tři souřadnice bodu –  $(w_i x_i, w_i y_i, w_i z_i)$ .

$$C(t) = \frac{\sum_{i=0}^n w_i P_i N_i^n(t)}{\sum_{i=0}^n w_i N_i^n(t)}.$$

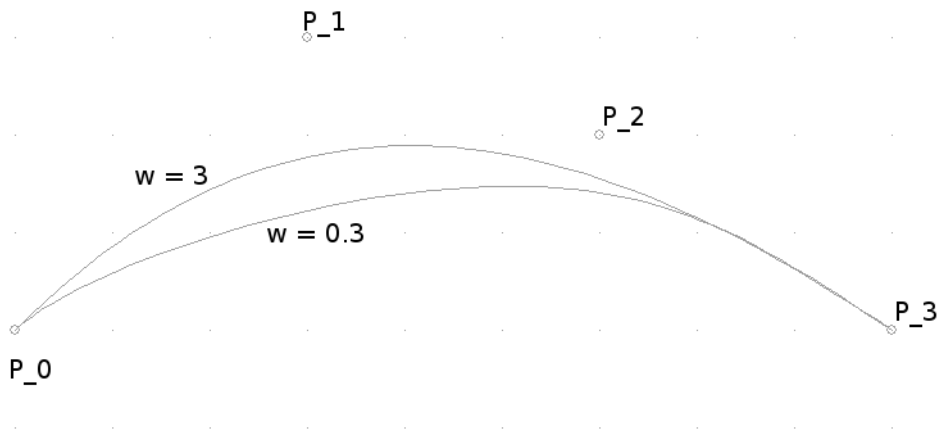
### Vliv váhy na tvar křivky

Váhový vektor tvoří posloupnost kladných reálných čísel. Délka váhového vektoru je stejná jako počet řídicích bodů, tedy každému bodu je jednoznačně přiřazena jeho váha. Pokud mají všechny body stejnou váhu, je váhový vektor posloupnost  $(1, 1, 1, \dots, 1)$ .

Pokud váha daného bodu leží v intervalu  $(0, 1)$  znamená to, že příslušný bod má menší vliv. S váhou větší než 1 se zvyšuje i vliv daného řídicího bodu a výsledná křivka se k němu více přibližuje.

*Příklad 2.9.* Křivka stupně 3 je dána čtyřmi řídicími body  $P_0, P_1, P_2, P_3$ , uzlovým vektorem  $(0, 0, 0, 0, 1, 1, 1, 1)$  a váhovým vektorem  $(w_0, w_1, w_2, w_3) = (1, 1, 1, 1)$ .

Změníme váhu bodu  $P_1$  na  $w_1 = 3$  a  $w_1 = 0.3$ . Výsledné křivky jsou zobrazeny na obr. 2.26.



Obrázek 2.26: Vliv váhy na tvar křivky

### Odvození kuželosečkových oblouků

Jak již bylo zmíněno v úvodu, velkou výhodou NURBS křivek je přímé zadání kuželosečkových oblouků. Část důkazu je převzata z Shene (Naposledy navštíveno 11. 12. 2006), pro odvození kružnice jsem použila vlastní metodu.

Základní rovnice kuželosečky je dána tvarem

$$Ax^2 + 2Hxy + By^2 + 2Gx + 2Fy + C = 0, \quad (2.26)$$

kde alespoň jedno z  $a, b, h$  je nenulové.

Kuželosečkový oblouk se zadává jako NURBS křivka následujícím způsobem:

1. stupeň křivky 2,
2. tři řídicí body  $P_0, P_1, P_2$ ,
3. uzlový vektoru  $(0, 0, 0, 1, 1, 1)$ ,
4. váhový vektor  $(1, w_1, 1)$ .

V rovnici 2.26 vystupuje šest neznámých. Je-li  $C \neq 0$ , pak lze celou rovnici vydělit  $C$  a počet neznámých se sníží o jednu. Pro  $C = 0$  je předchozí úvaha splněna. Základní rovnici lze přepsat do tvaru:

$$f : ax^2 + 2hxy + by^2 + 2gx + 2fy + 1 = 0 \quad (2.27)$$

Je nutné najít pět podmínek, které by jednoznačně určily parametry této rovnice.

Oblouk prochází krajními body  $P_0 = [x_0, y_0]$ ,  $P_2 = [x_2, y_2]$ , z čehož vyplývá tvar prvních dvou podmínek:

$$f(P_0) = ax_0^2 + 2hx_0y_0 + by_0^2 + 2gx_0 + 2fy_0 + 1 = 0 \quad (2.28)$$

$$f(P_2) = ax_2^2 + 2hx_2y_2 + by_2^2 + 2gx_2 + 2fy_2 + 1 = 0 \quad (2.29)$$

Ze zadání uzlového vektoru vyplývá, že spojnice bodů  $P_0P_1$  a  $P_2P_1$  jsou tečnami v krajních bodech  $P_0, P_2$ . Směrnice tečen procházejících danými body jsou:

$$k_1 = \frac{y_1 - y_0}{x_1 - x_0}, \quad (2.30)$$

$$k_2 = \frac{y_1 - y_2}{x_1 - x_2}. \quad (2.31)$$

Směrnice tečen dostaneme také z rovnice 2.27 jako podíl parciálních derivací.

$$\frac{\partial f}{\partial x} = 2ax + 2hy + 2g = ax + hy + g,$$

$$\frac{\partial f}{\partial y} = 2by + 2hx + 2f = by + hx + f.$$

Směrnice v bodech  $P_0, P_2$  jsou tedy

$$k_1 = -\frac{ax_0 + hy_0 + g}{by_0 + hx_0 + f}, \quad (2.32)$$

$$k_1 = -\frac{ax_2 + hy_2 + g}{by_2 + hx_2 + f}. \quad (2.33)$$

Porovnáním vztahů 2.30, 2.32 a 2.31, 2.33 dostáváme další dvě podmínky:

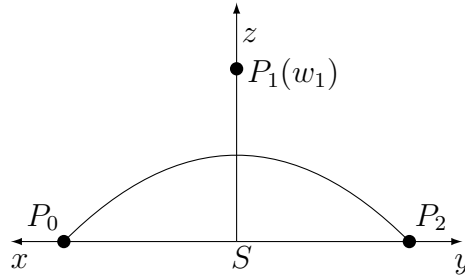
$$\frac{y_1 - y_0}{x_1 - x_0} = \frac{ax_0 + hy_0 + g}{by_0 + hx_0 + f}, \quad (2.34)$$

$$\frac{y_1 - y_2}{x_1 - x_2} = -\frac{ax_2 + hy_2 + g}{by_2 + hx_2 + f}. \quad (2.35)$$

Pro pátou podmínku je nutné použít bod  $P_1$  a jeho váhu  $w_1$ . Vezmeme NURBS křivku danou definicí 2.8 (str. 46) a vyjádříme její parametrickou rovnici vzhledem k zadaným hodnotám obecného oblouku:

$$c(t) = \frac{(1-t^2)P_0 + 2t(1-t)w_1P_1 + t^2P_2}{(1-t)^2 + 2t(1-t)w_1 + t^2} \quad (2.36)$$

Bez újmy na obecnosti vezmeme případ, kdy krajní body leží na ose  $x$  souměrně podle počátku, tedy  $P_0 = -P_2$ . Bod  $S$  je středem soustavy souřadnic. Situaci vidíme na obr. 2.27.



Obrázek 2.27: Umístění oblouku do souřadných os

Vyjádřením  $c(0.5)$  dostáváme:

$$c(0.5) = \frac{w_1}{1 + w_1} P_1.$$

Označíme  $c(0.5) = X$ . Poněvadž body  $S, X, P_1$  leží na jedné přímce, můžeme psát:

$$\frac{|SX|}{|SP_1|} = \frac{w_1}{1 + w_1}. \quad (2.37)$$

Pokud je podíl roven hodnotě 0.5, jedná se o **parabolu**, neboť subtangenta paraboly je vrcholem půlena. V tomto případě bod  $X$ , tedy vrchol paraboly, leží ve středu úsečky  $SP_1$ .

*Závěr 1:* Zvolíme-li váhu  $w_1 = 1$ , výsledná NURBS křivka je **parabola**.

Nyní je nutné rozlišit případ hyperboly a elipsy.

Označme  $D$  další průsečík přímky  $P_1X$  s kuželosečkou (obr. 2.28). Pro kuželosečky platí následující poměr:

$$\frac{|P_1X|}{|SX|} = \frac{|DP_1|}{|DS|}. \quad (2.38)$$



Vyjádříme podíl  $\frac{|SX|}{|SP_1|}$  :

$$\frac{|SX|}{|SP_1|} = \frac{|SX|}{|SX| + |P_1X|} = \frac{1}{1 + \frac{|P_1X|}{|SX|}} = \frac{1}{1 + \frac{|DP_1|}{|DS|}}. \quad (2.39)$$

Z rovnic 2.37 a 2.39 můžeme psát:

$$\frac{|SX|}{|SP_1|} = \frac{1}{1 + \frac{|DP_1|}{|DS|}} = \frac{w_1}{1 + w_1}. \quad (2.40)$$

V případě elipsy je  $DP_1 > DS$  (podle obr. 2.28), tedy  $\frac{|DP_1|}{|DS|} > 1$ , z toho vyplývá:

$$\frac{1}{1 + \frac{|DP_1|}{|DS|}} < \frac{1}{2}, \quad (2.41)$$

tedy podle rovnice 2.40 je

$$\frac{w_1}{1 + w_1} < \frac{1}{2} \quad (2.42)$$

a z řešení této nerovnice vidíme, že  $w_1$  musí být menší než jedna.

*Závěr 2* Zvolíme-li váhu  $w_1 < 1$  je výsledná křivka **elipsa**.

V případě hyperboly je  $|DP_1| < |DS|$  (podle obr. 2.28), tedy  $\frac{|DP_1|}{|DS|} < 1$ . Z toho vyplývá:

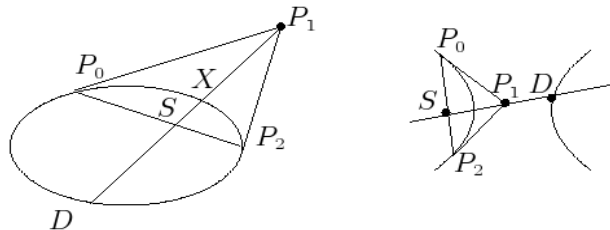
$$\frac{1}{1 + \frac{|DP_1|}{|DS|}} > \frac{1}{2}, \quad (2.43)$$

tedy podle rovnice 2.40 je

$$\frac{w_1}{1 + w_1} > \frac{1}{2} \quad (2.44)$$

a z řešení této nerovnice vyplývá, že  $w_1$  musí být větší než jedna.

*Závěr 3* Zvolíme-li váhu  $w_1 > 1$  je výsledná křivka **hyperbola**.

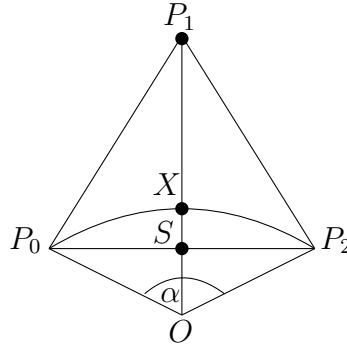


Obrázek 2.28: Odvození vah pro elipsu a hyperbolu

### Kružnicové oblouky

Kružnice je zvláštním případem elipsy, proto musíme určit, pro jakou váhu  $w_1$  se daný eliptický oblouk stává kružnicovým obloukem. Znamená to najít poměr:

$$\frac{|SX|}{|SP_1|} = \frac{w_1}{1 + w_1}. \quad (2.45)$$



Obrázek 2.29: Odvození váhy pro kružnicové oblouky

Podle obr. 2.29 vyjádříme délky úseček  $SX$  a  $SP_1$ . Označme středový úhel kružnicového oblouku  $\alpha$ .

Z pravoúhlého  $\triangle OSP_0$  plyne:

$$\cos \frac{\alpha}{2} = \frac{|OS|}{r} \rightarrow |OS| = r \cos \frac{\alpha}{2}. \quad (2.46)$$

V pravoúhlém  $\triangle OP_1P_0$  platí:

$$\cos \frac{\alpha}{2} = \frac{r}{|OP_1|} \rightarrow |OP_1| = \frac{r}{\cos \frac{\alpha}{2}}, \quad (2.47)$$

$$|SX| = |OX| - |OS| = r - r \cos \frac{\alpha}{2} \quad (2.48)$$

$$|SP_1| = |OP_1| - |OS| = \frac{r}{\cos \frac{\alpha}{2}} - r \cos \frac{\alpha}{2} = \frac{r(1 - \cos^2 \frac{\alpha}{2})}{\cos \frac{\alpha}{2}}. \quad (2.49)$$

Hledaný poměr je

$$\frac{|SX|}{|SP_1|} = \frac{r(1 - \cos \frac{\alpha}{2})}{r(1 - \cos^2 \frac{\alpha}{2}) / \cos \frac{\alpha}{2}} = \frac{\cos \frac{\alpha}{2}}{1 + \cos \frac{\alpha}{2}} = \frac{w_1}{1 + w_1}. \quad (2.50)$$

Z toho vyplývá řešení:

$$w_1 = \cos \frac{\alpha}{2}. \quad (2.51)$$

**Závěr 4:** Pro **kružnicový oblouk** je váha  $w_1$  rovna kosinu velikosti poloviny středového úhlu daného oblouku.

**Zadání kuželoseček****Kružnicové oblouky o středovém úhlu  $\alpha$** 

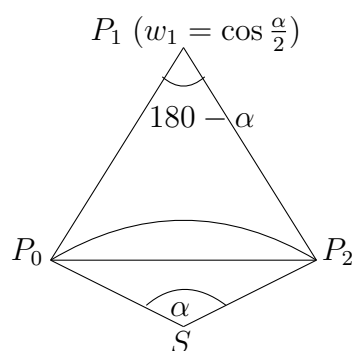
Řídící body tvoří rovnoramenný trojúhelník  $P_0, P_1, P_2$ , kde  $|P_0P_1| = |P_1P_2|$ .

Stupeň křivky: 2 (řád 3)

Uzlový vektor:  $(0, 0, 0, 1, 1, 1)$  – křivka musí začínat a končit v řídicím bodě

Váhový vektor:  $(1, \cos \frac{\alpha}{2}, 1)$

Na obr. 2.30 jsou znázorněny vztahy mezi úhly a výsledný oblouk.



Obrázek 2.30: Zadání kružnicového oblouku

**Kružnice**

Podle předchozí části lze nakreslit kružnici jako čtyři kružnicové oblouky o středových úhlech  $\alpha = 90$  stupňů (obr. 2.2.3). Pro kružnici se středem  $(0, 0)$  a poloměrem  $r = 1$  tvoří řídicí body čtverec a vezmeme:

Řídící body:

$$[1, 0], [1, 1], [0, 1], [-1, 1], [-1, 0], [-1, -1], [0, -1], [1, -1], [1, 0]$$

Stupeň křivky: 2 (řád 3)

Uzlový vektor:  $(0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1)$

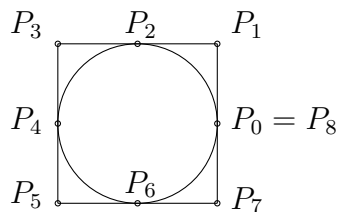
Váhový vektor:  $(1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1)$

**Kuželosečkové oblouky**

Řídící body:

$$[0, 0], [3, 4], [6, 0]$$

Stupeň křivky: 2 (řád 3)

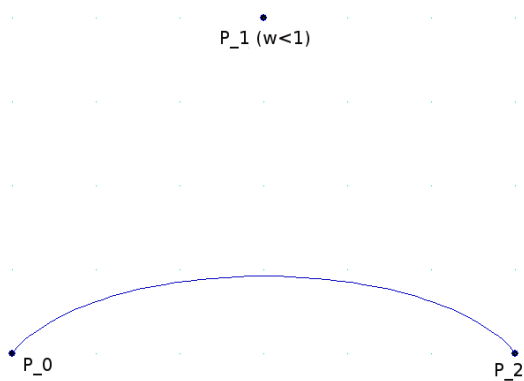


Obrázek 2.31: Kružnice jako NURBS křivka

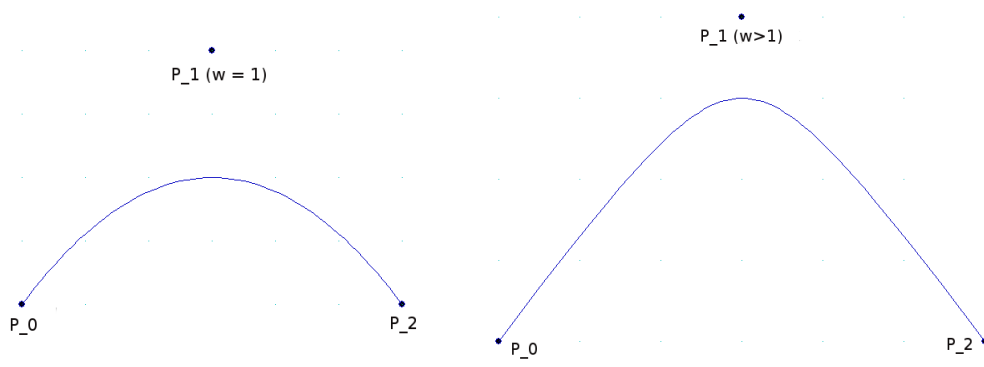
Uzlový vektor:  $(0, 0, 0, 1, 1, 1)$  Váhový vektor  $(w_0, w_1, w_2)$ :

- eliptický oblouk -  $w_0 = 1, w_1 < 1, w_2 = 1$
- parabolický oblouk -  $w_0 = 1, w_1 = 1, w_2 = 1$
- hyperbolický oblouk -  $w_0 = 1, w_1 > 1, w_2 = 1$

Na obr. 2.32, 2.33 vidíme eliptický, parabolický a hyperbolický oblouk.



Obrázek 2.32: Eliptický oblouk



Obrázek 2.33: Parabolický a hyperbolický oblouk

## 2.3 NURBS plochy

NURBS plochy vznikají tenzorovým součinem dvou NURBS křivek, proto se většina vlastností přenáší i na ně. Tenzorový přístup je popsán v kapitole 1. Teoretická východiska a v kapitole 3 Výsledky je potom vysvětlena aplikace tenzorového součinu. V následující části jsou uvedeny základní definice a vlastnosti NURBS ploch.

### 2.3.1 Definice NURBS plochy

*Definice 2.9.* Mějme dáno:

síť  $(q+1)(r+1)$  kontrolních bodů  $P_{i,j}$ , kde  $0 \leq i \leq q, 0 \leq j \leq r$ ,

$(q+1)(r+1)$  kladných reálných čísel  $w_{i,j}$  nazývaných váhy,

stupeň plochy pro řádky  $m$  a stupeň plochy pro sloupce  $n$ ,

řádkový uzlový vektor  $\mathbf{u} = (u_0, u_1, \dots, u_{m+q+1})$ ,

sloupcový uzlový vektor  $\mathbf{v} = (v_0, v_1, \dots, v_{n+r+1})$ .

Pak plochu určenou rovnicí

$$C(u, v) = \frac{\sum_{i=0}^q \sum_{j=0}^r w_{ij} P_{ij} N_i^m(u) N_j^n(v)}{\sum_{i=0}^q \sum_{j=0}^r N_i^m(u) N_j^n(v)}, \quad (2.52)$$

kde  $(u, v) \in \langle u_0, u_{m+q+1} \rangle \times \langle v_0, v_{n+r+1} \rangle$ , nazýváme **NURBS plochou**.

*Poznámka 14.* Pokud položíme

$$R_i^j(u, v) = \frac{N_i^m(u) N_j^n(v) w_{ij}}{\sum_{k=0}^q \sum_{l=0}^r N_k^m(u) N_l^n(v) w_{kl}}, \quad (2.53)$$

dostáváme zkrácený tvar:

$$C(u, v) = \sum_{i=0}^q \sum_{j=0}^r R_i^j(u, v) P_{ij}. \quad (2.54)$$

### 2.3.2 Vlastnosti NURBS ploch

Vlastnosti NURBS ploch se přenášejí z NURBS křivek. Jsou jen jejich rozšířením do prostoru. Jsou uvedeny například v Shene (Naposledy navštíveno 11. 12. 2006), Paseka (Naposledy navštíveno 20. 1. 2007), Piegl – Tiller (2002).

#### 1. Nezápornost

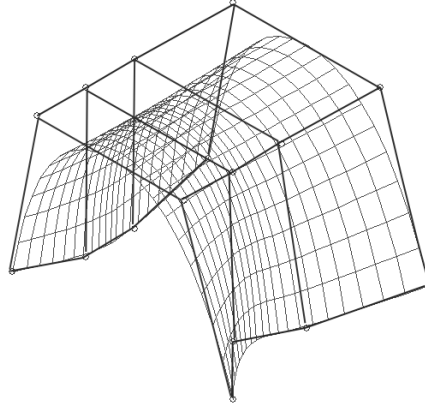
$N_i^m(u) \in \langle 0, 1 \rangle$ ,  $N_j^n(v) \in \langle 0, 1 \rangle$  pro  $(u, v) \in \langle 0, 1 \rangle \times \langle 0, 1 \rangle$ .

#### 2. Rozklad jednotky

$\sum_{i=0}^q \sum_{j=0}^r N_i^m(u) N_j^n(v) = 1$  pro libovolné  $(u, v) \in \langle 0, 1 \rangle \times \langle 0, 1 \rangle$ .

#### 3. Konvexní obal bodů

Body NURBS plochy leží v konvexním obalu sítě řídicích bodů (obr. 2.34).



Obrázek 2.34: Konvexní obal

## 4. Lokální kontrolovatelnost

Na intervalu  $\langle u_k, u_{k+1} \rangle \times \langle v_l, v_{l+1} \rangle$  je nejvýše  $(n+1) \times (m+1)$  bázových funkcí  $R_i^j(u, v)$  nenulových. Jsou to funkce odpovídající indexům  $k-m \leq i \leq k$ ,  $l-n \leq j \leq l$ .

$$R_i^j(u, v) = \begin{cases} > 0 & (u, v) \in \langle u_0, u_{i+m+1} \rangle \times \langle v_0, v_{j+n+1} \rangle \\ 0 & \text{jinak} \end{cases}$$

Z toho vyplývá, že změna polohy libovolného bodu  $P_{ij}$  nebo váhy  $w_{ij}$  změní tvar plochy pouze pro interval  $\langle u_i, u_{i+m+1} \rangle \times \langle v_i, v_{i+n+1} \rangle$ .

## 5. Spojitost

Křivka  $S(u, v)$  má  $C^{m-k}$  resp.  $C^{n-l}$  spojitě parciální derivace v bodě odpovídajícím parametru  $u$ , resp.  $v$ , je-li násobnost uzlu  $u$ , resp.  $v$  rovna  $k$ , resp.  $l$ .

## 6. Projektivní invariantnost

NURBS plocha je invariantní vůči projektivním transformacím. To znamená, že není nutné zobrazovat všechny body plochy, ale při projektivních transformacích stačí zobrazit pouze řídicí body.

## 7. Bézierova plocha – speciální případ NURBS plochy

NURBS křivka je Bézierovou plochou v případě, že počet bodů v řádku je roven řádkovému stupni  $n$ , počet bodů ve sloupci je roven sloupcovému stupni  $m$  a uzlové vektory jsou tvaru:

$$\mathbf{u} = (\underbrace{0, \dots, 0}_{m+1}, \underbrace{1, 1, \dots, 1}_{m+1}), \quad \mathbf{v} = (\underbrace{0, \dots, 0}_{n+1}, \underbrace{1, 1, \dots, 1}_{n+1}).$$

## 2.4 T-spline

Jak je zmíněno v úvodu, poslední novinkou matematických ploch se staly T-spline jako zobecnění NURBS. V komerčním použití se T-spline objevily v softwaru Autodesk Maya (2005) a posléze Rhinoceros (2006). Vážnou slabostí NURBS modelů je nutnost používání obdélníkové sítě řídicích bodů. V praxi to znamená velké množství zbytečných bodů, které jsou nutné pouze kvůli tvaru řídicí sítě. Řešení tohoto problému a mnoha dalších poskytují T-splines. T-splines jsou zobecněním NURBS povrchů, které redukují počet řídicích bodů použitím T-Junctions (viz část 2.4.3). Sederberg et al. (2003b)) popisuje základy T-spline ploch. Zaměřuje se především na vkládání nových řídicích bodů bez nutnosti přidání řádku a sloupce řídicích bodů a také na hladké napojování NURBS ploch s různými uzlovými vektory. Část je věnována T-NURCCs, což jsou Catmull-Clarkovy plochy s T-Junctions – obecně nadmnožina T-spline a NURCC (neuniformní racionální Catmull-Clarkova plocha). Sederberg také zavádí tzv. PB-splines, což je povrch, jehož řídicí body nemají mezi sebou žádnou vzájemnou vazbu (viz část 2.4.2).

Další využití T-spline je popsáno v Sederberg (Naposledy navštíveno 12. 4. 2007). Jedná se o algoritmus T-spline simplification. V praxi to znamená převod obecné NURBS plochy dané obdélníkovou sítí řídicích bodů do sítě nazvané T-mesh, která povoluje T-Junctions (viz část 2.4.3). Výhodou je omezení počtu řídicích bodů o více než polovinu a tedy výrazná úspora paměti. Významným nástrojem v práci s grafickými objekty je lokální zjemnění sítě – tzv. *local refinement*. Znamená to vložení jednoho či více řídicích bodů bez změny tvaru povrchu. Oproti Sederberg et al. (2003b) článek odpovídá na otázku, zda je možné vložit kamkoliv nový řídicí bod a zda je počet pomocných bodů vždy konečný (více v části 2.4.3). Autoři vychází z metody *spline wavelet decomposition* (uvedené v Lyche et al. (2001), Kazinnik – Elber (1997)) a prahování (viz Schroder – Sweldens (1995)).

Pro práci s T-splines je nutné zavést novou strukturu pro B-spline. Jedná se o větší využití uzlového vektoru, který viditelně působí na křivku. Základy této teorie jsou obsaženy v knize (Ramshaw (1989)), jejíž obsah je shrnut v Sederberg et al. (2004). Struktura byla nazvána *polar form* - polární forma nebo také *blossoms* (viz část 2.4.1). Zápis uzlového vektoru je používán pro popsání sítě T-mesh pro T-splines.

Před objevením T-spline v roce 2003 byly alternativou hierarchické B-spline (viz Wang et al. (2005), Bartels – Forsey (1995)). U hierarchických spline existují problémy s tím, že přesunutí stejného řídicího bodu změní plochu pokaždé jinak, v závislosti na aktuální úrovni hierarchie. Hierarchické spline se také mohou pomaleji zobrazovat a renderovat. Dále ve většině implementacích hierarchických B-spline lze rozdělit plochy pouze pravidelně na čtyři menší plošky. U T-splines je možné rozdělit jednu hranu libovolně.

Další využití T-spline je uvedeno v Li et al. (2006). Jedná se o nalezení sítě ve fotografii a její následné převedení do T-spline reprezentace.

Prof. Sederberg, který je jedním z průkopníků v daném oboru, uvádí, kam spěje trend. Uživatel grafického softwaru nebude muset zadávat plochy řídicími body a později modifikovat, ale např. pomocí tabletu načrtne kontury a základní rysy a software sám vytvoří vhodnou síť využívající T-splines.

### 2.4.1 Polární forma – blossoms

Název polární forma – *polar form* zavedl Ramshaw z DEC Systems Research Center. Jedná se nový pohled na B-spline pomocí polární formy. Všechny důležité algoritmy pro Bézierovy a B-spline křivky mohou být odvozeny z následujících čtyř pravidel pro řídicí body – Ramshawem nazvané *polar values*.

1. Pro Bézierovy křivky stupně  $n$  na intervalu  $\langle a, b \rangle$  jsou řídicí body přeznačeny:

$$P_i = P(u_1, u_2, \dots, u_n),$$

kde  $u_j = a$  pro  $j \leq n - i$  a  $u_j = b$  pro ostatní případy.

2. Pro B-spline křivku stupně  $n$  s uzlovým vektorem  $(t_1, t_2, t_3, \dots)$  přísluší řídicím bodům skupina  $n$  po sobě jdoucích uzlů, které splňují:

$$P(t_i, \dots, t_{i+n-1})$$

pro  $i$ -tý bod –  $i$ -tou *polar value*.

3. Polární forma je symetrická ve svých členech. To znamená, že můžeme měnit pořadí argumentů bez změny *polar value*.
4. Máme-li dáno  $P(u_1, u_2, \dots, u_{n-1}, a)$  a  $P(u_1, u_2, \dots, u_{n-1}, b)$  můžeme spočítat hodnotu  $P(u_1, u_2, \dots, u_{n-1}, c)$  pro libovolné  $c$  jako:

$$P(u_1, u_2, \dots, u_{n-1}, c) = \frac{(b - c)P(u_1, u_2, \dots, u_{n-1}, a) + (c - a)P(u_1, u_2, \dots, u_{n-1}, b)}{b - a}.$$

Řekneme, že bod  $P(u_1, u_2, \dots, u_{n-1}, c)$  je afinní kombinací bodů  $P(u_1, u_2, \dots, u_{n-1}, a)$  a  $P(u_1, u_2, \dots, u_{n-1}, b)$ .

#### Příklady

**ad 1.** Je-li Bézierova křivka stupně 3, potom jsou řídicí body určeny jako:

$$\begin{aligned} P_0 &= P(a, a, a) & P_1 &= P(a, a, b) \\ P_2 &= P(a, b, b) & P_3 &= P(b, b, b) \end{aligned}$$

**ad 2.** *Polar values*: B-spline křivka stupně 3 s uzlovým vektorem  $(0, 0, 1, 2, 4, 4, 4)$   
Řídicí body:

$$P_0 = P(0, 0, 1), P_1 = P(0, 1, 2), P_2 = P(1, 2, 4), P_3 = P(2, 4, 4), P_4 = P(4, 4, 4)$$

**ad 3.** Symetrie:

$$P(1, 0, 0, 2) = P(1, 0, 2, 0) = P(2, 0, 0, 1) = P(0, 0, 1, 2) \quad \text{atd.}$$

**ad 4.** Například výpočet:

$$P(0, t, 1) = (1 - t)P(0, 0, 1) + tP(0, 1, 1)$$



### Symetrické polynomy

Zápis polární formy je založen na symetrických polynomech. Hlavní myšlenkou je reprezentovat polynom  $p(t)$  stupně  $m$  jedné proměnné jako polynom více proměnných  $p[t_1, \dots, t_n]$ , kde každá proměnná je stupně jedna,  $n \geq m$  a platí:

$$p[t, \dots, t] = p(t).$$

Polynomy jsou symetrické, neboť podle podmínky 3 předchozí části můžeme argumenty libovolně měnit. Například  $p[a, b, c] = p[b, c, a] = p[c, a, b]$ , atd. Symetrický polynom má tvar:

$$p[t_1, \dots, t_n] = \sum_{i=0}^n c_i p_i[t_1, \dots, t_n],$$

kde

$$p_0[t_1, \dots, t_n] = 1$$

$$p_i[t_1, \dots, t_n] = \frac{\sum_{j=1}^n t_j p_{i-1}[t_1, \dots, t_{j-1}, t_{j+1}, \dots, t_n]}{n}, \quad i = 1, \dots, n.$$

### Uzlový vektor, uzlové intervaly, *polar labels*

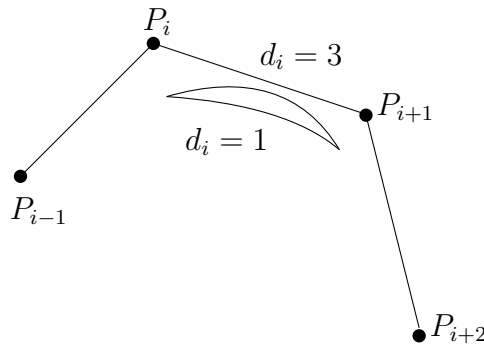
Uzlový vektor je neklesající posloupnost reálných čísel, s jehož pomocí se zadávají obecné NURBS objekty. V Ramshaw (1989) je ukázán nový pohled na uzlový vektor pomocí uzlových intervalů. Uzlové intervaly tvoří posloupnost, jejíž hodnoty jsou rozdíly po sobě jdoucích uzlů v uzlovém vektoru.

#### Příklad:

Uzlový vektor:  $\mathbf{u} = (0, 1, 3, 6, 7)$ .

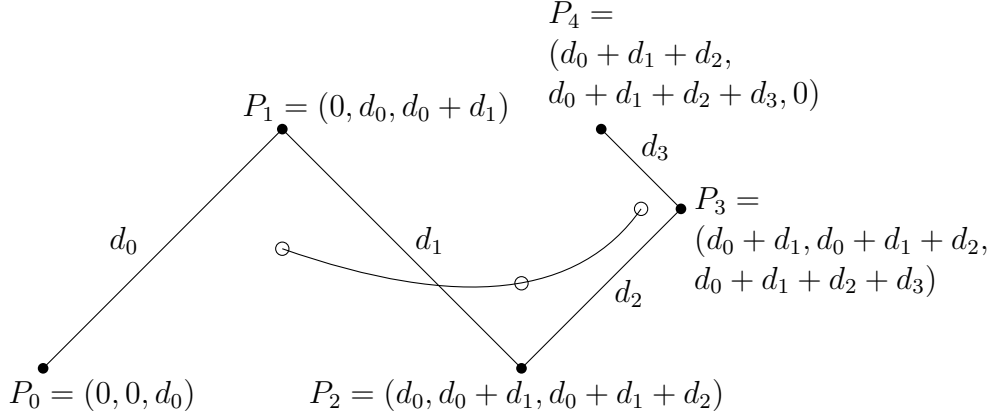
Uzlové intervaly:  $(1, 2, 3, 1)$ .

Uzlové intervaly jsou pouze alternativou zápisu pro uzlový vektor, ale mají lepší vlastnosti. Například jsou více svázány s kontrolním polygonem. Čím větší je uzlový interval pro danou hranu, tím větší délku má výsledná křivka (viz obr. 2.35).



Obrázek 2.35: Vliv velikosti uzlového intervalu na délku křivky

*Polar labels* jsou hodnoty, které jsou přiřazeny každému řídicímu bodu a které využívají uzlové intervaly. Je-li křivka stupně  $n$ , pak má polar label všech řídicích bodů právě  $2n - 1$  členů. Prostřední z nich je hodnota uzlového intervalu hrany předcházející. Odvození je názorně vidět na obr. 2.36.



Obrázek 2.36: *Polar labels*

Pomocí uzlových intervalů lze odvodit i metodu vložení uzlu – *knot insertion* nebo odebrání uzlu – *knot removal*. Obě jsou popsány v Sederberg (Naposledy navštíveno 12. 4. 2007), Ramshaw (1989).

### 2.4.2 PB-splines

PB-splines jsou speciální plochy, jejichž řídicí body nemají vzájemně žádný topologický vztah. Název je odvozen z anglického *point-based*. Základní rovnice je dána vztahem:

$$P(s, t) = \frac{\sum_{i=1}^n P_i B_i(s, t)}{B_i(s, t)}, \quad (s, t) \in D,$$

kde  $P_i$  jsou řídicí body a  $B_i(s, t)$  jsou báze funkce dané předpisem

$$B_i(s, t) = N_{i0}^3(s) N_{i0}^3(t).$$

Funkce  $N_{i0}^3(s)$  je funkce třetího stupně pro uzlový vektor

$$s_i = [s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}]$$

a funkci  $N_{i0}^3(t)$  je přiřazen uzlový vektor

$$t_i = [t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}].$$

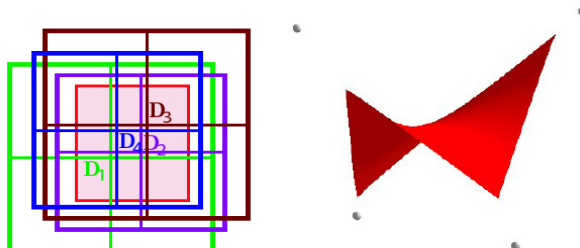
$D$  v rovnici značí definiční obor, kde je daná PB-spline definována. Jediné omezení na  $D$  je, že  $\sum_{i=1}^n B_i(s, t) > 0$ . Každý bod  $P_i$  má definiční obor  $D_i$ . Tedy musí platit:

$$D \subset (D_1 \cup D_2 \cup \dots \cup D_n).$$

PB-splines splňují vlastnosti konvexního obalu. Pokud tedy parametry  $(s, t) \in D$  leží pouze v jednom definičním oboru  $D_i$ , potom  $P(s, t) = P_i$ . Pokud parametry  $(s, t) \in D$

leží ve dvou definičních oborech  $D_i$  a  $D_j$ , potom  $P(s, t)$  leží na spojnici  $P_i$  a  $P_j$ , atd. Proto je vhodné, aby každý bod měl alespoň tři definiční obory vlivu.

U PB-splines není žádná kontrolní síť, uzlový vektor pro jednotlivé báze funkce je úplně nezávislý na uzlových vektorech pro další báze funkce. PB-splines mají pouze teoretický význam jako základ pro T-splines.



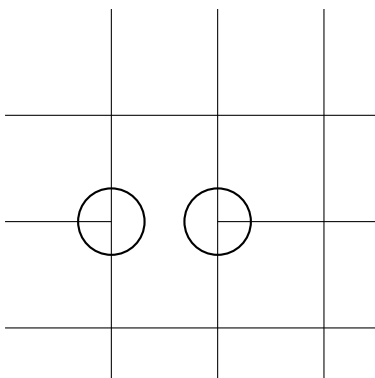
Obrázek 2.37: Definiční obor a výsledná PB-spline, zdroj Sederberg et al. (2003b)

### 2.4.3 T-splines

T-spline je PB-spline, ke které přidáme uspořádání řídicích bodů nazvané T-mesh. Pokud je toto uspořádání pravidelné, dostáváme klasickou NURBS plochu.

#### T-mesh, T-junctions

T-mesh je obdélníková síť, která dovoluje T-junctions. K jejímu popisu se používají konstanty  $t, s$  (nazýváme je  $t$ -hrana,  $s$ -hrana). T-junction je vrchol, kde se sbíhají dvě  $s$ -hrany a jedna  $t$ -hrana nebo dvě  $t$ -hrany a jedna  $s$ -hrana. Každá hrana je označena pomocí uzlových intervalů – dle části 2.4.1.

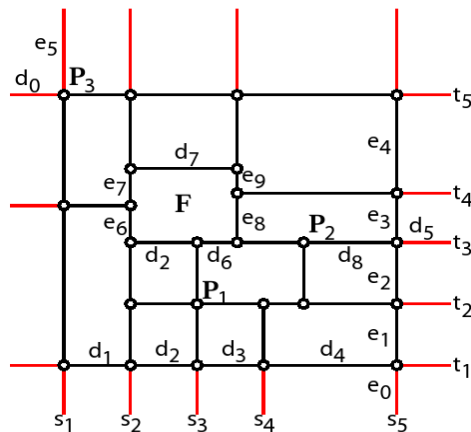


Obrázek 2.38: T-junction

Základní pravidla pro T-mesh jsou:

**Pravidlo 1:** Součet uzlových intervalů na protějších hranách každé stěny se musí rovnat. Podle obr. 2.39 musí například platit:  $d_7 = d_2 + d_6$  a  $e_6 + e_7 = e_8 + e_9$

**Pravidlo 2:** Pokud T-junction na jedné hraně stěny má být spojena pomocí T-junction na protější hraně bez porušení pravidla 1, je nutné zahrnout tuto hranu do T-mesh (rozdělení stěny na dvě části).



Obrázek 2.39: Předloha T-mesh, zdroj Sederberg et al. (2003b)

### Odvození uzlových vektorů

V části 2.4.1 je popsáno využití uzlových intervalů pro křivky. Rozšířením do prostoru dostáváme označení pro hrany T-mesh. Máme tedy danu předlohu pro T-mesh v prostoru parametrů  $(s, t)$ . Označme  $d_i$  a  $e_i$  uzlové intervaly v obou směrech (viz obr. 2.39).

Pro každý bod  $P_i$  sítě existuje bazová funkce definována pomocí uzlových vektorů  $s_i$ ,  $t_i$ . Tyto vektory lze odvodit z T-grid (mřížka znázorňující T-mesh). Délka uzlových vektorů je dána stupněm plochy  $-2$  · stupeň plochy  $-1$ .

Každý uzel v T-mesh odpovídá jednomu řídicímu bodu T-spline plochy. Uzlové souřadnice bodu jsou prostřední členy uzlových vektorů. Odvození zbývajících hodnot se provádí následujícím způsobem. Vedou se čtyři paprsky (nahoru, dolů, doprava, doleva) z pozice odpovídající danému řídicímu bodu a první  $s$ - nebo  $t$ -hrany, které protnou jsou dalšími uzly. Tedy z obr. 2.39 lze odvodit uzlové vektory T-spline plochy stupně tři. Jejich délka je  $2 \cdot 3 - 1 = 5$ .

$P_1$  : pozice  $(s_3, t_2)$

$$s_i = [s_1, s_2, s_3, s_4, s_5 - d_8]$$

$$t_i = [t_1 - e_0, t_1, t_2, t_3, t_4 + e_9]$$

$P_2$  : pozice  $(s_5 - d_8, t_3)$

$$s_i = [s_3, s_3 + d_6, s_5 - d_8, s_5, s_5 + d_5]$$

$$t_i = [t_1, t_2, t_3, t_4, t_5]$$

$P_3$  : pozice  $(s_1, t_5)$  – hraniční bod, nezávisí na  $s_{3,0}$  a  $t_{3,4}$

$$s_i = [s_1 - d_0, s_1 - d_0, s_1, s_2, s_2 + d_7]$$

$$t_i = [t_1, t_4 - e_9 + e_7, t_5, t_5 + e_5, t_5 + e_5]$$

### Základní vztahy

Obecná T-spline plocha je dána vztahem:

$$P(s, t) = \frac{\sum_{i=0}^n P_i w_i B_i(s, t)}{\sum_{i=0}^n w_i B_i(s, t)}. \quad (2.55)$$

kde  $P_i$  jsou řídicí body s vahami  $w_i$  a báze funkce jsou dány jako:

$$B_i(s, t) = N[s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}](s) N[t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}](t).$$

Bázová funkce  $N[s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}](s)$  je asociována s uzlovým vektorem

$$s_i = [s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}]$$

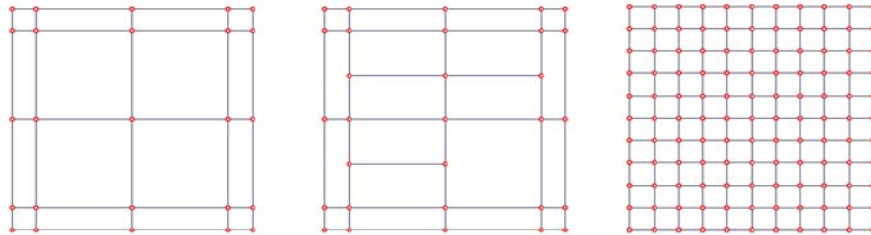
a funkci  $N[t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}](t)$  je přiřazen uzlový vektoru

$$t_i = [t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}].$$

### T-spline Local Refinement

T-spline Local Refinement je algoritmus, který slouží k vložení nových řídicích bodů do T-mesh beze změny tvaru výsledného povrchu. Samotný proces má dvě fáze – topologickou a geometrickou. V topologické části se určí, zda je nutné přidávat další řídicí body do T-mesh a určí se. V geometrické části je potom proveden samotný výpočet.

Definujme T-spline prostor, což je množina T-spline ploch, které mají stejnou T-mesh topologii a uzlový souřadný systém. T-spline prostor může být reprezentován pomocí diagramu znázorňující T-mesh. Na T-spline prostorech lze také definovat relaci uspořádání. Řekneme, že T-spline prostor  $S_1$  je podprostorem  $S_2$ , pokud existuje lokální zjemnění, které převede T-spline v  $S_1$  do T-spline v  $S_2$  (viz obr. 2.40).



Obrázek 2.40: T-spline podprostory

Obecně mějme dvě T-spline plochy  $P(s, t) \in S_1$  a  $\bar{P}(s, t) \in S_2$ . Označíme  $P$  sloupcový vektor řídicích bodů pro  $P(s, t)$  danou rovnicí (2.55) a  $\bar{P}$  sloupcový vektor řídicích bodů pro  $\bar{P}(s, t)$ . Vezmeme lineární transformaci, která zobrazí  $P$  na  $\bar{P}$ .

$$M_{12}P = \bar{P}.$$

Ze vztahu podprostoru můžeme vyjádřit bázeovou funkci  $B_i(s, t)$  jako lineární kombinaci  $\bar{B}_i(s, t)$ :

$$B_i(s, t) = \sum_{j=1}^{\bar{n}} c_i^j \bar{B}_j(s, t). \quad (2.56)$$

Chceme, aby

$$P(s, t) \equiv \bar{P}(s, t). \quad (2.57)$$

To je splněno, pokud

$$\bar{P}_i = \sum_{i=1}^n c_i^j P_i. \quad (2.58)$$

### Algoritmus T-spline local refinement

Připomeňme, že pro každý bod T-spline plochy je pevně svázán se dvěma uzlovými vektory, které jsou odvozené z T-mesh a které určují příslušnou báзовou funkci. Během algoritmu vkládání bodu se mohou vyskytnout tři druhy porušení tohoto pravidla.

**Chyba 1** Báзовá funkce nemá uzel, který by měla mít odvozením z T-mesh výše popsaným způsobem.

**Chyba 2** Báзовá funkce má uzel, který by neměla mít odvozením z T-mesh výše popsaným způsobem.

**Chyba 3** Existuje řídicí bod, který není spjat s žádnou báзовou funkcí.

Pokud se po vložení bodu neobjeví žádná chyba, je T-spline v pořádku a může se vykreslit. Pokud se vyskytnou nějaké chyby, musí se podniknout vhodné kroky na jejich odstranění. Algoritmus úpravy lze popsat následujícími čtyřmi kroky:

1. Vložíme všechny požadované body do T-mesh.
2. Pokud nějaká báзовá funkce nemá uzel, který by měla mít odvozením – **Chyba 1**, provede se vložení uzlu do této funkce podle vzorců uvedených níže.
3. Pokud nějaká báзовá funkce má uzel, který by neměla mít odvozením – **Chyba 2**, provede se vložení vhodného řídicího bodu do T-mesh.
4. Opakujeme kroky 2 a 3 dokud nejsou všechny chyby ošetřeny.

### Vkládání nového uzlu do uzlového vektoru báзовé funkce

Máme dán uzlový vektor  $\mathbf{s} = (s_0, s_1, s_2, s_3, s_4)$  a jemu odpovídající báзовou funkci  $N(s) = (N(s_0, s_1, s_2, s_3, s_4))$ . Vložíme uzel  $s_0 < k < s_4$  a podle pořadí se provede odpovídající přepočítání.

Pro  $\bar{\mathbf{s}} = (s_0, k, s_1, s_2, s_3, s_4)$

$$N(s) = c_0 N(s_0, k, s_1, s_2, s_3)(s) + d_0(k, s_1, s_2, s_3, s_4)(s)$$

kde  $c_0 = \frac{k-s_0}{s_3-s_0}$  a  $d_0 = 1$ .

Pro  $\bar{s} = (s_0, s_1, k, s_2, s_3, s_4)$

$$N(s) = c_1 N(s_0, s_1, k, s_2, s_3)(s) + d_1(s_1, k, s_2, s_3, s_4)(s)$$

kde  $c_1 = \frac{k-s_0}{s_3-s_0}$  a  $d_0 = \frac{s_4-k}{s_4-s_1}$ .

Pro  $\bar{s} = (s_0, s_1, s_2, k, s_3, s_4)$

$$N(s) = c_2 N(s_0, s_1, s_2, k, s_3)(s) + d_2(s_1, s_2, k, s_3, s_4)(s)$$

kde  $c_2 = \frac{k-s_0}{s_3-s_0}$  a  $d_2 = \frac{s_4-k}{s_4-s_1}$ .

Pro  $\bar{s} = (s_0, s_1, s_2, s_3, k, s_4)$

$$N(s) = c_3 N(s_0, s_1, s_2, s_3, k)(s) + d_3(s_1, s_2, s_3, k, s_4)(s)$$

kde  $c_3 = 1$  a  $d_3 = \frac{s_4-k}{s_4-s_1}$ .

Vložení uzlu můžeme provést pro parametr  $s$  i pro parametr  $t$  pouze opakováním předchozí konstrukce.

### T-spline Simplification

Při návrhu obecných tvarů v NURBS modelovacích prostředích dochází ke vzniku tzv. *superflous* – nadbytečných bodů. T-spline simplification slouží k odstranění většiny z nich z původní NURBS, resp. T-spline plochy.

Jako vstup máme T-spline nebo NURBS plochu, kterou chceme zjednodušit. Nejprve pomocí postupu uvedeného v (Lyche (1993)) nalezneme aproximace nejmenšími čtverci výchozí plochy  $T_n$ , která se skládá ze čtverců  $4 \times 4$ . Tuto plochu označíme  $T_0$  a patří do prostoru  $S_0$ . Označíme  $\mathbf{P}^0$  vektor bodů plochy  $T_0$  a  $\mathbf{P}^n$  vektor bodů původní plochy  $T_n$ . Vypočítáme matici  $M_{0,n}$  přechodu mezi jednotlivými prostory a rozdíl:

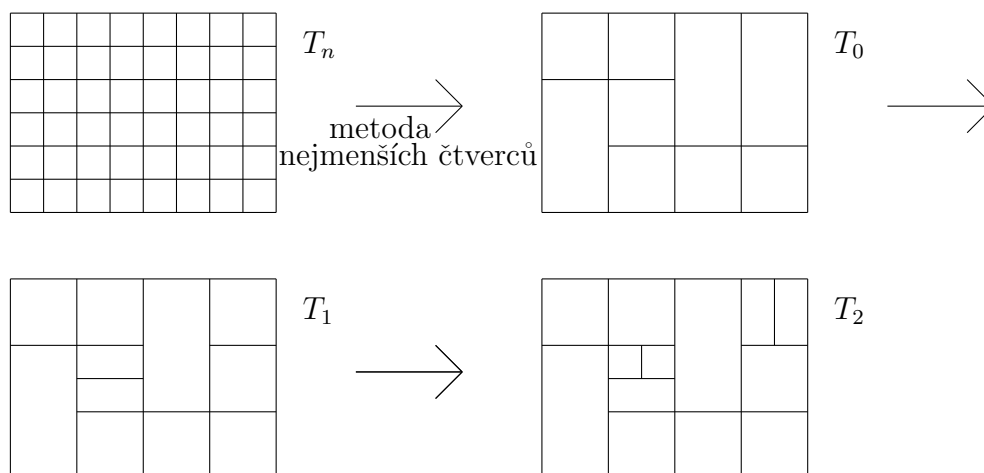
$$D_{0n} = M_{0n} \mathbf{P}^0 - \mathbf{P}^n$$

Pokud velikost (odmocnina z druhých mocnin jednotlivých složek) chybového vektoru  $D_{0n}$  přesahuje určitou prahovou hodnotu, dojde k dělení čtverců, ve kterých chybové body leží. Dělení se provádí v polovině strany, která obsahuje více uzlových čar. Tím přejdeme k podprostoru  $S_1$  a můžeme znovu zkontrolovat chybovou matici  $D_{1n}$ . Obecně pro  $i$ -tý krok bereme:

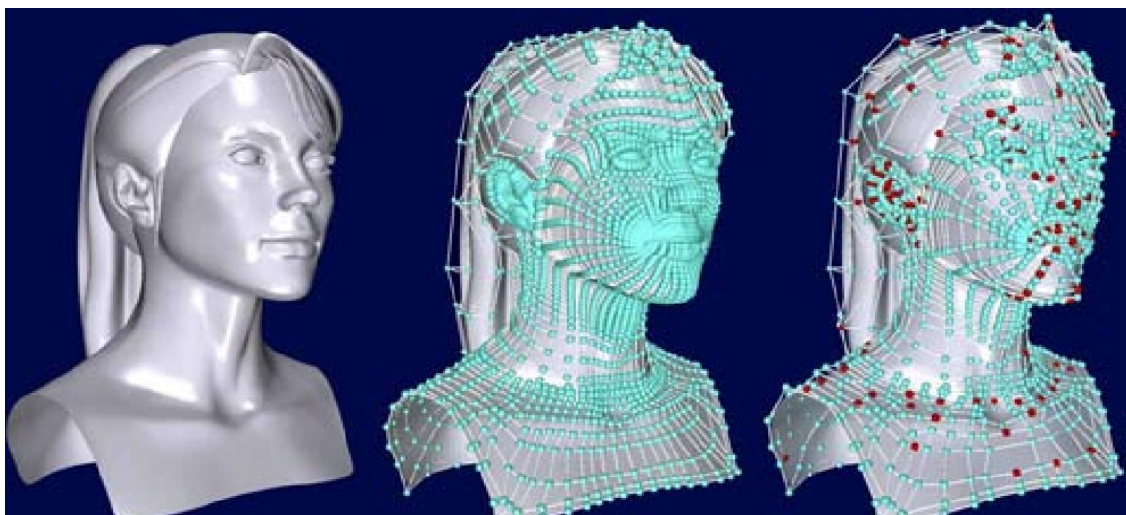
$$D_{in} = M_{in} \mathbf{P}^i - \mathbf{P}^n$$

Pokud je chyba pro všechny body z vektoru  $D_{in}$  pod prahovou úrovní, je plocha  $T_i$  prostoru  $S_i$  hledanou zjednodušenou plochou bez většiny nadbytečných bodů. Na schématu 2.41 je znázorněn postup T-spline simplification a na obr. 2.43 a 2.44 je ukázka ze softwaru Maya. K vykreslení stejného modelu je potřeba méně než polovina původních bodů při zachování modelu bez viditelné změny.

Při vkládání nových hran se používá algoritmus local refinement uvedený výše, kde se krajní body vložené hrany vkládají jako nové body sítě. Někdy je nutné vložit i další pomocné body.

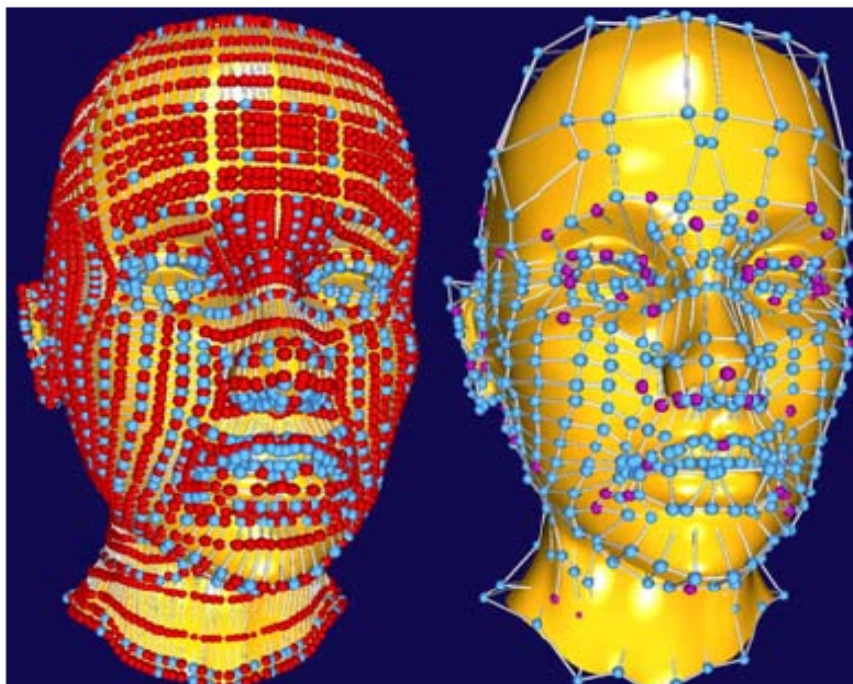


Obrázek 2.41: Postup při T-spline simplification



Obrázek 2.42: Srovnání NURBS plochy (10 305 bodů) a T-spline (3 955 bodů)





Obrázek 2.43: Srovnání NURBS plochy (4712 bodů) a T-spline (1109 bodů)



Obrázek 2.44: NURBS plochy před a po algoritmu T-spline simplification



# Kapitola 3

## Výsledky práce

Při vědecké práci se lze zabývat známými pojmy, vlastnostmi a algoritmy s cílem odhalit nové souvislosti, objevit elegantnější důkaz či efektivnější algoritmus. Druhou možností je vydat se cestou nového vývoje do neprobádaných oblastí. Ve své práci jsem využila oba přístupy. První přístup při zkoumání derivací, vlastností a způsobech implementace NURBS objektů, druhý přístup při rozboru a využití T-spline ploch.

### Hlavní cíle disertační práce:

1. Analytické vyjádření derivace NURBS křivek, nový přímý důkaz vzorce pro výpočet, srovnání se současně používanou numerickou derivací.
2. Využití softwaru Maple pro modelaci vlivu uzlového vektoru na tvar NURBS.
3. Využití NURBS interpolací v lékařství.
4. Matematická formulace nových trendů (T-Junctions, T-splines, T-NURCCs) a jejich nové uplatnění v geografických informačních systémech.
5. Popsat jednoduchá NURBS tělesa (válec, kužel, koule, anuloid) z netradičního tenzorového pohledu.
6. Navrhnout efektivní způsoby implementace NURBS křivek, ploch a základních těles s ohledem na jejich stabilitu a rychlost.
7. Ověřit navržené metody implementací do německého komerčního CAD softwaru *RFEM3D*.
8. Srovnat implementaci a stabilitu NURBS v grafických programech, kde jsou implementovány s programem RFEM 3D.

### 3.1 Derivace NURBS křivek

Pro technickou praxi jsou první derivace nepostradatelné. Využívají se při mnoha fyzikálních výpočtech. NURBS křivka je definována rekurentně, přímý výpočet je tedy velmi obtížný.

Následující pasáž popisuje mnou navržený důkaz vzorce pro výpočet derivací NURBS křivek, který je uveřejněn na stránkách Michigan Technological University (viz Shene (Naposledy navštíveno 11. 12. 2006)). Při dalším studiu jsem objevila obdobný důkaz v Piegl – Tiller (2002). Tento důkaz však na rozdíl od zde prezentovaného předpokládá existenci speciálních vlastností NURBS křivek. Zde provedený důkaz je tedy obecnější. Důkaz i jeho aplikace byly prezentovány na Konferenci o počítačové grafice a geometrii (viz Procházková (2005)) a na evropské konferenci WSCG 2007 (Procházková – Procházka (2007)).

### 3.1.1 Věta o derivaci B-spline funkce

**Věta 3.1.** *Máme-li danu B-spline funkci danou definicí 2.1 na str. 24, pak její první derivaci lze vyjádřit jako*

$$C(t)' = \sum_{i=0}^m N_i^n(t)' P_i, \quad (3.1)$$

kde

$$N_i^n(t)' = \frac{n}{t_{i+n} - t_i} N_i^{n-1}(t) - \frac{n}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \quad (3.2)$$

**Důkaz:**

Důkaz provedeme matematickou indukcí vzhledem ke stupni křivky  $n$ .

1.  $n = 0$

$$N_i^0(t) = \begin{cases} 1 & \text{pro } t \in \langle t_i, t_{i+1} \rangle \\ 0 & \text{jinak} \end{cases}$$

Je zřejmé, že derivace je ve všech případech nula.

Po dosazení  $n = 0$  do rovnice 3.2 je daný výraz také roven nule.

Pro  $n = 0$  tedy věta platí.

2. Předpokládejme, že věta platí pro krok  $k = 0, 1, 2, \dots, n$ .

Hledáme  $N_i^{n+1}(t)'$ . Rozepsáním B-spline funkcí podle definice máme

$$N_i^{n+1}(t)' = \left( \frac{t - t_i}{t_{i+n+1} - t_i} N_i^n(t) + \frac{t_{i+n+2} - t}{t_{i+n+2} - t_{i+1}} N_{i+1}^n(t) \right)', \quad (3.3)$$

což zderivujeme jako součet dvou součinů

$$\begin{aligned} N_i^{n+1}(t)' &= \frac{t - t_i}{t_{i+n+1} - t_i} N_i^n(t)' + \frac{1}{t_{i+n+1} - t_i} N_i^n(t) \\ &+ \frac{t_{i+n+2} - t}{t_{i+n+2} - t_{i+1}} N_{i+1}^n(t)' - \frac{1}{t_{i+n+2} - t_{i+1}} N_{i+1}^n(t). \end{aligned} \quad (3.4)$$

Podle předpokladu známe derivace stupně  $n$ , které dosadíme do předchozí rovnice a roznásobíme

$$N_i^{n+1}(t)' = \frac{t - t_i}{t_{i+n+1} - t_i} \left( \frac{n}{t_{i+n} - t_i} N_i^{n-1}(t) \right) \quad (3.5)$$

$$- \frac{t - t_i}{t_{i+n+1} - t_i} \left( \frac{n}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \right) \quad (3.6)$$

$$+ \frac{t_{i+n+2} - t}{t_{i+n+2} - t_{i+1}} \left( \frac{n}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \right) \quad (3.7)$$

$$- \frac{t_{i+n+2} - t}{t_{i+n+2} - t_{i+1}} \left( \frac{n}{t_{i+n+2} - t_{i+2}} N_{i+2}^{n-1}(t) \right) \quad (3.8)$$

$$+ \frac{1}{t_{i+n+1} - t_i} N_i^n(t) - \frac{1}{t_{i+n+2} - t_{i+1}} N_{i+1}^n(t). \quad (3.9)$$

Pokud uděláme malou úpravu s hledaným výrazem  $N_i^{n+1}(t)'$  a rozepíšeme polynomy  $N_i^n$ ,  $N_{i+1}^n$  podílů obsahujících v čitateli proměnnou  $n$ , dostáváme:

$$N_i^{n+1}(t)' = \frac{1}{t_{i+n+1} - t_i} N_i^n(t) - \frac{1}{t_{i+n+2} - t_{i+1}} N_{i+1}^n(t) \quad (3.10)$$

$$+ \frac{n}{t_{i+n+1} - t_i} \left( \frac{t - t_i}{t_{i+n} - t_i} N_i^{n-1}(t) \right) \quad (3.11)$$

$$+ \frac{n}{t_{i+n+1} - t_i} \left( \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \right) \quad (3.12)$$

$$- \frac{n}{t_{i+n+2} - t_{i+1}} \left( \frac{t - t_{i+1}}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \right) \quad (3.13)$$

$$- \frac{n}{t_{i+n+2} - t_{i+1}} \left( \frac{t_{i+n+2} - t}{t_{i+n+2} - t_{i+2}} N_{i+2}^{n-1}(t) \right). \quad (3.14)$$

Porovnání obou vyjádření je zřejmé, že části 3.10 a 3.9, 3.11 a 3.5, 3.14 a 3.8 se rovnají. Pro části 3.12, 3.13 a 3.6, 3.7 u polynomu  $N_{i+1}^{n-1}$  však rovnost není zřejmá. Musí se provést úpravy částí 3.6, 3.7. Části 3.6, 3.7 mají společný součin  $n \cdot N_{i+1}^{n-1}$ , budeme počítat bez něj. Výraz

$$- \frac{t - t_i}{(t_{i+n+1} - t_i)(t_{i+n+1} - t_{i+1})} + \frac{t_{i+n+2} - t}{(t_{i+n+2} - t_{i+1})(t_{i+n+1} - t_{i+1})}$$

převědeme na společného jmenovatele a roznásobíme

$$\frac{-tt_{i+n+2} + tt_{i+1} + t_it_{i+n+2} - t_it_{i+1} + t_{i+n+2}t_{i+n+1} - t_it_{i+n+2} - tt_{i+n+1} + tt_i}{(t_{i+n+1} - t_i)(t_{i+n+1} - t_{i+1})(t_{i+n+2} - t_{i+1})}.$$

Součin  $t_it_{i+n+2}$  se odečte a nyní provedeme umělý krok – v čitateli přičteme a odečteme součin  $t_{i+1}t_{i+n+1}$ . Vytýkáním upravíme na tvar:

$$\frac{(t_{i+n+2} - t_{i+1})(t_{i+n+1} - t) + (t_{i+n+1} - t_i)(t_{i+1} - t)}{(t_{i+n+1} - t_i)(t_{i+n+1} - t_{i+1})(t_{i+n+2} - t_{i+1})}.$$

Rozdělíme na dva zlomky, zkrátíme a přidáním společné části  $n \cdot N_{i+1}^{n-1}$  dostáváme

$$\frac{n(t_{i+n+1} - t)}{(t_{i+n+1} - t_i)(t_{i+n+1} - t_{i+1})} N_{i+1}^{n-1} - \frac{n(t - t_{i+1})}{(t_{i+n+1} - t_{i+1})(t_{i+n+2} - t_{i+1})} N_{i+1}^{n-1},$$

což jsou sčítance shodné s částmi 3.12, 3.13. Tím jsme dokázali rovnost vztahů a věta platí.

### 3.1.2 Programování derivací NURBS křivek

V rámci své spolupráce s firmou Fem Consulting, jsem programovala výpočet derivací s použitím dokázaného vzorce do jejich komerčního softwaru RFEM 3D, kde byl v roce 2006 implementován.

NURBS křivky jsou definovány jako lomená funkce s B-spline funkcí v čitateli i jmenovateli. Je nutné tento vztah derivovat jako podíl. Nejdříve je nutné ošetřit diferencovatelnost, zda násobnost uzlového vektoru nezpůsobí bod spojitosti  $C_0$ . V dalším kroku zkontrolovat speciální případ, který může nastat a poté samotný výpočet. Následující věta udává, jaký vliv má opakování uzlu v uzlovém intervalu:

**Věta 3.2.** *Je-li násobnost uzlu rovna  $s$ , pak křivka stupně  $n$  má v bodě, který odpovídá tomuto parametru spojitost  $C_{n-s}$*

#### Algoritmus výpočtu

**Vstup:** NURBS křivka, parametr  $t$

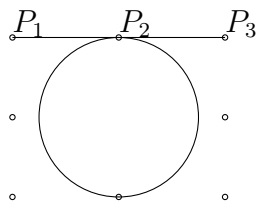
**Výstup:** derivace v bodě křivky  $C(t)$

##### 1. Ověření diferencovatelnosti

- Nalezení intervalu, kde parametr  $t$  leží.
- Je-li parametr  $t$  roven uzlu, určení jeho násobnosti.
- Je-li spojitost podle věty 3.2 alespoň  $C_1$ , lze derivaci spočítat.
- *speciální případ*

Je-li spojitost v daném bodě  $C_0$  a okolní řídicí body od bodu, ve kterém se počítá derivace, leží na přímce, mění se tím spojitost na  $C_1$  a derivaci lze spočítat.

Názorně to vidíme na případě kružnice (obr. 3.1).



Obrázek 3.1: Speciální případ derivace

Uzlový vektor pro kružnici je  $(0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1)$  a stupeň je dva. Pro parametr  $t = 0.25$  je násobnost uzlu 2, spojitost podle věty 3.2 by měla být  $C_0$ . Pro tento parametr je odpovídajícím bod křivky bod  $P_2$ . Okolní řídicí body  $P_1$  a  $P_3$  společně s tímto bodem leží na jedné přímce. V tomto bodě musí derivace existovat, neboť zde existuje tečna  $P_1P_3$ . K ověření této podmínky jsem sestavila konečný automat, který je znázorněn na obr. 3.3. Jednotlivé stavy nabývají hodnot od nuly do 17, koncový stav je 50, pokud podmínka není splněna, nebo 100, pokud podmínka platí.

## 2. Odvození vzorce pro výpočet

Máme danu obecnou NURBS křivku (viz definice 2.20, str. 46). Její derivaci provedeme zderivováním výrazu jako podílu:

$$C(t)' = \left( \frac{\sum_{i=0}^m w_i P_i N_i^n(t)}{\sum_{i=0}^m w_i N_i^n(t)} \right)' = \quad (3.15)$$

$$= \frac{\sum_{i=0}^m w_i P_i N_i^n(t)' \sum_{i=0}^m w_i N_i^n(t) - \sum_{i=0}^m w_i P_i N_i^n(t) \sum_{i=0}^m w_i N_i^n(t)'}{(\sum_{i=0}^m w_i N_i^n(t))^2}.$$

## 3. Programové zpracování

- Výpočet polynomů  $N_i^n(t)$  – pomocí deBoorova algoritmu.

Postup:

1. Inicializace: naplníme pole nulami, kromě proměnné  $N[0][\text{koeft}]$ , kde **koeft** je koeficient levého uzlu intervalu, kde parametr  $t$  leží. Podle definice B-spline funkce je  $N[0][\text{koeft}] = 1$ .

2. Výpočet: pomocí dvou **for**-cyklů, postup je zřejmý z obr. 2.11 a podrobně popsán v části týkající se B-spline funkcí.

$N[0]=1.0;$

```
for (j=1; j<=stupen-pocetOpakovaniUzlu; j++){
    left[j]=t-uzlVektor[i+1-j]
    right[j]=uzlVektor[i+j]-t;
    saved = 0.0;
    for (r = 0; r<j; r++){
        temp = N[r]/(right[r+1]+left[j-r]);
        N[r] = saved+right[r+1]*temp;
        saved = left[j-r]*temp;
    }
    N[j]=saved;
}
```

- Výpočet  $\sum_{i=0}^m P_i w_i N_i^n(t)'$  a  $\sum_{i=0}^m w_i N_i^n(t)'$

Podle věty 3.1 platí:

$$\sum_{i=0}^m P_i w_i N_i^n(t)' = \sum_{i=0}^m P_i w_i \left( \frac{n}{t_{i+n} - t_i} N_i^{n-1}(t) - \frac{n}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \right).$$

Z předchozího kroku jsou uloženy hodnoty  $N_i^{n-1}(t)$  a ty můžeme použít. Výsledek bude uložen ve dvou polích – `PiWiNinDer`, `WiNinDer`, přičemž první z nich je trojrozměrné a druhé jednorozměrné.

- Výpočet  $\sum_{i=0}^m P_i w_i N_i^n(t)$  a  $\sum_{i=0}^m w_i N_i^n(t)$

Výpočet je velmi jednoduchý, z de Boorova algoritmu známe hodnoty všech polynomů  $N_i^n(t)$  uložených v poli `N`. Výsledné součty uložíme do dvou polí – `soucetPiWiNin` a `soucetWiNin`.

- Výsledek

Nyní stačí vypočtené hodnoty dosadit do vztahu (3.15) a výsledné derivace jsou v trojrozměrném poli `exit`.

```
for (i = 0, 1, 2)
    exit[i] =
        (PiWiNinDer[i]*soucetWiNin - soucetPiWiNin[i]*WiNinDer)/
        soucetWiNin^2;
```

*Poznámka 15.* Hodnota `soucetWiNin` použitá v předchozím výpočtu je rovna jedné, pokud mají všechny váhy hodnotu jedna (podle vlastnosti 4, str. 33 – součet jednotky).

V následující tabulce jsou uvedené vypočtené hodnoty derivace numerickou metodou, která byla již implementována do softwaru RFEM 3D. Druhé hodnoty jsou vypočtené analyticky pomocí výše uvedeného algoritmu a dokázaného vzorce. Analytická metoda dává přesné hodnoty. Její použití bylo vhodnější, neboť používaná numerická metoda není naprogramovaná s tak velkou přesností a při fyzikálních výpočtech je na přesnost kladen velký důraz.

Testovací křivka byla zadána:

Řídicí body:  $[0, 0, 0]$ ,  $[1, 1, 0]$ ,  $[2, 0, 0]$ ,  $[3, 1, 0]$ ,  $[4, 0, 0]$

Váhový vektor:  $(1, 1, 1, 1, 1)$

Stupeň křivky: 3

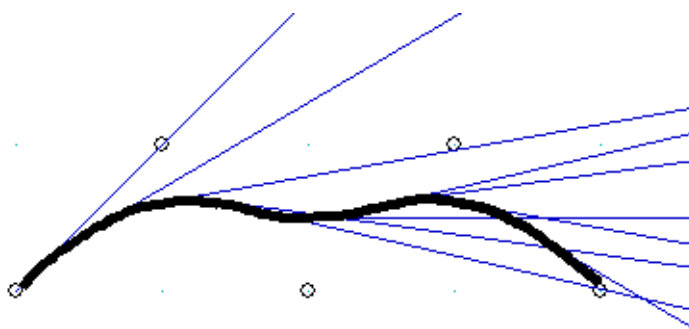
Uzlový vektor:  $(0, 0, 0, 0, 0.5, 1, 1, 1, 1)$

Pro hodnotu  $t = 0$  se počítá derivace v prvním řídicím bodě  $[0, 0, 0]$ . Vzhledem k zadání křivky je tečnou spojnice prvních dvou řídicích bodů, tedy  $(1, -1)$ . (V softwaru je osa  $y$  opačně orientovaná.) V numerickém výpočtu vychází hodnoty  $(5.97, -5.91)$ . Zlepšení oproti dříve implementované metodě je v řádech setin.

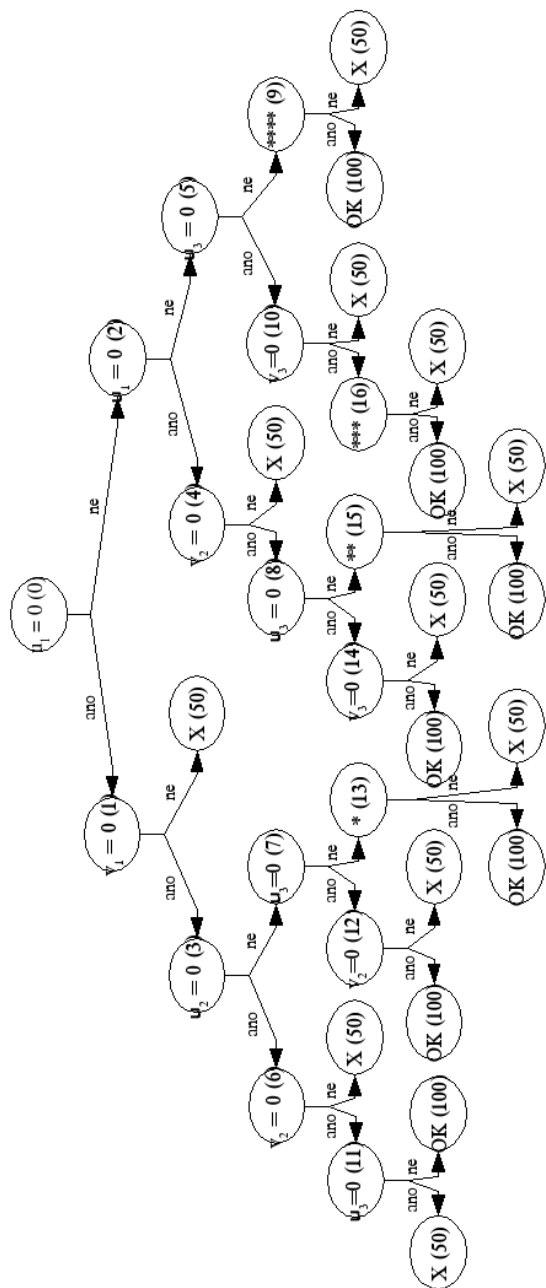


<b>t</b>	Numerická derivace		Analytická derivace		Rozdíl hodnot	
	$dx$	$dy$	$dx$	$dy$	$\Delta dx$	$\Delta dy$
0	5.9701	-5.9104	6	-6	-0.0299	0.0896
0.1	4.9201	-2.8804	4.92	-2.88	0.0001	-0.0004
0.2	4.0801	-0.7204	4.08	-0.72	0.0001	-0.0004
0.3	3.4801	0.4796	3.48	0.48	0.0001	-0.0004
0.4	3.1201	0.7196	3.12	0.72	0.0001	-0.0004
0.5	3.001	0.000	3	0	0.0001	0
0.6	3.1201	-0.7196	3.12	-0.72	0.0001	0.0004
0.7	3.4801	-0.4796	3.48	-0.48	0.0001	0.0004
0.8	4.0801	0.7204	4.08	0.72	0.0001	0.0004
0.9	4.9201	2.8804	4.92	2.88	0.0001	0.0004
0.9999	5.9695	5.908616	5.9988	5.9964	-0.0293	-0.0878

Tabulka 3.1: Derivace vypočítané numericky a analyticky v softwaru RFEM 3D



Obrázek 3.2: Vykreslené tečny pro hodnoty z tabulky



Obrázek 3.3: Konečný automat pro ověření kolineace bodů

## 3.2 Vliv uzlového vektoru na NURBS plochy

Každá NURBS plocha potřebuje ke svému zadání dva uzlové vektory. Jak již bylo řečeno, jsou to neklesající posloupnosti kladných reálných čísel, které mají určitá omezení a jejich modifikací lze měnit tvar plochy. Ovládání tvaru plochy pomocí uzlových vektorů je v literatuře silně opomíjeno a přitom se jedná o obtížný problém.

V následující části jsou uvedeny základní typy uzlových vektorů. Zároveň jsou v softwaru Maple vykresleny jejich báze funkce a výsledné plochy jsou všechny zobrazené v aplikaci FemDev, což je testovací prostředí softwaru RFEM 3D.

Bázovými funkcemi pro NURBS plochy je součin dvou B-spline funkcí (viz rovnice (2.52)). Počet těchto součinů je  $(q + 1) \times (r + 1)$ . Rozepsáním jsou to:

$$\sum_{i=0}^q \sum_{j=0}^r N_i^m(u) N_j^n(v) = N_0^n N_0^m + N_0^n N_1^m + \dots + N_0^n N_r^m + N_1^n N_0^m + \dots + N_q^n N_r^m$$

Každý z těchto součinů je násoben příslušným řídicím bodem  $P_{ij}$ . Hodnota součinu je vždy číslo v intervalu  $\langle 0, 1 \rangle$  podle vlastnosti 1 v části 2.3.2. Toto číslo udává – stejně jako u křivek – kolika procenty má daný bod vliv na polohu výsledného bodu.

### Příklad:

Vezměme dva ekvidistantní uzlové vektory pro plochu stupně dva.

$$\mathbf{u} = (0, 1, 2, 3, 4, 5, 6, 7)$$

$$\mathbf{v} = (0, 1, 2, 3, 4, 5, 6, 7)$$

Pro libovolnou dvojici  $(u, v)$  spočítáme vliv jednotlivých báze funkcí na výslednou hodnotu bodu:

Zvolíme  $(u, v) = (2.8, 4.2)$ . Hodnota 2.8 leží v intervalu  $\langle 2, 3 \rangle$  a hodnota  $v = 4.2$  v intervalu  $\langle 4, 5 \rangle$ . Pro parametr  $u$  jsou nenulové báze funkce  $N_0^2, N_1^2, N_2^2$  a pro parametr  $v$   $N_2^2, N_3^2, N_4^2$ .

Vypočtené hodnoty:

$$N_0^2(2.8) = 0.02$$

$$N_1^2(2.8) = 0.66$$

$$N_2^2(2.8) = 0.32$$

$$N_2^2(4.2) = 0.32$$

$$N_3^2(4.2) = 0.66$$

$$N_4^2(4.2) = 0.02$$

Výsledný bod tedy ovlivňuje devět součinů vytvořených z hodnot uvedených výše:

$$\begin{aligned} S(2.8, 4.2) &= P_{02} N_0^2(2.8) N_2^2(4.2) + P_{03} N_0^2(2.8) N_3^2(4.2) + P_{04} N_0^2(2.8) N_4^2(4.2) \\ &+ P_{12} N_1^2(2.8) N_2^2(4.2) + P_{13} N_1^2(2.8) N_3^2(4.2) + P_{14} N_1^2(2.8) N_4^2(4.2) \\ &+ P_{22} N_2^2(2.8) N_2^2(4.2) + P_{23} N_2^2(2.8) N_3^2(4.2) + P_{24} N_2^2(2.8) N_4^2(4.2). \end{aligned}$$

Dosazením hodnot polynomů dostáváme:

$$\begin{aligned} S(2.8, 4.2) &= P_{02}0.0064 + P_{03}0.0132 + P_{04}0.0004 \\ &+ P_{12}0.2112 + P_{13}0.4356 + P_{11}0.0132 \\ &+ P_{22}0.1024 + P_{23}0.2112 + P_{21}0.0064. \end{aligned}$$

Znamená to, že například bod  $P_{13}$  má na výslednou hodnotu vliv 43.56 procent. Také podle vlastnosti 2 na straně 55 je vidět, že součet všech vypočítaných hodnot je jedna.

### 3.2.1 Ekvidistantní uzlový vektor $(0, 1, 2, 3, 4, 5, 6, 7)$

Bázové funkce stupně jedna

$$N_0^1(u) N_0^1(v) = \begin{cases} uv & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 0, 1 \rangle \\ u(2-v) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 1, 2 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^1(u) N_1^1(v) = \begin{cases} u(v-1) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 1, 2 \rangle \\ u(3-v) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^1(u) N_2^1(v) = \begin{cases} u(v-2) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 2, 3 \rangle \\ u(4-v) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 3, 4 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^1(u) N_3^1(v) = \begin{cases} u(v-3) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 3, 4 \rangle \\ u(5-v) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 4, 5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^1(u) N_4^1(v) = \begin{cases} u(v-4) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 4, 5 \rangle \\ u(6-v) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 5, 6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^1(u) N_5^1(v) = \begin{cases} u(v-5) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 5, 6 \rangle \\ u(7-v) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 6, 7 \rangle \\ 0 & \text{jinak} \end{cases}$$

Nyní je třeba vzít funkci  $N_0^1$  na intervalu  $\langle 1, 2 \rangle$ . V předchozích případech je nutné nahradit člen  $u$  polynomem  $2-u$ . Poté se vypočítají všechny součiny

$$N_i^1(u) N_j^1(v), \text{ kde } i = 1, 2, 3, 4, 5, j = 0, 1, 2, 3, 4, 5.$$

Analogicky část pro  $N_j^1(v)$  zůstává stejná, jen  $N_i^1(u)$  se mění a to následujícím způsobem.

$$N_1^1(u) = \begin{cases} u-1 & \text{pro } u \in \langle 1, 2 \rangle \\ 3-u & \text{pro } u \in \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_2^1(u) = \begin{cases} u-2 & \text{pro } u \in \langle 2, 3 \rangle \\ 4-u & \text{pro } u \in \langle 3, 4 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^1(u) = \begin{cases} u-3 & \text{pro } u \in \langle 3, 4 \rangle \\ 5-u & \text{pro } u \in \langle 4, 5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^1(u) = \begin{cases} u-4 & \text{pro } u \in \langle 4, 5 \rangle \\ 6-u & \text{pro } u \in \langle 5, 6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_5^1(u) = \begin{cases} u-5 & \text{pro } u \in \langle 5, 6 \rangle \\ 7-u & \text{pro } u \in \langle 6, 7 \rangle \\ 0 & \text{jinak} \end{cases}$$

**Bázové funkce stupně dva**

$$N_0^2(u) N_0^2(v) = \begin{cases} \frac{u^2}{2} \frac{v^2}{2} & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 0, 1 \rangle \\ \frac{u^2}{2} (-v^2 + 3v - \frac{3}{2}) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 1, 2 \rangle \\ \frac{u^2}{2} \frac{(3-v)^2}{2} & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 2, 3 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_1^2(v) = \begin{cases} \frac{u^2}{2} \frac{(v-1)^2}{2} & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 1, 2 \rangle \\ \frac{u^2}{2} (-v^2 + 5v - \frac{11}{2}) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 2, 3 \rangle \\ \frac{u^2}{2} \frac{(4-v)^2}{2} & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 3, 4 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_2^2(v) = \begin{cases} \frac{u^2}{2} \frac{(v-2)^2}{2} & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 2, 3 \rangle \\ \frac{u^2}{2} (-v^2 + 9v - \frac{23}{2}) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 3, 4 \rangle \\ \frac{u^2}{2} \frac{(5-v)^2}{2} & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 4, 5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_3^2(v) = \begin{cases} \frac{u^2}{2} \frac{(v-3)^2}{2} & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 3, 4 \rangle \\ \frac{u^2}{2} (-v^2 + 9v - \frac{39}{2}) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 4, 5 \rangle \\ \frac{u^2}{2} \frac{(6-v)^2}{2} & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 5, 6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_4^2(v) = \begin{cases} \frac{u^2}{2} \frac{(v-4)^2}{2} & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 4, 5 \rangle \\ \frac{u^2}{2} (-v^2 + 11v - \frac{59}{2}) & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 5, 6 \rangle \\ \frac{u^2}{2} \frac{(7-v)^2}{2} & \text{pro } (u, v) \in \langle 0, 1 \rangle \times \langle 6, 7 \rangle \\ 0 & \text{jinak} \end{cases}$$

Nyní vezmeme interval  $\langle 1, 2 \rangle$ , kde je  $N_0^2(u)$  také nenulová. V předchozích výrazech se pouze změní polynom  $\frac{u^2}{2}$  na polynom  $(-u^2 + 3u - \frac{3}{2})$  a interval výpočtů bude  $(u, v) \in \langle 1, 2 \rangle \times \langle 0, 7 \rangle$ .

Pro další interval  $\langle 2, 3 \rangle$ , kde je  $N_0^2(u)$  také nenulová, nahradíme polynom  $(-u^2 + 3u - \frac{3}{2})$  polynomem  $\frac{(3-u)^2}{2}$ .

V dalším kroku je třeba vypočítat součiny  $N_1^2(u) N_j^2(v)$  pro  $j = 0, 1, 2, 3, 4$ .

$$N_1^2(u) = \begin{cases} \frac{(u-1)^2}{2} & \text{pro } u \in \langle 1, 2 \rangle \\ -u^2 + 5u - \frac{11}{2} & \text{pro } u \in \langle 2, 3 \rangle \\ \frac{(4-u)^2}{2} & \text{pro } u \in \langle 3, 4 \rangle \\ 0 & \text{jinak} \end{cases}$$

Členy  $N_j^2(v)$  pro  $j = 0, 1, 2, 3, 4$  jsou stejné jako pro funkci  $N_0^2(u)$  uvedené výše. Analogicky se spočítají součiny s bázovou funkcí  $N_2^2(u), N_3^2(u), N_4^2(u)$ . Výsledné plochy jsou v Maplu vykresleny na obr. 3.4, 3.5.

$$N_2^2(u) = \begin{cases} \frac{(u-2)^2}{2} & \text{pro } u \in \langle 2, 3 \rangle \\ -u^2 + 7u - \frac{23}{2} & \text{pro } u \in \langle 3, 4 \rangle \\ \frac{(5-u)^2}{2} & \text{pro } u \in \langle 4, 5 \rangle \\ 0 & \text{jinak} \end{cases}$$

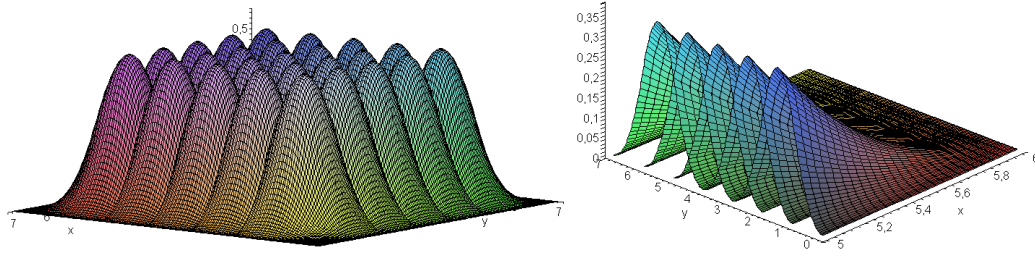
$$N_3^2(u) = \begin{cases} \frac{(u-3)^2}{2} & \text{pro } u \in \langle 3, 4 \rangle \\ -u^2 + 9u - \frac{39}{2} & \text{pro } u \in \langle 4, 5 \rangle \\ \frac{(6-u)^2}{2} & \text{pro } u \in \langle 5, 6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^2(u) = \begin{cases} \frac{(u-4)^2}{2} & \text{pro } u \in \langle 4, 5 \rangle \\ -u^2 + 11u - \frac{59}{2} & \text{pro } u \in \langle 5, 6 \rangle \\ \frac{(7-u)^2}{2} & \text{pro } u \in \langle 6, 7 \rangle \\ 0 & \text{jinak} \end{cases}$$

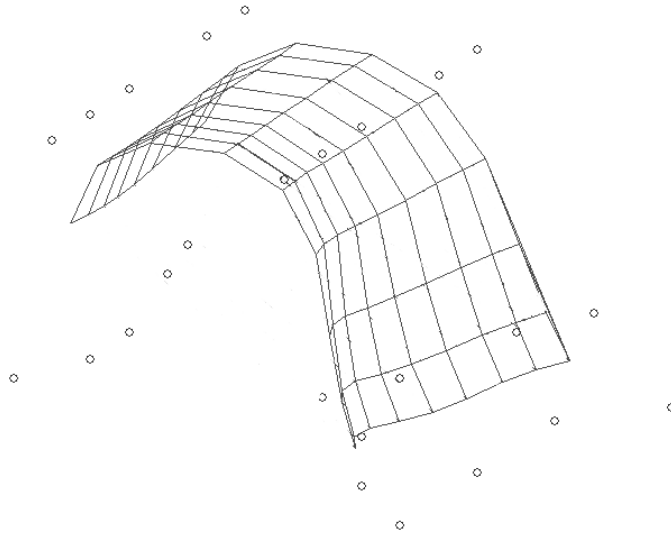
### 3.2.2 Neekvidistantní uzlový vektor

**Uzlový vektor**  $\mathbf{u} = \mathbf{v} = (0, 0, 0, 0.3, 0.6, 1, 1, 1)$

Vezmeme krajní hodnoty intervalu a zjistíme, které bázové funkce jsou nenulové a tedy které body působí na tvar plochy.



Obrázek 3.4: Bázové polynomy pro ekvidistantní uzlový vektor  $(0, 1, 2, 3, 4, 5, 6, 7)$ .



Obrázek 3.5: Výsledná plocha pro ekvidistantní uzlový vektor  $(0, 1, 2, 3, 4, 5, 6, 7)$ .

Pro  $(u, v) = (0, 0)$  jsou  $N_0^2(u) = 1, N_0^2(v) = 1$ . Všechny ostatní jsou nulové. Tedy:

$$N_0^2(u)N_0^2(v) = 1.$$

Pouze bod  $P_{00}$  má vliv na výslednou plochu pro hodnoty  $(u, v) = (0, 0)$  100 procent. Plocha tedy tímto bodem prochází.

Pro parametry  $(u, v) = (0, 1), (1, 0), (1, 1)$  je situace stejná. Bázové polynomy jsou:

$$N_0^2(u)N_4^2(u) = 1,$$

$$N_4^2(u)N_0^2(u) = 1,$$

$$N_4^2(u)N_4^2(u) = 1.$$

Body  $P_{04}, P_{40}, P_{44}$  leží na výsledné NURBS ploše. Obecně, je-li uzlový vektor tvaru:

$$\underbrace{0, 0, \dots, 0}_{stupen+1}, \dots, \underbrace{1, 1, \dots, 1}_{stupen+1},$$

poté NURBS plocha  $(m+1) \times (n+1)$  bodů prochází body  $P_{00}, P_{0n}, P_{m0}, P_{mn}$ . Toto tvrzení lze dokázat zobecněním předchozího případu.

### Bázové funkce stupně jedna

$$N_0^1(u) N_i^1(v) = 0$$

pro všechna  $i = 0, 1, \dots, 4$

$$N_1^1(u) N_1^1(v) = \begin{cases} \frac{100}{9}(0.3-u)(0.3-v) & \text{pro } (u, v) \in \langle 0, 0.3 \rangle \times \langle 0, 0.3 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_1^1(u) N_2^1(v) = \begin{cases} \frac{100}{9}(0.3-u)v & \text{pro } (u, v) \in \langle 0, 0.3 \rangle \times \langle 0, 0.3 \rangle \\ \frac{100}{9}(0.3-u)(0.6-v) & \text{pro } (u, v) \in \langle 0, 0.3 \rangle \times \langle 0.3, 0.6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_1^1(u) N_3^1(v) = \begin{cases} \frac{100}{9}(0.3-u)(v-0.3) & \text{pro } (u, v) \in \langle 0, 0.3 \rangle \times \langle 0.3, 0.6 \rangle \\ \frac{100}{12}(0.3-u)(1-v) & \text{pro } (u, v) \in \langle 0, 0.3 \rangle \times \langle 0.6, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_1^1(u) N_4^1(v) = \begin{cases} \frac{100}{12}(0.3-u)(v-0.6) & \text{pro } (u, v) \in \langle 0, 0.3 \rangle \times \langle 0.6, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

Analogicky k ekvidistantnímu uzlovému vektoru vezmeme všechny kombinace pro parametr  $u$ . To jsou následující polynomy:

$$N_2^1(u) = \begin{cases} \frac{10}{3}u & \text{pro } u \in \langle 0, 0.3 \rangle \\ \frac{10}{3}(0.6-u) & \text{pro } u \in \langle 0.3, 0.6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^1(u) = \begin{cases} \frac{10}{3}(u-0.3) & \text{pro } u \in \langle 0.3, 0.6 \rangle \\ \frac{10}{4}(1-u) & \text{pro } u \in \langle 0.6, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^1(u) = \begin{cases} \frac{10}{4}(u-0.6) & \text{pro } u \in \langle 0.6, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$



**Bázové funkce stupně dva**

$$N_0^2(u) N_0^2(v) = \begin{cases} \frac{10}{3}(0.3-u)\frac{10}{3}(0.3-v) & \text{pro } (u,v) \in \langle 0, 0.3 \rangle \times \langle 0, 0.3 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_1^2(v) = \begin{cases} \frac{100}{9}(0.3-u)^2 \left( \frac{100}{9}v(0.3-v) + \frac{100}{18}v(0.6-v) \right) & \text{pro } (u,v) \in \langle 0, 0.3 \rangle \times \langle 0, 0.3 \rangle \\ \frac{100}{9}(0.3-u)^2 \frac{100}{18}(0.6-v)^2 & \text{pro } (u,v) \in \langle 0, 0.3 \rangle \times \langle 0.3, 0.6 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_2^2(v) = \begin{cases} \frac{100}{9}(0.3-u)^2 \frac{100}{18}v^2 & \text{pro } (u,v) \in \langle 0, 0.3 \rangle \times \langle 0, 0.3 \rangle \\ \frac{100}{9}(0.3-u)^2 \left( \frac{100}{18}v(0.6-v) + \frac{100}{21}(1-v)(v-0.3) \right) & \text{pro } (u,v) \in \langle 0, 0.3 \rangle \times \langle 0.3, 0.6 \rangle \\ \frac{100}{9}(0.3-u)^2 \frac{100}{28}(1-v)^2 & \text{pro } (u,v) \in \langle 0, 0.3 \rangle \times \langle 0.6, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_3^2(v) = \begin{cases} \frac{100}{9}(0.3-u)^2 \frac{100}{21}(v-0.3)^2 & \text{pro } (u,v) \in \langle 0, 0.3 \rangle \times \langle 0.3, 0.6 \rangle \\ \frac{100}{9}(0.3-u)^2 \left( \frac{100}{28}(v-0.3)(1-v) + \frac{100}{16}(1-v)(v-0.6) \right) & \text{pro } (u,v) \in \langle 0, 0.3 \rangle \times \langle 0.6, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_4^2(v) = \begin{cases} \frac{100}{9}(0.3-u)^2 \frac{100}{16}(v-0.6)^2 & \text{pro } (u,v) \in \langle 0, 0.3 \rangle \times \langle 0.6, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

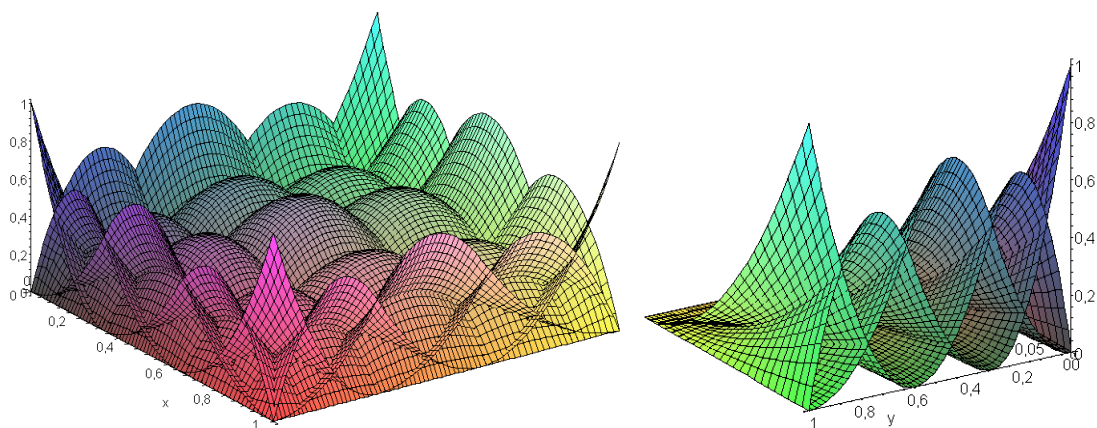
K získání všech bázových polynomů musíme vyměnit polynom v proměnné  $u$  za následující výrazy. Bázové funkce jsou vykresleny na obr. 3.6 a výsledná plocha je na obr. 3.7.

$$N_1^2(u) = \begin{cases} \frac{100}{9}u(1-u) + \frac{100}{18}u(0.6-u) & \text{pro } u \in \langle 0, 0.3 \rangle \\ \frac{100}{18}(0.6-u)^2 & \text{pro } u \in \langle 0.3, 0.6 \rangle \\ 0 & \text{jinak} \end{cases}$$

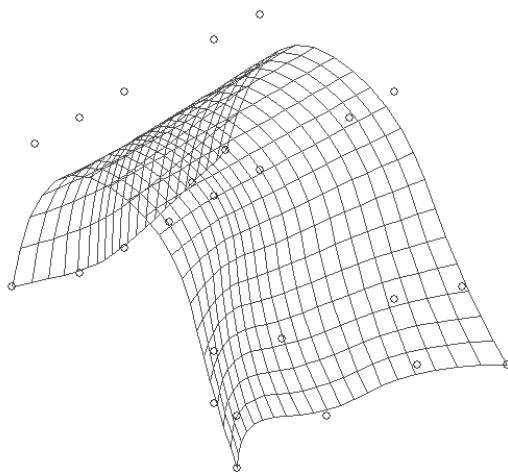
$$N_2^2(u) = \begin{cases} \frac{100}{18}u^2 & \text{pro } u \in \langle 0, 0.3 \rangle \\ \frac{100}{18}u(0.6-u) + \frac{100}{21}(u-0.3)(1-u) & \text{pro } u \in \langle 0.3, 0.6 \rangle \\ \frac{100}{28}(1-u)^2 & \text{pro } u \in \langle 0.6, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^2(u) = \begin{cases} \frac{100}{21}(u - 0.3)^2 & \text{pro } u \in \langle 0.3, 0.6 \rangle \\ \frac{100}{28}(u - 0.3)(1 - u) + \frac{100}{16}(u - 0.6)(1 - u) & \text{pro } u \in \langle 0.6, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^2(u) = \begin{cases} \frac{100}{16}(u - 0.6)^2 & \text{pro } u \in \langle 0.6, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$



Obrázek 3.6: Bázové polynomy pro vektory  $u = v = (0, 0, 0, 0.3, 0.6, 1, 1, 1)$



Obrázek 3.7: Výsledná plocha pro vektory  $u = v = (0, 0, 0, 0.3, 0.6, 1, 1, 1)$

**Uzlový vektor  $\mathbf{u} = \mathbf{v} = (0, 0, 0, 0.5, 0.5, 1, 1, 1)$**

Vnitřní uzel 0.5 se opakuje dvakrát, což je rovno stupni výsledné NURBS plochy. Podle vlastnosti 5 (str. 56) je spojitost v bodě  $S(0.5, 0.5)$  rovna nule, tedy plocha prochází bodem  $S(0.5, 0.5)$  a vytváří zde ostrý přechod. Podle obr. 3.8 je zřejmé, že existuje jediná bázová funkce, která zde má vliv 100 procent. Výslednou plochu vidíme na obr. 3.9.

**Bázové funkce stupně jedna**

$$N_0^1(u) N_i^1(v) = 0$$

pro všechna  $i = 0, 1, \dots, 4$

$$N_i^1(u) N_0^1(v) = 0$$

pro všechna  $i = 0, 1, \dots, 4$

$$N_1^1(u) N_1^1(v) = \begin{cases} (0.5 - u)(0.5 - v)/0.25 & \text{pro } (u, v) \in \langle 0, 0.5 \rangle \times \langle 0, 0.5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_1^1(u) N_2^1(v) = \begin{cases} (0.5 - u)v/0.25 & \text{pro } (u, v) \in \langle 0, 0.5 \rangle \times \langle 0, 0.5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_1^1(u) N_3^1(v) = \begin{cases} (0.5 - u)(1 - v)/0.25 & \text{pro } (u, v) \in \langle 0, 0.5 \rangle \times \langle 0.5, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_1^1(u) N_4^1(v) = \begin{cases} (0.5 - u)(v - 0.5) & \text{pro } (u, v) \in \langle 0, 0.5 \rangle \times \langle 0.5, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

Analogicky k ekvidistantnímu uzlovému vektoru bereme všechny kombinace pro parametr  $u$ . To jsou následující polynomy:

$$N_2^1(u) = \begin{cases} 2u & \text{pro } u \in \langle 0, 0.5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_3^1(u) = \begin{cases} 2(1 - u) & \text{pro } u \in \langle 0.5, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^1(u) = \begin{cases} 2(u - 0.5) & \text{pro } u \in \langle 0.5, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

**Bázové funkce stupně dva**

$$N_0^2(u) N_0^2(v) = \begin{cases} 16(0.5 - u)^2(0.5 - v)^2 & \text{pro } (u, v) \in \langle 0, 0.5 \rangle \times \langle 0, 0.5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_1^2(v) = \begin{cases} 32(0.5 - u)^2 v(0.5 - v) & \text{pro } (u, v) \in \langle 0, 0.5 \rangle \times \langle 0, 0.5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_2^2(v) = \begin{cases} 16v^2(0.5 - u)^2 & \text{pro } (u, v) \in \langle 0, 0.5 \rangle \times \langle 0, 0.5 \rangle \\ 16(0.5 - u)^2(1 - v)^2 & \text{pro } (u, v) \in \langle 0, 0.5 \rangle \times \langle 0.5, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_3^2(v) = \begin{cases} 32(0.5 - u)^2(v - 0.5)(1 - v) & \text{pro } (u, v) \in \langle 0, 0.5 \rangle \times \langle 0.5, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_0^2(u) N_4^2(v) = \begin{cases} 16(0.5 - u)^2(v - 0.5)^2 & \text{pro } (u, v) \in \langle 0, 0.5 \rangle \times \langle 0.5, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

A opět k získání všech bázových polynomů musíme vyměnit polynom v proměnné  $u$  za následující výrazy.

$$N_1^2(u) = \begin{cases} 8u(0.5 - u) & \text{pro } u \in \langle 0, 0.5 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_2^2(u) = \begin{cases} 4u^2 & \text{pro } u \in \langle 0, 0.5 \rangle \\ 4(1 - u)^2 & \text{pro } u \in \langle 0.5, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

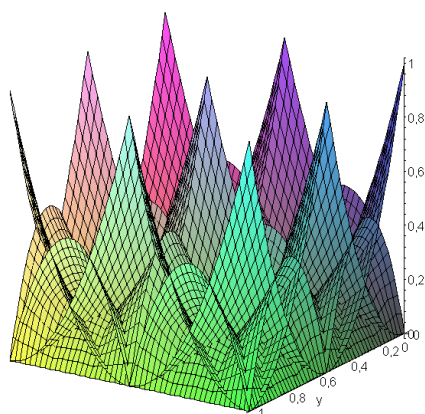
$$N_3^2(u) = \begin{cases} 8(u - 0.5)(1 - u) & \text{pro } u \in \langle 0.5, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

$$N_4^2(u) = \begin{cases} 4(u - 0.5)^2 & \text{pro } u \in \langle 0.5, 1 \rangle \\ 0 & \text{jinak} \end{cases}$$

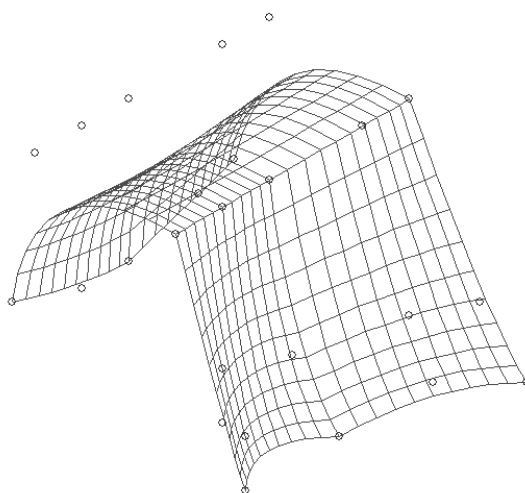
Bázové funkce jsou vykresleny na obr. 3.8 a výsledná plocha je vykreslena na obr. 3.9. Vzhledem k tomu, že řídicí body prostředního sloupce leží na přímce, vznikla zde hrana místo ostrého vrcholu.

**3.2.3 Ovládání NURBS ploch**

Ovládání NURBS ploch pomocí uzlového vektoru není jednoduché. Pro srovnání jsou zde uvedeny další možnosti práce – využití stupně, řídicích bodů a jejich vah. V poslední části je přehledně vysvětlena práce s uzlovými vektory s ukázkami výsledných ploch.



Obrázek 3.8: Bázové polynomy pro vektory  $u = v = (0, 0, 0, 0.5, 0.5, 1, 1, 1)$

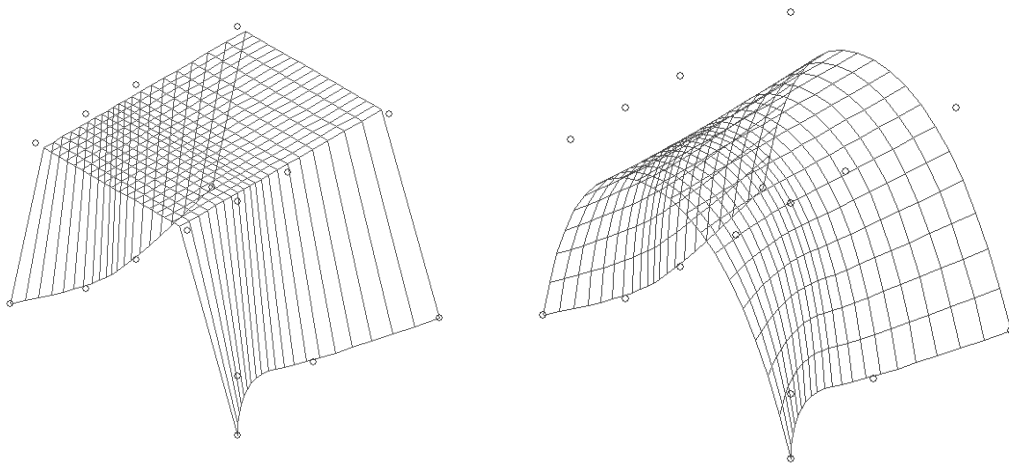


Obrázek 3.9: Výsledná plocha pro vektory  $u = v = (0, 0, 0, 0.5, 0.5, 1, 1, 1)$

### Vliv stupně plochy

Stupeň plochy je jediná změna, která je globální. Největší rozdíl je samozřejmě při stupni jedna (přímková plocha) a stupněm dva a vyššími. Rozdíl mezi vyššími stupni již není tak zřejmý, ale zvyšuje se hladkost plochy. Může se lépe pracovat s uzlovým vektorem a zvyšovat násobnost některých uzlů.

Na obr. 3.10 je vidět rozdíl mezi přímkovou a kvadratickou plochou. U křivek se zadával nejnižší stupeň křivky dva, neboť není potřeba zadávat úsečku jako NURBS křivku. Pro NURBS plochy povolíme i stupeň jedna, protože mnoho ploch technické praxe je přímkových - válec, kužel, přímkové konoidy, hyperboloid, atp.



Obrázek 3.10: Obecná plocha stupně jedna a dva

### Vliv polohy bodu

Změna polohy bodu je jedna z nejnázornějších. Změna je to pouze lokální, dojde ke změně plochy pouze v okolních bodech. Podle vlastnosti 4 na straně 56 je to pro bod  $P_{ij}$  interval parametru  $\langle u_i, u_{i+n+1} \rangle \times \langle v_i, v_{i+m+1} \rangle$ , kde  $m, n$  jsou stupně v obou směrech. Pro názornost jsem sestrojila obecnou plochu  $4 \times 4$  body (viz obr. 3.10) zadanou následovně:

Stupeň:

$$m = n = 2$$

Uzlové vektory:

$$\mathbf{u} = (0, 0, 0, 0.5, 1, 1, 1)$$

$$\mathbf{v} = (0, 0, 0, 0.5, 1, 1, 1)$$

řídící body:

$$1. \text{ sloupec: } (-1, 0, 6), (0, -2, 6), (0, -4, 6), (-1, -8, 6)$$

$$2. \text{ sloupec: } (0, 0, 0), (0, -2, 0), (0, -4, 0), (0, -8, 0)$$

$$3. \text{ sloupec: } (6, 0, 0), (6, -2, 0), (6, -4, 0), (6, -8, 0)$$

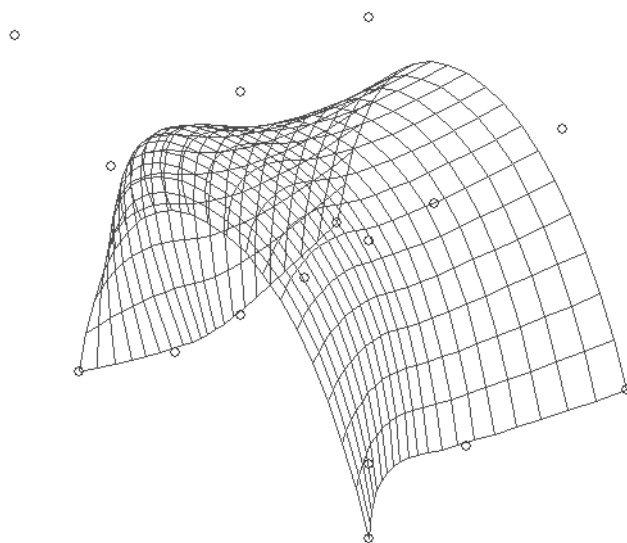
$$3. \text{ sloupec: } (8, 0, 6), (6, -2, 6), (7, -4, 6), (8, -8, 6)$$

Váhy:

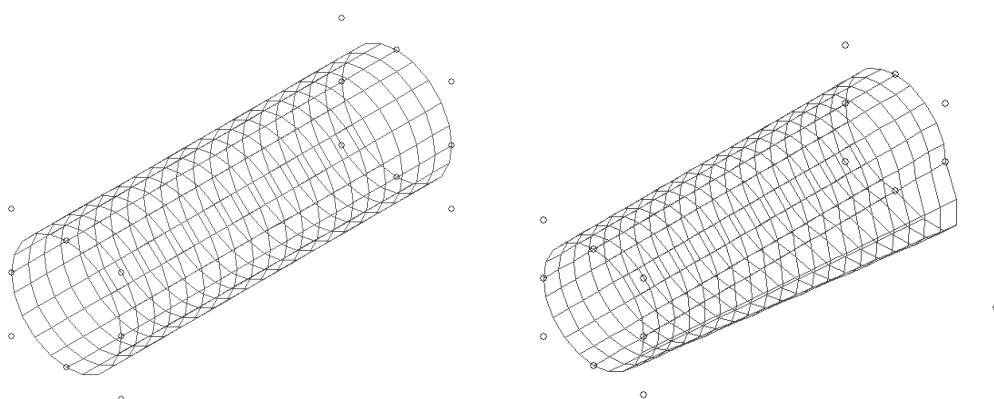
$$w_{ij} = 1 \text{ pro všechna } P_{ij}, i = 0, \dots, n, j = 0, \dots, m.$$

Na obr. 3.11 je zobrazeno posunutí bodu  $P_{11}$  – druhý sloupec, druhý bod. Souřadnice jsou změněny na  $(-5, -2, 0)$ . Velmi dobře je vidět posun plochy k bodu. Důležité je, že spojitost zůstává zachována.

Je nutné upozornit na speciální případy popsané u derivací NURBS křivek v části 3.1. V řídicím bodě  $P_i$  určíme spojitost  $C_0$  podle tvaru uzlového vektoru. Vzhledem k tomu, že řídicí body  $P_{i-1}$  a  $P_{i+1}$  leží na přímce, je spojitost  $C_1$ . Dojde-li k posunu některého z těchto bodů, je spojitost porušena. V případě válce nastane posunem bodu snížení spojitosti na  $C_0$  (viz obr. 3.12).



Obrázek 3.11: Vliv polohy bodu na tvar křivky

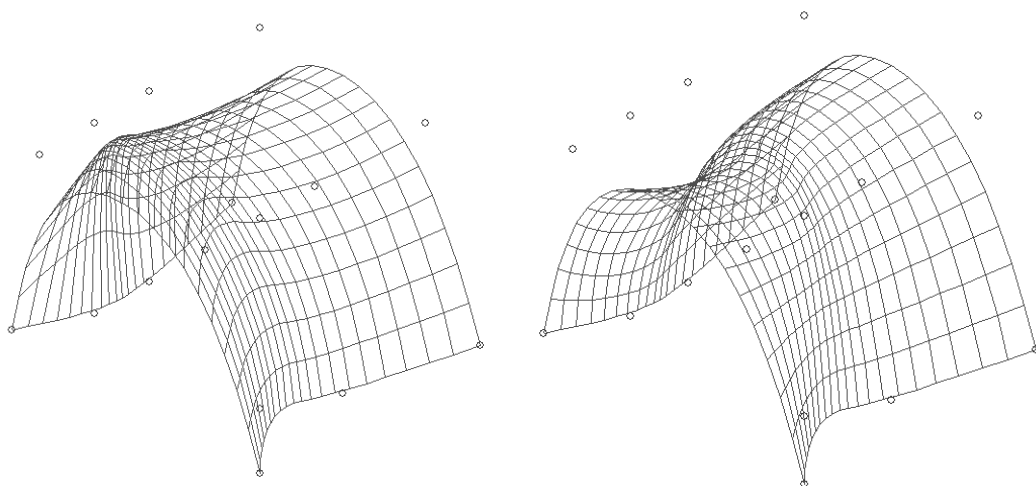


Obrázek 3.12: Vliv polohy bodu na tvar křivky – porušení spojitosti

### Vliv váhy

Všem bodům NURBS ploch jsou přiřazeny váhy. Implicitně jsou všechny nastaveny na hodnotu jedna. Je-li váha větší než jedna, zvětšuje se vliv daného bodu a plocha se k bodu více přibližuje. Je-li váha v intervalu  $(0, 1)$ , snižuje se vliv bodu a plocha se oddaluje. V mojí práci jsem záporné hodnoty nedovolila. V literatuře Piegl – Tiller (2002), Piegl (1989) se uvádí možnosti použití záporných vah pro konstrukci doplňkových oblouků. Pro uživatele mi tato možnost nepřipadá vhodná a dá se obejít jinou volbou zadání.

Na obr. 3.13 jsou vidět dvě modifikace základní obecné plochy zadané v předchozí části pomocí změny váhy.

Obrázek 3.13: Vliv váhy na tvar plochy,  $w_2 = 4.0$ ,  $w_2 = 0.2$ 

### Vliv uzlového vektoru

Nejobtížnější k ovládní je modifikace uzlového vektoru. V části 2.2 věnované B-spline křivkám je podrobně vysvětleno, jak se křivka mění v závislosti na uzlovém vektoru. Chování ploch je analogické. Nejdříve si uvedeme případ, kdy plocha prochází rohovými body, poté vezmeme změnu neekvidistantního uzlového vektoru a na závěr vytvoření bodů nespojitosti.

Je nutné si uvědomit, že každá dvojice součinů  $N_i^n(u)N_j^p(v)$  je spojena s jediným bodem  $P_{ij}$ , jehož vliv na tvar plochy pro vstupní parametry určuje.

Vezměme uzlové vektory  $\mathbf{u}, \mathbf{v}$ :

$$u = (0, 0, 0, 1/3, 2/3, 1, 1, 1)$$

$$v = (0, 0, 0, 1/3, 2/3, 1, 1, 1).$$

Pro parametry  $(u, v) = (0, 0)$  jsou  $N_0^2(u) = 1, N_0^2(v) = 1$ . Všechny další polynomiální funkce stupně nula jsou rovny nule. Existuje jediná nezáporná báze funkce

$$N_0^2(u)N_0^2(v) = 1,$$

která je spojena s bodem  $P_{00}$ . Tento bod leží na výsledné ploše, neboť báze funkce má v něm vliv 100 procent. Stejná situace nastává pro parametry  $(u, v) = (0, 1), (1, 0), (1, 1)$ . Nenulové báze funkce jsou:

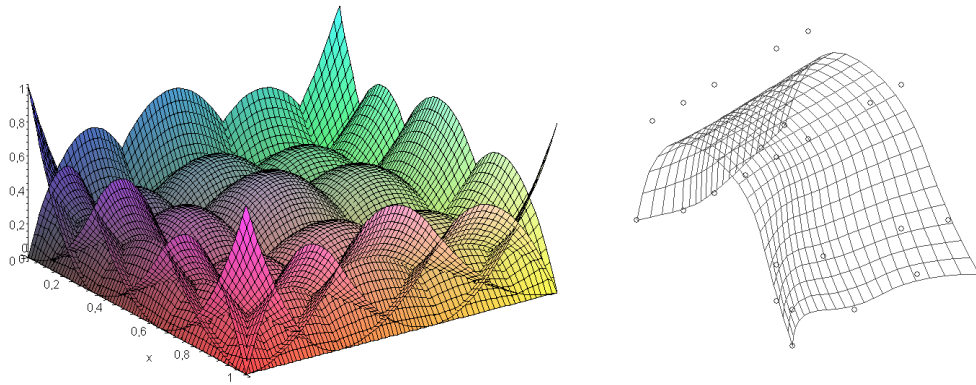
$$N_0^2(u)N_4^2(u) = 1, \quad N_4^2(u)N_0^2(u) = 1, \quad N_4^2(u)N_4^2(u) = 1.$$

Body  $P_{04}, P_{40}, P_{44}$  leží na výsledné ploše. Obecně, mají-li uzlové vektory tvar:

$$\underbrace{0, 0, \dots, 0}_{\text{degree}+1}, \dots, \underbrace{1, 1, \dots, 1}_{\text{degree}+1},$$

potom NURBS plocha složená z  $m \times n$  bodů prochází body  $P_{00}, P_{0n}, P_{m0}, P_{mn}$ . Důkaz je pouhým zobecněním předchozí úvahy. Báze funkce a výsledná plocha tohoto typu jsou znázorněny na obr. 3.14.



Obrázek 3.14: Neuniformní uzlový vektor  $(0, 0, 0, 1/3, 2/3, 1, 1, 1)$ 

Vezměme uzlové vektory  $\mathbf{u}, \mathbf{v}$ :

$$u = (0, 0, 0, 0.5, 0.5, 1, 1, 1)$$

$$v = (0, 0, 0, 0.5, 0.5, 1, 1, 1).$$

Vzhledem k vlastnosti 5 (str. 56) je spojitost plochy  $S(u, v)$  v bodech odpovídajících parametrům  $u = 0.5$  nebo  $v = 0.5$  nulová. Jak je vidět na obr. 3.15 body  $P_{02}, P_{20}, P_{22}, P_{24}, P_{42}$  leží na NURBS ploše. Vzhledem k tomu, že řídicí body druhého sloupce leží na přímce, vznikla zde hrana.

$$N_0^2(u)N_2^2(u) = 1 \quad N_2^2(u)N_0^2(u) = 1$$

$$N_2^2(u)N_2^2(u) = 1 \quad N_2^2(u)N_4^2(u) = 1$$

$$N_4^2(u)N_2^2(u) = 1.$$

Bázové funkce v těchto bodech jsou rovny jedné, což je patrné z grafu na obr. 3.15, kde jsou zřejmě viditelné lokální extrémy. Podle předchozí části plocha prochází také body  $P_{00}, P_{40}, P_{04}, P_{44}$ .

Nechť  $S(u, v)$  a  $\bar{S}(\bar{u}, \bar{v})$  jsou dvě NURBS plochy se stejnými parametry, ale s odlišnými uzlovými vektory:

$$u = (0, 0, 0, 0.2, 0.4, 1, 1, 1) \quad v = (0, 0, 0, 0.2, 0.4, 1, 1, 1).$$

$$\bar{u} = (0, 0, 0, 0.6, 0.8, 1, 1, 1) \quad \bar{v} = (0, 0, 0, 0.6, 0.8, 1, 1, 1).$$

Porovnejme rozdíly mezi plochami  $S, \bar{S}$ . Např. pro parametry  $(u, v) = (0.7, 0.7)$ . spočítáme báze polynomy.

**Plocha  $S(u, v)$**

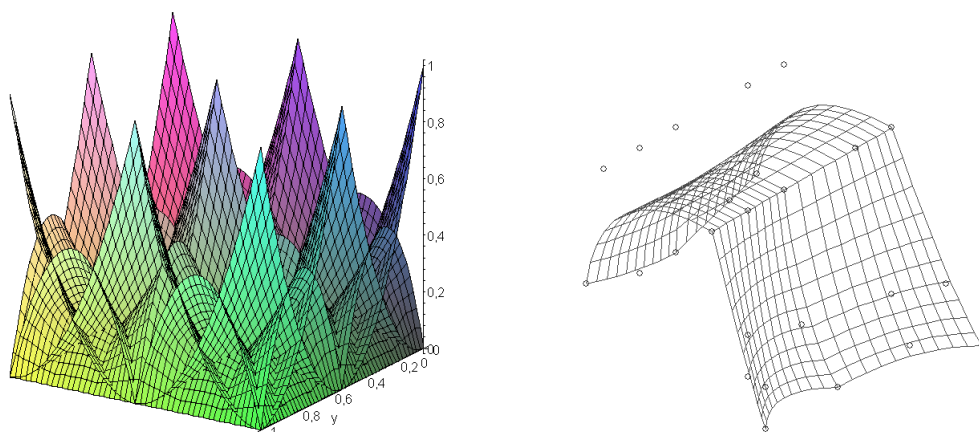
$$0.7 \in \langle 0.4, 1 \rangle \implies N_2^2(0.7), N_3^2(0.7), N_4^2(0.7) \neq 0 \implies$$

Body  $P_{ij}$ , kde  $i, j = 2, 3, 4$ , mají vliv na výslednou plochu.

**Plocha  $\bar{S}(u, v)$**

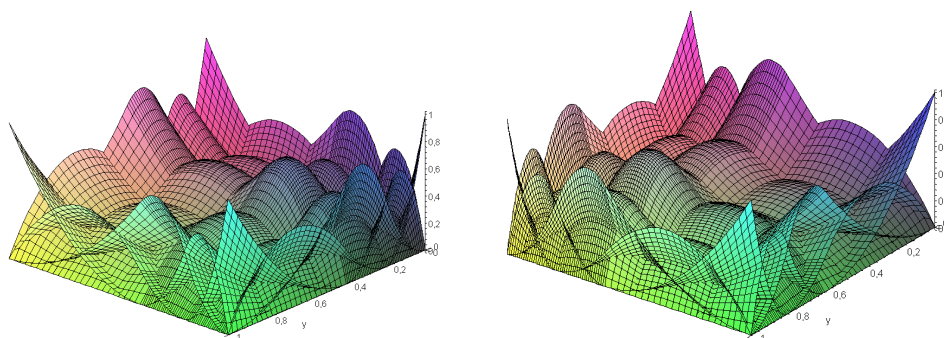
$$0.7 \in \langle 0.6, 0.8 \rangle \implies N_1^2(0.7), N_2^2(0.7), N_3^2(0.7) \neq 0 \implies$$

Body  $P_{ij}$ , kde  $i, j = 1, 2, 3$ , mají vliv na výslednou plochu.

Obrázek 3.15: Uzlový vektor  $(0, 0, 0, 0.5, 0.5, 1, 1, 1)$ 

Je zřejmé, že pro parametry  $(0.7, 0.7)$  se použijí různé řídicí body. Bod  $P_{22}$  má vliv 3.5 procent na plochu  $S$ , ale pro plochu  $\bar{S}$  je vliv 66 procent.

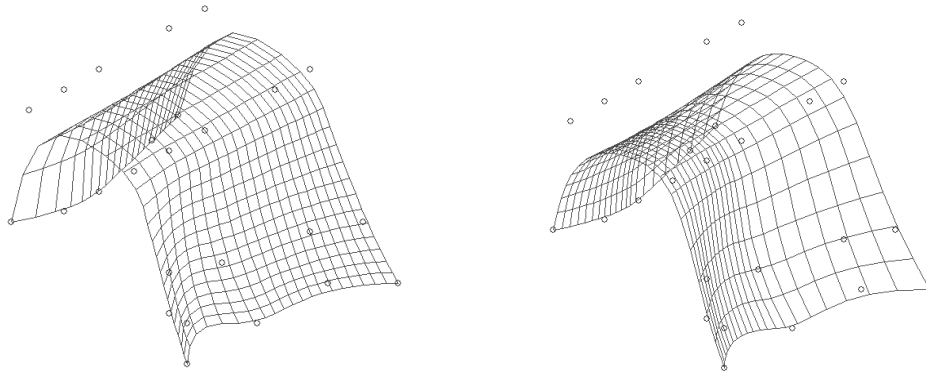
Názorně je to vidět na obr. 3.16, kde jsou bazové polynomy pro obě plochy. Na obr. 3.17 vidíme výsledné NURBS plochy. Ovládání pomocí uzlového vektoru je náročné, pro uživatele je vhodné k vytvoření bodů nespojitosti či zdůraznění určité části plochy.



Obrázek 3.16: Bazové funkce pro neuniformní uzlové vektory

### 3.3 NURBS plochy jako tenzorový součin

V literatuře (Piegl – Tiller (2002), Shene (Naposledy navštíveno 11. 12. 2006)) se lze setkat s pojmem *tensor product surface*, kterým se popisují plochy, avšak není vysvětleno jakým způsobem jednotlivé plochy vznikají. V této části proto popíšeme jednotlivá NURBS tělesa jako tenzorový součin. Teorie týkající se tenzorového součinu je popsána v části 1.1.4. Nejdříve ukážeme již uveřejněné přístupy a nesrovnalosti, na které jsme při jejich studiu narazili. Z tohoto důvodu jsme zavedli pojem kvazitenzorového součinu, který je z geometrického i matematického pohledu korektní.



Obrázek 3.17: Výsledné NURBS plochy pro neuniformní uzlové vektory

### 3.3.1 Tenzorový součin mezi Grassmannovými prostory

Racionální křivky jsou prvky Grassmannových prostorů. Dalším možným způsobem definování tenzorového součinu může být přímá definice mezi dvěma Grassmannovými prostory.

Grassmannovy prostory jsou rozšířením afinních prostorů o základní operace. Teorii a jejich vzájemné propojení lze nalézt například v Goldman (2000). Základní myšlenka je převzata z klasické mechaniky, kde jsou body (umístění) a vektory (síly). K bodům můžeme také přidat objekty (hmotu), na kterou budou síly působit. Tak vznikají hmotné body, v našem případě je nazýváme body s vahami.

V Grassmannově prostoru jsou tyto body dány dvojicí  $(P, w)$ , kde  $P$  je bod afinního prostoru a  $w$  je skalár. Body s nulovou vahou nazýváme vektory a značíme  $(v, 0)$ . Násobení bodu skalárem definujeme tak, že pozice bodu zůstává stejná, pouze vynásobíme váhu.

$$\begin{aligned} c(wP, w) &= (cwP, cw) \quad c \neq 0 \\ c(wP, w) &= (0, 0) \quad c = 0 \end{aligned} \quad (3.16)$$

Operace sčítání dvou bodů s vahami je dána jako:

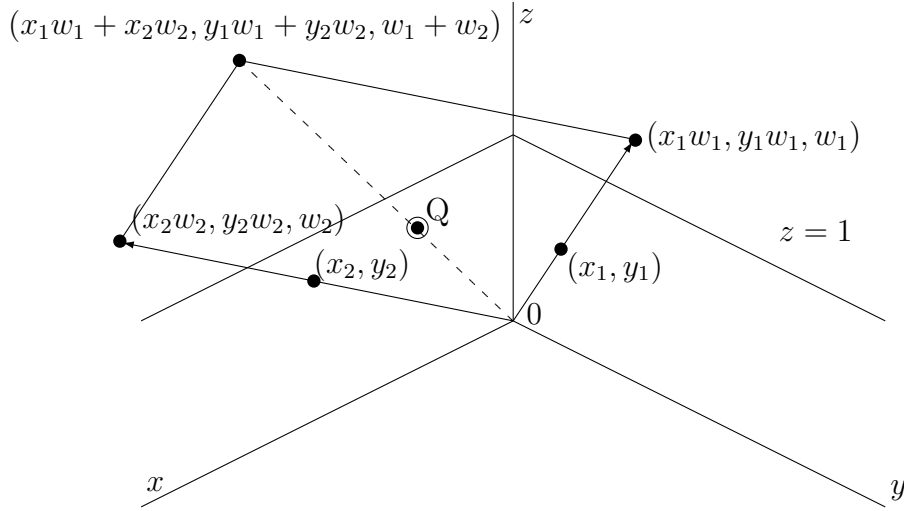
$$(w_1P_1, w_1) + (w_2P_2, w_2) = (w_1P_1 + w_2P_2, w_1 + w_2) \quad (3.17)$$

kde  $w_1 + w_2 \neq 0$ . Pokud  $w_1 + w_2 = 0$ , dostáváme vektor  $(m(P_2 - P_1), 0)$ .

Geometricky lze vysvětlit zavedení operací jako práci s vektory vytvořenými ze základních bodů a libovolného pevně zvoleného počátku. Součet dvou bodů s vahami (umístěných vektorů) je roven součtu těchto vektorů po složkách. Na obr.3.18 je znázorněn případ pro rovinné body s vahami. Jsou dány dva body s vahami  $w_1, w_2$ . Sečteme vektory, které určují a průsečík  $Q$  výsledného vektoru s rovinou  $z = 1$  je hledaný součet.

Sčítání bodů je komutativní a asociativní. Pro násobení bodu skalárem platí distributivní zákony.

$$c((w_1P_1, w_1) + (w_2P_2, w_2)) = c(w_1P_1, w_1) + c(w_2P_2, w_2) \quad (3.18)$$



Obrázek 3.18: Reprezentace bodu s váhou v Grassmannově prostoru

Pro práci s vektory platí:

$$\begin{aligned}(v, 0) + (u, 0) &= (v + u, 0) \\ c(v, 0) &= (cv, 0)\end{aligned}\tag{3.19}$$

a součet bodu s vektorem je definován jako:

$$(wP, w) + (v, 0) = (wP + v, w)\tag{3.20}$$

Grassmannovy prostory bodů s váhami slouží ke konstrukci racionálních nebo po částech racionálních křivek a ploch. Vezmeme posloupnosti řídících bodů s jejich váhami  $(w_0P_0, w_0), \dots, (w_nP_n, w_n)$ . Vybereme vhodné bázové funkce (Bézier, B-spline) a potom definujeme křivku  $R(t)$  jako množinu bodů v afinním prostoru, která je určena rovnicí:

$$R(t) = \sum_{i=0}^n B_i(t)(w_iP_i, w_i)\tag{3.21}$$

Úpravou podle pravidel pro váhové body dostaneme známý tvar pro racionální křivku:

$$R(t) = \frac{\sum_{i=0}^n w_i B_i(t) P_i}{\sum_{j=0}^n w_j B_j(t)}\tag{3.22}$$

Problém nastává, pokud je některá z vah nulová. V Grassmannově prostoru existují body s nulovou váhou – vektory. Proto můžeme vyjádřit křivku v Grassmannově prostoru i s nulovými váhami jako:

$$R(t) = \sum_{w_i \neq 0} B_i(t)(w_iP_i, w_i) + \sum_{w_j = 0} B_j(t)(v_j, 0),\tag{3.23}$$

což můžeme opět s použitím pravidel pro sčítání a násobení skalárem upravit na známý tvar:

$$R(t) = \sum_{i=0}^n \left( \frac{w_i B_i(t)}{\sum_{j=0}^n w_j B_j(t)} \right) (P_i + v(t)), \quad (3.24)$$

kde  $v(t)$  reprezentuje sumaci pro nulové body. Podrobný popis lze nalézt v Goldman (2000).

Obecně máme dva Grassmannovy vektorové prostory, jejichž prvky jsou body afinního prostoru reprezentované pomocí vektorů umístěných v počátku. Mezi nimi jsme zavedli v předchozím odstavci operace sčítání a násobení skalárem.

Nechť je dán vektorový prostor  $V$  a v něm báze vektory  $\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_n$ . Hledáme tenzorový součin  $V \otimes V$ . Vezmeme dva vektory  $\mathbf{u}, \mathbf{v} \in V$ . V souřadnicích máme:

$$\mathbf{u} = u_0 \mathbf{e}_0 + \dots + u_n \mathbf{e}_n \quad (3.25)$$

$$\mathbf{v} = v_0 \mathbf{e}_0 + \dots + v_n \mathbf{e}_n \quad (3.26)$$

Pak můžeme zapsat tenzorový součin vektorů  $\mathbf{u} \otimes \mathbf{v}$  jako:

$$\mathbf{u} \otimes \mathbf{v} = u_0 v_0 \mathbf{e}_0 \otimes \mathbf{e}_0 + u_0 v_1 \mathbf{e}_0 \otimes \mathbf{e}_1 + \dots + u_n v_n \mathbf{e}_n \otimes \mathbf{e}_n = \sum_{i=0}^n \sum_{j=0}^n u_i v_j \mathbf{e}_i \otimes \mathbf{e}_j$$

V našem případě se budeme pohybovat v prostoru  $V_4$ , což je prostor trojrozměrných bodů s váhami. Máme tedy  $V_4 \otimes V_4$ . Dimenze výsledného prostoru je součin obou dimenzí, tedy  $V_{16}$ . Naším úkolem je tedy najít vnoření prostoru  $V_{16}$  do  $V_4$  a následně do  $V_3$ .

Toto vnoření však není jednoznačně dáno. Musí se stanovit výrazná omezení. Dalším problémem v tomto přístupu je provádění operací s body, resp. vektory, které je vyjadřují. Například posunutí bodů, které je nezbytné při hledání bodů válce, otáčení pro anuloid, atd. Z tohoto důvodu je tento přístup pouze formální a přispívá k pochopení významu vah pro jednotlivé řídicí body.

### 3.3.2 Odvození tenzorového součinu pomocí vektorových prostorů a přidružených lineárních forem

Následující přístup je popsán v jednom z prvních děl týkajících se B-spline profesora Carl de Boora (de Boor (1976)).

Nechť  $U$  je vektorový prostor funkcí, všechny jsou definovány na množině  $X$  do reálných čísel. A nechť  $V$  je také vektorový prostor funkcí definovaný na množině  $Y$  do  $\mathbb{R}$ . Pro každé  $u \in U$  a  $v \in V$  pravidlo

$$w(t, s) = u(t)v(s), \text{ pro } \forall (t, s) \in X \times Y \quad (3.27)$$

definuje funkci na  $X \times Y$ , kterou nazýváme tenzorovým součinem  $u$  a  $v$  a značíme  $u \otimes v$ .

Množina všech lineárních kombinací funkcí na  $X \times Y$  typu  $u \otimes v$  pro libovolné  $u \in U$  a  $v \in V$  se nazývá tenzorový součin  $U$  s  $V$  a značíme jej  $U \otimes V$ .

**Příklad:** Vezměme vektorový prostor  $V_k$  polynomů stupně  $k$ , které jsou definovány na  $\mathbb{R}$ . Druhý vektorový prostor  $U_l$  je prostorem polynomů stupně jedna definovaných také na  $\mathbb{R}$ . Tenzorový součin  $U \otimes V$  je vektorový prostor  $W_{k,l}$  polynomů dvou proměnných stupňů nejvýše  $k$ , resp.  $l$  definovaných na rovině  $X \times Y = \mathbb{R}^2$ .

Nechť máme dvě lineární formy  $f$ , resp.  $g$  na  $U$ , resp.  $V$ . Poté můžeme definovat  $f \otimes g$  jako:

$$f \otimes g \left( \sum_i u_i \otimes v_i \right) = \sum_i f(u_i)g(v_i) \quad \text{pro } \forall u_i \otimes v_i. \quad (3.28)$$

Pokud je  $f \otimes g$  zobrazení na  $U \otimes V$ , které splňuje rovnici (3.28), potom je  $f \otimes g$  lineární formou na  $U \otimes V$ . Toto tvrzení musíme nejdříve zdůvodnit.

Vezměme libovolnou funkci  $w$  definovanou na  $X \times Y$  a necht'  $\bar{s}$  je prvkem množiny  $Y$ . Potom

$$w_{\bar{s}}(t) = w(t, \bar{s}) \quad \text{pro } \forall t \in X \quad (3.29)$$

definuje funkci  $w_{\bar{s}}$  na  $X$ , kterou nazýváme  $\bar{s}$ -řez funkce  $w$ .

Pokud vyjádříme  $w$  jako

$$w = \sum_i u_i \otimes v_i, \quad \text{pro některá } (u_i, v_i) \in U \times V$$

pak z definice tenzorového součinu můžeme vypočítat  $w_{\bar{s}}(t)$  jako:

$$w_{\bar{s}}(t) = \sum_i u_i(t)v_i(\bar{s}) = \sum_i v_i(\bar{s})u_i(t). \quad (3.30)$$

Z rovnice (3.30) vyplývá, že  $w_{\bar{s}} \in U$ , můžeme tedy spočítat  $f(w_{\bar{s}})$ , kde  $f$  je lineární forma na  $V$ , jako:

$$f(w_{\bar{s}}) = \sum_i v_i(\bar{s})(f(u_i)). \quad (3.31)$$

Označme  $w_f$  množinu všech  $f$ -řezů funkce  $w$ , což je funkce definovaná na  $Y$  jako:

$$w_f(s) = f w_s, \quad \text{pro } \forall s \in Y. \quad (3.32)$$

Výraz  $w_f(s)$  z předchozí rovnice závisí pouze na  $w$  a  $f$ , ale ne na vyjádření  $w$  pomocí  $u_i$  a  $v_i$ . Také můžeme vyjádřit  $w_f$  z rovnice (3.31) jako:

$$f(w_{\bar{s}}) = \sum_i f(u_i)v_i, \quad \text{pokud } w = \sum_i u_i \otimes v_i. \quad (3.33)$$

Rovnice (3.33) ukazuje, že  $w_f$  je členem  $V$ , tedy můžeme spočítat výraz  $g(w_f)$  jako:

$$g(w_f) = g \left( \sum_i f(u_i)v_i \right) = \sum_i f(u_i)g(v_i) \quad (3.34)$$

Z toho vyplývá, že výraz  $\sum_i f(u_i)g(v_i)$  závisí pouze na  $f, g$  a funkci  $w = \sum_i u_i \otimes v_i$ . Důsledkem tohoto tvrzení je, že pro libovolné lineární formy  $f$ , resp.  $g$  na  $U$ , resp.  $V$ , rovnice (3.28) určuje lineární formu na  $U \otimes V$  a tato lineární forma splňuje:

$$(f \otimes g)w = f(w_g) = g(w_f), \quad \forall w \in U \otimes V. \quad (3.35)$$

**Tenzorový součin**

Mějme dány dva vektorové prostory  $U, V$ .  $U$  obsahuje B-spline funkce stupně  $n$  na uzlovém vektoru  $\mathbf{t}$ .  $V$  je vektorový prostor B-spline funkcí stupně  $m$  na uzlovém vektoru  $\mathbf{s}$ . Naším úkolem je najít jejich tenzorový součin  $w \in U \otimes V$  pro parametry  $(\bar{t}, \bar{s})$ .

Máme tedy  $w = \sum_i \sum_j \alpha_{ij} N_{ij}$ , kde  $\alpha_{ij}$  je libovolná matice koeficientů a platí:

$$N_{ij}(t, s) = N_i^m(t) N_j^n(s). \quad (3.36)$$

Podle teorie uvedené výše je tento vztah odvozen z

$$w(t, s) = \sum_i \left( \sum_j \alpha_{ij} N_j^n(s) \right) N_i^m(t) = f(w_g) \quad (3.37)$$

nebo

$$w(t, s) = \sum_j \left( \sum_i \alpha_{ij} N_i^m(t) \right) N_j^n(s) = g(w_f), \quad (3.38)$$

kde zobrazení  $f$ , resp.  $g$  jsou vyčísľující funkce pro B-spline funkce.

**B-spline plochy**

Nyní ukážeme, co znamenají vztahy (3.28) – (3.33) pro B-spline funkce, resp. B-spline křivky.

Mějme dány dva vektorové prostory B-spline funkcí.  $U$  je vektorový prostor stupně  $n$  na uzlovém vektoru  $\mathbf{t}$  a  $V$  je vektorový prostor stupně  $m$  na uzlovém vektoru  $\mathbf{s}$ . Dále necht'  $f$  je lineární forma na  $U$  a  $g$  je lineární forma na  $V$ , které můžeme nazvat vyčísľující funkce. Nyní vezmeme dva prvky prostorů  $U$  a  $V$  –  $N_i^n(t)$  a  $N_j^m(s)$  a příslušné vyčísľující funkce  $f, g$  určují souřadnice bodů B-spline křivky.

$$f : U \rightarrow R : f(\bar{t}) = \sum_{i=0}^p P_i N_i^n(\bar{t}) \quad (3.39)$$

$$g : V \rightarrow R : g(\bar{s}) = \sum_{j=0}^q Q_j N_j^m(\bar{s}) \quad (3.40)$$

Podle (3.37) můžeme zapsat výslednou plochu jako:

$$(f \otimes g)(w) = \sum_{i=0}^p \sum_{j=0}^q R_{ij} N_i^n(t) N_j^m(s), \quad (3.41)$$

kde  $R_{ij}$  jsou souřadnice řídících bodů  $R_{ij} = (x_{ij}, y_{ij}, z_{ij})$  výsledné B-spline plochy a jejich odvození bude ukázáno později, což de Boor (1976) neuvádí.

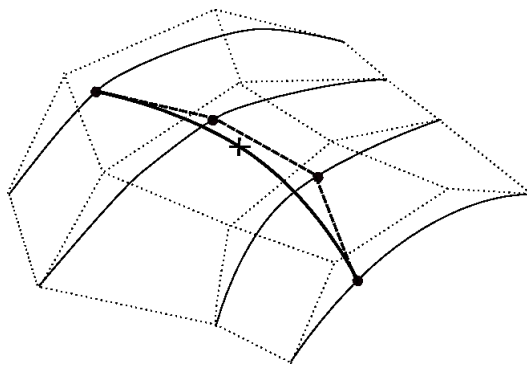
Výslednou B-spline plochu můžeme zapsat maticově jako:

$$w(u, v) = [N_i^n(t)]^T [R_{ij}] [N_j^m(s)], \quad (3.42)$$

kde  $[N_i^n(t)]$  je  $(1) \times (p+1)$  řádkový vektor,  $[N_j^m(s)]$  je  $(q+1) \times (p)$  sloupcový vektor a  $[R_{ij}]$  je  $(p+1) \times (q+1)$  matice třídimenzionálních bodů. Tato rovnice odpovídá obecné rovnici (1.12) uvedené v části Teoretická východiska na str. 13.

Zde je názorně vidět matematická podstata výpočtu bodů na ploše vzniklé tenzorovým součinem. Pro pevný parametr vypočítáme na každém sloupci (resp. řádku) pomocí první vyčíslovací funkce jeden bod. Ty tvoří nové hodnoty, na které aplikujeme druhou vyčíslovací funkci. Výsledkem je bod plochy.

Pro pevný parametr aplikujeme f-řezy (resp. g-řezy) na B-spline plochu. Tím se vypočítá systém rovnoběžek (resp. poledníků). Tyto křivky se též nazývají isoparametrické.



Obrázek 3.19: Výpočet bodu na ploše

Tento, na první pohled názorný způsob má však velkou mezeru. Tenzorový součin je definován jako zobrazení dvou či více vektorových prostorů do tělesa. Rovnice (3.27) definuje tenzorový součin do jiného prostoru, což není korektní. Pro názornost je sice tento způsob ideální, avšak nejedná se o čistou definici tenzorového součinu, kterou zavedeme pomocí kvazitenzorového součinu.

### 3.3.3 Kvazitenzorový součin

Při následujícím rozboru budeme pracovat s vektorovými prostory B-spline funkcí. To, že se jedná o vektorový prostor není na první pohled zřejmé. Částečné ověření uvádí Cline (Naposledy navštíveno 20. 2. 2007). Zde uvedeme pouze stěžejní poznatky.

#### PP-prostory

Mějme dán prostor  $S(K, D, C)$  po částech polynomiálních funkcí, které jsou popsány pomocí tří polí:

1.  $K$ : *pole uzlů*, pro  $m \geq 0$ ,  $K = (z_1, z_2, \dots, z_m)$ , kde  $z_1 < z_2 < \dots < z_m$ ,
2.  $D$ : *pole stupňů*,  $D = (d_1, d_2, \dots, d_m)$ , kde  $d_i \geq 0$  pro  $i = 0, 1, \dots, m$ ,
3.  $C$ : *pole spojitosti*,  $C = (k_1, k_2, \dots, k_m)$ , kde  $k_i \geq -1$  pro  $i = 1, \dots, m-1$ .

Funkce  $f \in S(K, D, C)$  tehdy a jen tehdy:



- $f$  je polynom stupně  $d_0$  na intervalu  $(-\infty, z_1)$ ,
- $f$  je polynom stupně  $d_i$  na intervalu  $(z_i, z_{i+1})$  pro  $i = 1, \dots, m-1$ ,
- $f$  je polynom stupně  $d_m$  na intervalu  $(z_m, \infty)$ ,
- $f$  je diferencovatelná a  $k_i$ -tá derivace je spojitá v  $z_i$  pro  $i = 0, \dots, m$ .

*Poznámka 16.* Prostor  $S(K, D, C)$  musí být grupou. Cline (Naposledy navštíveno 20. 2. 2007) neuvádí důkaz tohoto tvrzení. Pro grupu musí existovat nulový prvek a prvek opačný. Nulovým prvkem bude pravděpodobně nulová funkce. Problém nastává při opačném prvku. Jak určit operaci sčítání, musí být oba prostory definovány nad stejným uzlovým vektorem, jak se sčítají jednotlivé funkce?

Podle mého názoru by řešením mohlo být využití symetrie bázevých funkcí (vlastnost 6, část 2.2.2, str. 36). Opačný prvek by byl prvek s opačným průběhem.

**Linearita** Je zřejmé, že součet dvou prvků  $f, g \in S(K, D, C)$  a součin  $\lambda f$ , kde  $\lambda \in \mathbb{R}$ , jsou také prvky prostoru  $S$ .

**Báze a dimenze** K nalezení báze spline vektorového prostoru budeme používat tzv. *plus funkci*.

$$(x-a)_+^k = \begin{cases} (x-a)^k & x \geq a \\ 0 & x < a \end{cases} \quad (3.43)$$

Dimenzi prostoru  $S$  lze vypočítat jako:

$$1 + d_0 + \sum_{i=1}^m (d_i - k_i)$$

**Příklad:** Vezměme prostor, který má konstantní stupeň a konstantní spojitost, tj.  $d_i = d, k_i = k$ , pro  $i = 0, 1, \dots, m$  (resp.  $m-1$ ).

Báze:  $1, x, \dots, x^d, (x-z_1)_+^{k+1}, \dots, (x-z_1)_+^d, \dots, (x-z_m)_+^{k+1}, \dots, (x-z_m)_+^d$

Dimenze:  $m(d-k) + d + 1$

### Aplikace tenzorového součinu na válcovou plochu

V dalších úvahách budeme vycházet z teorie uvedené v části 1.1.4 a nejdříve podrobně popíšeme případ válcové plochy.

#### Vyjádření kružnice

řídící body:  $P_0, \dots, P_8$

váhový vektor:  $\mathbf{w} = (1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1)$

uzlový vektor:  $\mathbf{t} = (0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1)$

stupeň: 2

Rovnice je potom:

$$k(t) = \sum_{i=0}^8 P_i w_i N_i^2(t) \quad (3.44)$$

Souřadnice kružnice v bázi  $N_i^2(t)$ ,  $i = 0, \dots, 8$  jsou  $\omega = (1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1)$ .

### Vyjádření přímky

řídící body:  $P_0, P_1$

váhový vektor:  $\mathbf{w} = (1, 1)$

uzlový vektor:  $\mathbf{s} = (0, 0, 1, 1)$

stupeň: 1

Rovnice je potom:

$$u(t) = \sum_{j=0}^1 Q_j w'_j N_j^1(s) \quad (3.45)$$

Souřadnice přímky v bázi  $N_0^1(s), N_1^1(s)$  jsou  $\bar{\omega} = (1, 1)$ .

Množina  $N_8 = \langle N_0^2(t), N_1^2(t), \dots, N_8^2(t) \rangle$  všech lineárních kombinací funkcí

$$N_0^2(t), N_1^2(t), \dots, N_8^2(t)$$

nad uzlovým vektorem  $\mathbf{t}$  je vektorovým prostorem množin křivek druhého stupně. Jeho prvkem je například vektor  $(1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1)$ , který představuje množinu všech kružnic (*abstraktní* kružnici). Vektor  $(1, 0.5, 1, 0.5, 1, 0.5, 1, 0.5, 1)$  je pak obloukem *abstraktní* elipsy apod. Konkrétní kružnici, přesněji řečeno  $i$ -té souřadnice bodů konkrétní kružnice, pak dostaneme skalárním součinem vektoru  $\omega$  a vektoru  $j$ -tých souřadnic řídících bodů, tj.

$$k_j(t) = \mathbf{P}_j \cdot \omega = (p_{0j}, p_{1j}, \dots, p_{8j}) \begin{pmatrix} \omega_0 N_0^2(t) \\ \omega_1 N_1^2(t) \\ \vdots \\ \omega_8 N_8^2(t) \end{pmatrix} = \sum_{i=0}^8 p_{ij} \omega_i N_i^2(t) \quad (3.46)$$

Analogicky dostaneme prostor všech *abstraktních* přímek jako množinu

$$N_2 = \langle N_0^1(s), N_1^1(s) \rangle$$

všech lineárních kombinací funkcí

$$N_0^1(s), N_1^1(s).$$

Jejím prvkem je například vektor

$$\bar{\omega} = (1, 1)$$

který vyjadřuje *abstraktní* přímku. Reprezentaci  $j$ -tých souřadnic konkrétní přímky dostaneme skalárním součinem jako:

$$u_j(t) = \mathbf{Q}_j \cdot \bar{\omega} = (q_{0j}, q_{1j}) \begin{pmatrix} \bar{\omega}_0 N_0^1(t) \\ \bar{\omega}_1 N_1^1(t) \end{pmatrix} = \sum_{k=0}^1 q_{kj} \bar{\omega}_k N_k^1(s). \quad (3.47)$$

Tenzorovým součinem prostorů  $N_8$ ,  $N_2$  je množina všech bilineárních forem  $N_8 \times N_2 \rightarrow \mathbb{R}$ . V případě kružnice a přímky se tedy jedná o formu:

$$c_j = \bar{\omega} \cdot \mathbf{R} \cdot \omega^T = (1, 1) \begin{pmatrix} r_{00} & \cdots & r_{08} \\ r_{10} & \cdots & r_{18} \end{pmatrix} \begin{pmatrix} 1 \\ \frac{\sqrt{2}}{2} \\ \vdots \\ 1 \end{pmatrix}, \quad (3.48)$$

která představuje  $j$ -tou souřadnici *abstraktního* válce.

Je zřejmé, že další postup ke konkrétní implementaci touto cestou, tj. využitím definice 1.33 tenzorového součinu (str. 11), by byl značně těžkopádný. Definici 1.33 tedy zobecníme.

*Definice 3.1.* Nechtě  $U, V, Z$  jsou vektorové prostory nad tělesem  $T$ . Zobrazení  $\varphi : U \times V \rightarrow Z$  se nazývá **bilineární**, jestliže pro každé pevné  $\mathbf{u} \in U$  je zobrazení

$$\varphi(\mathbf{u}, -) : V \rightarrow Z \text{ lineární}$$

a pro každé pevné  $\mathbf{v} \in V$  je zobrazení

$$\varphi(-, \mathbf{v}) : U \rightarrow Z \text{ lineární.}$$

Totéž můžeme zapsat jako:

$$\begin{aligned} \varphi(a\mathbf{u}_1 + b\mathbf{u}_2, \mathbf{v}_1) &= a\varphi(\mathbf{u}_1, \mathbf{v}_1) + b\varphi(\mathbf{u}_2, \mathbf{v}_1) \\ \varphi(\mathbf{u}_1, a\mathbf{v}_1 + b\mathbf{v}_2) &= a\varphi(\mathbf{u}_1, \mathbf{v}_1) + b\varphi(\mathbf{u}_1, \mathbf{v}_2), \end{aligned}$$

kde  $\mathbf{u}_1, \mathbf{u}_2 \in U$ ,  $\mathbf{v}_1, \mathbf{v}_2 \in V$  a  $a, b \in T$ .

Analogicky jako u bilineárních forem definujeme operace sčítání a násobení skalárem. Potom platí věta:

**Věta 3.3.** *Prostor všech bilineárních zobrazení  $\varphi : U \times V \rightarrow Z$  tvoří vektorový prostor.*

*Definice 3.2.* Vektorový prostor bilineárních zobrazení z věty 3.3 nazýváme **kvazitenzorový součin** prostorů  $U$  a  $V$ .

### Válcová a kuželová plocha

Aplikací předešlých úvah můžeme zapsat válcovou či kuželovou plochu jako:

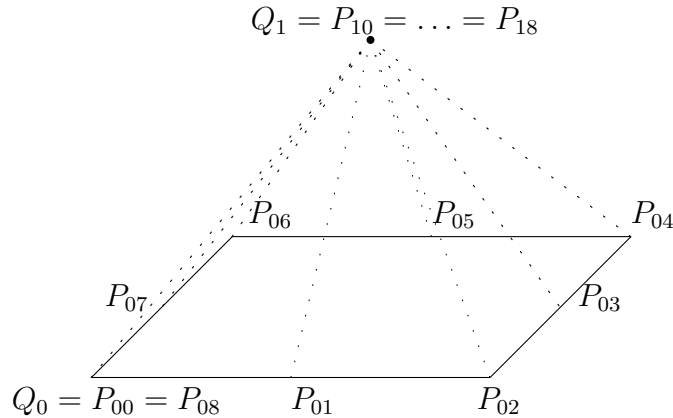
$$V(s, t) = k(s) \cdot u(t) = \bar{\omega}[\mathbf{PQ}]\omega \quad (3.49)$$

kde výraz  $[\mathbf{PQ}]$  je nutné odvodit. Matici  $\mathbf{P}$  tvoří řídicí body kružnice (dle def. 1.27 na str. 10 matice  $1 \times 9$  nad množinou bodů). Analogicky matici  $\mathbf{Q}$  tvoří řídicí body přímky (matice  $1 \times 2$ ). Matice

$$\mathbf{R} = [\mathbf{PQ}]$$

je matice bodů typu  $2 \times 9$ , kde řádky jsou řídicí body kružnic, sloupce řídicí body přímek a je tedy tvaru:



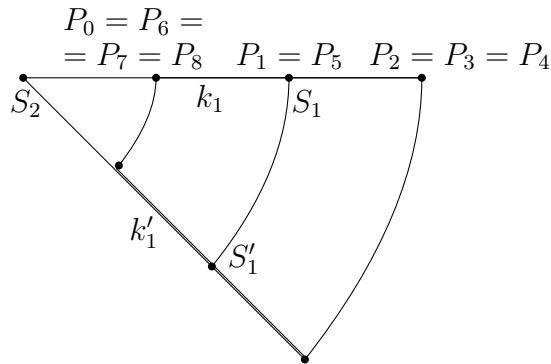


Obrázek 3.21: Řídicí body pro kužel

Anuloid dostaneme jako tenzorový součin vyjádření dvou kružnic v daným bázích:

$$A(s, t) = k_1(s) \cdot k_2(t) = \bar{\omega}[\mathbf{PQ}]\omega, \quad (3.52)$$

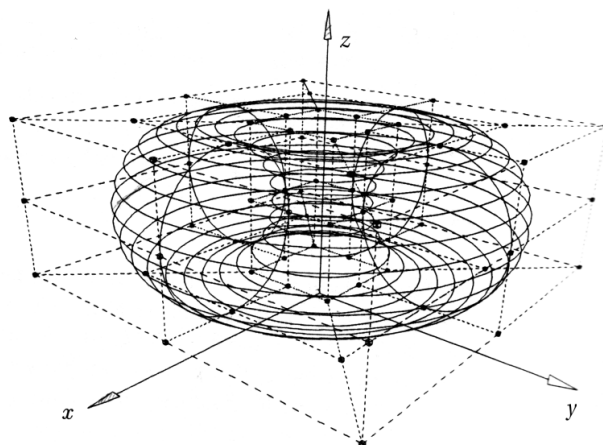
kde součin  $[\mathbf{PQ}]$  vychází ze souřadnic vstupních bodů. Pro klasický anuloid je nutné umístit kružnice tak, aby jejich roviny byly k sobě kolmé a střed kružnice otáčení  $k_1$  ležel na jednom z řídicích bodů pevné kružnice  $k_2$ . Další body jsou vypočítány postupným otáčením kružnice  $k_1$  o 45 stupňů. Každý řídicí bod hybné kružnice se otočí kolem středu pevné kružnice o úhel 45 stupňů v rovině rovnoběžné s rovinou pevné kružnice (viz obr. 3.22).



Obrázek 3.22: Řídicí body pro anuloid (pohled shora)

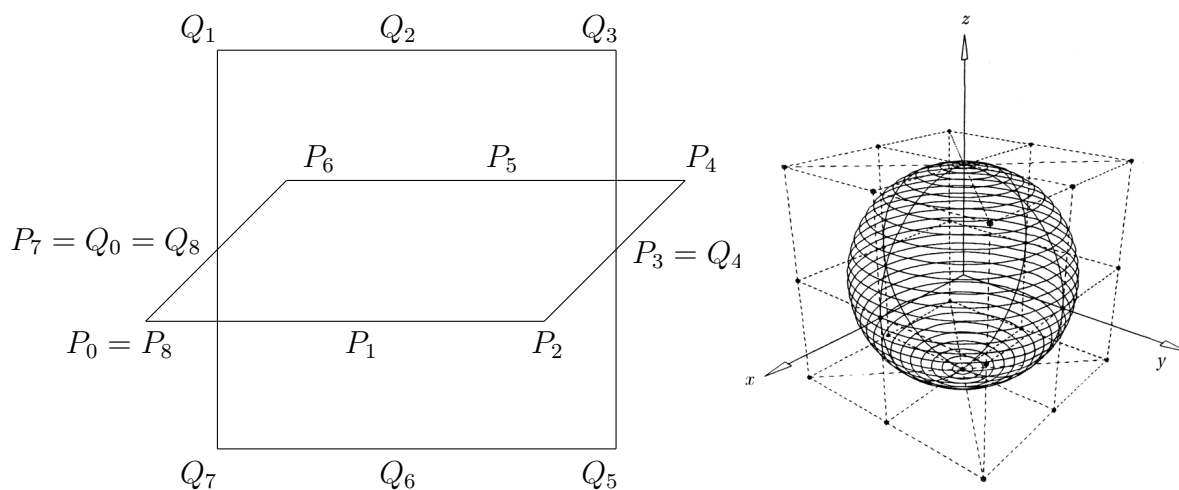
## 2. Kulová plocha

Vyjádření kulové plochy jako tenzorového součinu dvou kružnic je stejné jako v případě anuloidu. Jediným rozdílem je odvození řídicích bodů. Kružnice leží opět v rovinách vzájemně kolmých, ale mají dva společné řídicí body (viz obr. 3.24) a shodný poloměr. Zbývající body kontrolní sítě dostaneme posunutím jedné kružnice o vektor velikosti po-



Obrázek 3.23: Síť řídicích bodů anuloidu

loměru v obou směrech kolmých k rovině dané kružnice. Podle popisu uvedeného na obr. 3.24 se posouváme řídicí body  $P_i$  o vektory  $\overrightarrow{Q_0Q_1}$  a  $\overrightarrow{Q_0Q_7}$ .



Obrázek 3.24: Řídicí body pro kulovou plochu

### 3. Obecná plocha

Nechť jsou dány dvě obecné křivky  $l_1, l_2$  s různým počtem bodů, s různými váhami, stupněm a uzlovým vektorem. Jedinou podmínkou je jejich společný počáteční bod. Jednu křivku zvolíme za pevnou, například  $l_1$ . Body jednotlivých sloupců získáme posunutím druhé křivky (jejích řídicích bodů) o vektor určený na pevné křivce dvěma sousedními řídicími body. Na obr. 3.25 jsou dány dvě obecné křivky dané třemi řídicími body a je zde vyznačeno posunutí bodů, které vytvoří síť řídicích bodů  $R_{ij}$  pro výpočet výsledné plochy.

**Matematický zápis**

Křivka  $l_1$ :

řídící body:  $P_0, \dots, P_m$

váhový vektor:  $\mathbf{w} = (w_0, \dots, w_m)$

uzlový vektor:  $\mathbf{t}$

stupeň:  $p$

Vyjádření NURBS křivky je:

$$l_1(t) = \sum_{i=0}^m P_i w_i N_i^p(t) \quad (3.53)$$

Křivka  $l_2$ :

řídící body:  $Q_0, \dots, Q_n$

váhový vektor:  $\mathbf{w}' = (w'_0, \dots, w'_n)$

uzlový vektor:  $\mathbf{s}$

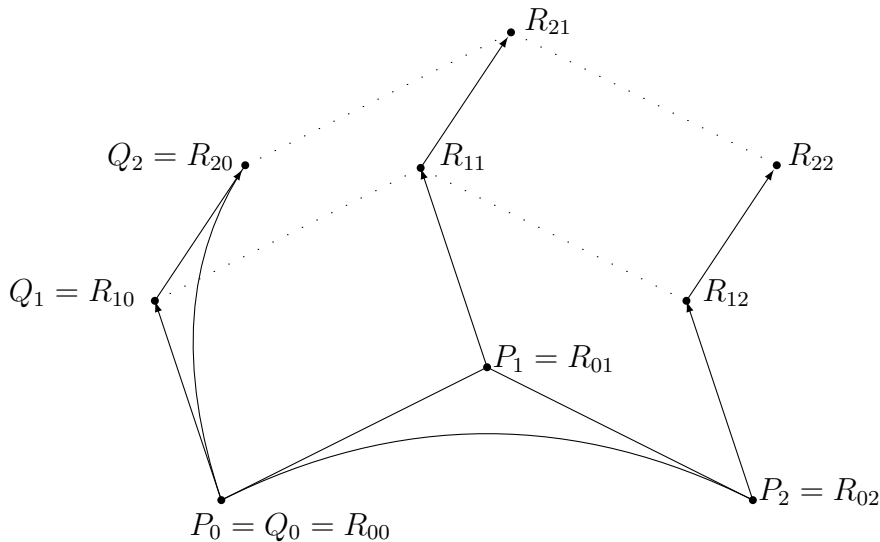
stupeň:  $q$

Můžeme psát:

$$l_2(s) = \sum_{j=0}^n Q_j w'_j N_j^q(s) \quad (3.54)$$

Označme  $R_{ij}$  síť řídících bodů výsledné plochy vytvořenou pomocí uvedeného postupu. Aplikací výše uvedených teoretických rozborů můžeme psát výslednou plochu jako:

$$S(t, s) = \sum_{i=0}^m \sum_{j=0}^n R_{ij} w_i w'_j N_i^p(t) N_j^q(s) \quad (3.55)$$



Obrázek 3.25: Vytvoření sítě bodů plochy jako tenzorového součinu křivek

### 3.4 Programové zpracování NURBS objektů

V následující části je uvedena implementace NURBS objektů – křivky, obecné plochy, základních a rotačních těles. Je zde podrobně popsán objektový návrh řešení, postup při hledání základních parametrů pro NURBS plochy. Důraz je kladen na výstižný popis použitých metod a také na kritická a problémová místa algoritmu. Všechny tyto postupy byly naprogramovány v C++ a implementovány do softwaru Fem Dev, což je testovací prostředí komerčního CAD systému RFEM 3D. V roce 2006 obsahoval nový release softwaru RFEM 3D implementované NURBS křivky a v letošním roce budou přidány plochy a tělesa. Všechny algoritmy jsou důsledně testovány, aby byly co nejvíce stabilní a odladěné.

Implementace a postupy obecné NURBS plochy jsem publikovala v Procházková – Procházka (2006a), další práce týkající se kužele a válce byla publikována na mezinárodní konferenci SCG (viz Procházková – Procházka (2006b)).

Uvedené algoritmy jsou určeny výhradně pro studijní účely, neboť autorská práva na jejich využití vlastní firma Fem Consulting.

#### 3.4.1 NURBS křivky

Výpočet obecného bodu NURBS křivky je základní úlohou. Křivka je zadána řídicími body, uzlovým a váhovým vektor, stupněm křivky. Pro parametr  $t$  algoritmus počítá bod křivky  $C(t)$ . K výpočtu se používá deBoorův algoritmus, který je popsán v části 2.2.2 (str. 37).

Algoritmus výpočtu:

##### 1. Ověření vstupních podmínek

- Stupeň křivky  $n$  se zadává mezi 2 a 16 (je možné i od jedné, potom kreslí lomenou čáru).
- Všechny váhy musí být kladná reálná čísla.
- Počet řídicích bodů je nejméně roven stupni křivky plus jedna.
- Uzly tvoří neklesající posloupnost kladných reálných čísel.
- Prvních a posledních  $n + 1$  složek uzlového vektoru může být shodných, vnitřní složky se mohou opakovat nejvýše  $n$ -krát, kde  $n$  je stupeň křivky.
- Počet uzlů je roven počtu bodů plus stupeň křivky plus jedna.
- Uzlový vektor není tvořen stejnými čísly.
- Počet vah je stejný jako počet bodů.

##### 2. Projektivizace

Vytvoříme nové pole čtyřrozměrných bodů, které bude obsahovat homogenní souřadnice původních bodů. Každou souřadnici bodu vynásobíme váhou a čtvrtou souřadnici bude tvořit samotná váha. Princip projektivizace je popsán v části 2.2.3.



```

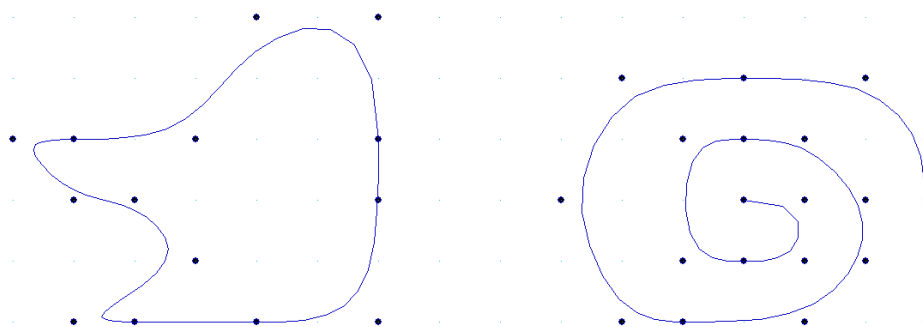
bod4D.x = bod3D.x * vaha;
bod4D.y = bod3D.y * vaha;
bod4D.z = bod3D.z * vaha;
bod4D.w = vaha;

```

3. De Boorův algoritmus pro zhomogenizované body (str. 37, část 2.2.2).
4. Souřadnice výsledného bodu jsou v homogenních souřadnicích. Pro vykreslení je nutné provést zpětnou projektivizaci. Tedy první tři souřadnice vydělíme čtvrtou souřadnicí, která udává váhu výsledného bodu.
5. Vykreslení křivky

### Ukázka křivek

Na obr. 3.26 jsou vykresleny obecné NURBS křivky.



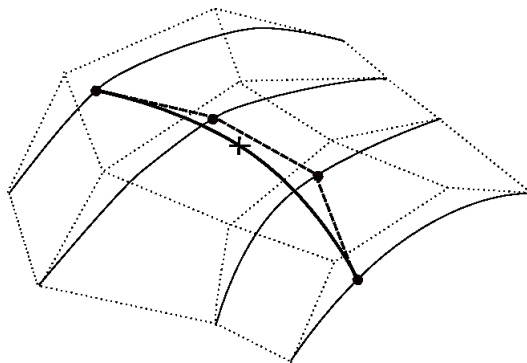
Obrázek 3.26: Ukázka výsledných NURBS křivek

### 3.4.2 Obecná NURBS plocha

Obecná NURBS plocha je dána sítí řídicích bodů spolu s jejich váhami. Dále je nutné znát řádkový a uzlový vektor a stupně křivek v obou směrech. Toto zadání je uživatelsky nepříjemné a nepoužívá se. Ve své práci jsem vybrala pro zadávání ploch určení jejích okrajových křivek. Program dogeneruje vnitřní body automaticky.

Princip programování obecné NURBS plochy spočívá v tom, že rozdělíme výpočet na dva výpočty pro NURBS křivky, které už známe z předchozí kapitoly. Při výpočtu postupujeme po sloupcích. Body v každém sloupci bereme jako body na obecné NURBS křivce a pro ni vypočítáme bod pro parametr  $u$ . V každém sloupci vznikne nový bod, jak je vidět na obr. 3.27.

Tyto body jsou řídicími body pro novou NURBS křivku. Pro parametr  $v$  se vypočítá výsledný bod NURBS plochy  $S(u, v)$ . Parametry  $u, v$  z intervalu  $\langle 0, 1 \rangle \times \langle 0, 1 \rangle$  určí body na výsledné ploše a plochu lze zobrazit.



Obrázek 3.27: Výpočet bodu na ploše

### Programové zpracování

Celé vykreslení NURBS ploch probíhá v rámci třídy **SNurbs**. Výpočet bodů plochy se provádí metodou **param**, která jako parametr dostává hodnoty  $(u, v) \in \langle 0, 1 \rangle \times \langle 0, 1 \rangle$ . Výstupem jsou body na ploše, které se předávají k následnému vykreslení. Další dvě použité třídy jsou **CBod** a **SBod**. **CBod** zadává bod třemi souřadnicemi. Výstup metody **param** je třídy **CBod**. Vstupní řídicí body plochy jsou třídy **SBod**, která má čtyři parametry. První tři jsou souřadnice bodu a čtvrtá souřadnice udává váhu.

```
class CBod{          class SBod{
    double x;         double x;
    double y;         double y;
    double z;         double z;
}                    double w;
                    }
```

### Algoritmus

**Vstup:** řídicí body s váhami třídy **SBod** v poli **Sarray**, počet řádků, počet sloupců, uzlové vektory pro řádky a sloupce - **knot1**, **knot2**, stupeň plochy pro řádky a sloupce - **degree1**, **degree2**, parametry  $u, v$

**Výstup:** bod NURBS plochy zadané předchozími podmínkami

Základem výpočtu je metoda **param**. Pseudokódem ji můžeme zapsat jako:

```
CBod SNurbs::param(double u, double v)
{
    for (sl = 0; sl < sloupec; sl++)
        Surftest1(sl, degree1, radek, knot_length1, knot1, Sarray);

    Surftest2(degree2, sloupec, knot_length2, knot2);
    Koeficient1(u, degree1, radek, knot1, &pocetOpak1, &koefUzlu1);

    for (sl = 0; sl < sloupec; sl++){
        for (i = 0; i < radek; i++){
```

```

    Homogenizace(sl, i, Sarray, pomBod);
    novyBod = DeBoor1(radek, degree1, degree2, knot_length1,
        knot_length2, knot1, pomBod, pocetOpak1, koefUzlu1, u);
}
}
// vypocet pro body novyBod
Koeficient2(v, degree2, sloupec, knot2, &pocetOpak2, &koefUzlu2);
vyslednyBod = DeBoor2(sloupec, degree1, degree2, knot_length1,
    knot_length2, knot2, novyBod, pocetOpak2, koefUzlu2, v);
return CBod(vyslednyBod);
}

```

### Popis použitých metod

**Koeficient1** - vypočítá, ve kterém intervalu uzlového vektoru **knot1** vstupní hodnota *u* leží a jaká je její násobnost.

**Koeficient2** - vypočítá, ve kterém intervalu uzlového vektoru **knot2** vstupní hodnota *v* leží a jaká je její násobnost.

**Homogenizace** - provede homogenizaci zadaných bodů.

**deBoor1** - výpočet bodu NURBS křivky pro jednotlivé sloupce pro parametr *u*, vrací body v poli **NovyBod**.

**DeBoor2** - výpočet bodu NURBS křivky z bodů v poli **NovyBod** vypočtených metodou **deBoor1** pro parametr *v*.

**Surftest1** - kontrola vstupních údajů pro **deBoor1**.

**Surftest2** - kontrola vstupních údajů pro **deBoor2**.

### Kritická místa

V literatuře jsou uvedeny vždy celkové algoritmy. V žádné knize jsem nenašla diskuzi o následujících kritických místech. Proto je zde uvedeno podrobné řešení.

#### 1. Krajní hodnoty uzlových intervalů

Pro potřeby technické praxe je uzlový vektor omezen na případ, kdy prvních  $n + 1$  členů uzlových vektorů je rovno nule a posledních  $n + 1$  složek uzlových vektorů je roven jedné. ( $n$  je stupeň křivky.) NURBS plocha prochází svými rohovými body, podle vlastnosti uvedené v části 3.2.2.

V krajních hodnotách dochází ke kolizi. V deBoorově algoritmu se objevuje záporný index bodu a výsledek není správný. Řešení spočívá v úpravě metod **Koeficient1** a **Koeficient2**. Pro hodnoty nula a jedna se pevně stanoví správně hodnoty, které de-Boor algoritmus zpracuje. Metoda **Koeficient1** vypadá následovně, **Koeficient2** se sestaví analogicky.

```

int SNURBS::Koeficient1(double u, int degree1, int radek, double
    *knot1, int *pocetOpak1, int *koefUzlu1)
{
    if (u == 0){

```

```

    *pocetOpak1 = degree1 + 1;
    *koefUzlu1 = degree1 + 1;
}else if (u~== 1){
    *pocetOpak1 = degree1 + 1;
    *koefUzlu1 = radek + degree1;
}else{
    najdi interval uzloveho vektoru, kde u~lezi
    urci nasobnost uzlu
u~}
}

```

## 2. Vstupní podmínky

Dalším úskalím je vhodně určit vstupní podmínky. V části věnované implementaci NURBS křivek jsou uvedeny podmínky, které musí být splněny také pro jednotlivé křivky sloupců, které v metodě **DeBoor1** procházíme.

Metoda **Surftest1** obsahuje následující podmínky:

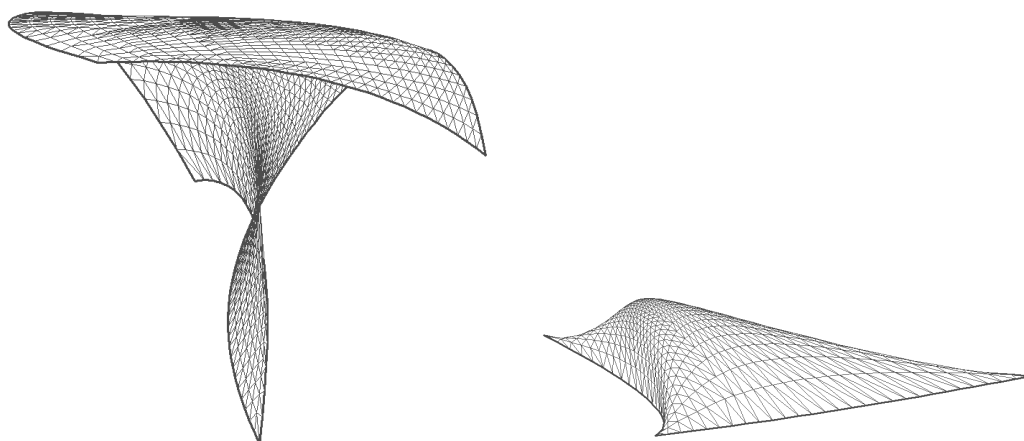
- je zadána pravidelná síť bodů,
- omezení stupně plochy pro **degree1** – interval  $\langle 1, 17 \rangle$ ,
- všechny váhy řídicích bodů musí být kladné,
- počet řídicích bodů ve sloupci musí být nejméně roven řádkovému stupni plus jedna,
- uzlový vektor **knot1** tvoří neklesající posloupnost,
- prvních a posledních **degree1+1** složek uzlového vektoru **knot1** je stejných, vnitřní uzly se můžou opakovat nejvýše **degree1**-krát.

Pro metodu **Surftest2** se některé podmínky změnil a jsou následující:

- omezení stupně plochy pro **degree2** – interval  $\langle 1, 17 \rangle$ ,
- počet sloupců musí být nejméně roven sloupcovému stupni plus jedna, neboť z metody **deBoor1** se vrací tolik bodů, kolik je sloupců, ty pak vstupují do metody **deBoor2**,
- uzlový vektor **knot2** tvoří neklesající posloupnost,
- prvních a posledních **degree2+1** složek uzlového vektoru **knot2** je stejných, vnitřní uzly se mohou opakovat nejvýše **degree2**-krát.

## Výsledné modely

Na obr. 3.28 je ukázka obecných NURBS ploch.



Obrázek 3.28: Ukázka obecné NURBS plochy

### 3.4.3 Implementace NURBS těles

Dalším krokem v implementaci NURBS objektů jsou předdefinovaná tělesa – kužel, válec, koule, anuloid, rovina, hranol a rotační tělesa. Zadání těchto objektů je konvenční. Uživatel má možnost na rozdíl od analytického zadání určit také počet rovnoběžek a poledníků či úhel rozevření tělesa. Každý objekt lze jednoduše upravovat – měnit polohu bodů, jejich váhy, uzlový vektor.

NURBS tělesa jsou teoreticky popsána v předchozí části 3.3. Zde uvedeme způsob implementace s důrazem na popis použitých metod a kritických či problémových míst algoritmů.

#### Objektový návrh řešení

Pro obecné NURBS plochy byla vytvořena třída **SNurbs**. Tělesa budou implementována jako samostatné třídy, které jsou potomky třídy **SNurbs** a to z důvodu přístupu metody **param** pro výpočet obecných bodů ploch.

Systém implementace pro každé těleso je stejný. Uživatel zadá vstupní hodnoty pro dané těleso, v konstruktoru příslušné třídy se vstup zkontroluje metodou **test** a pokud je vše v pořádku, tak se metodou **spoctiParametry** vypočítají základní parametry pro obecnou plochu (řídící body, váhy, uzlové vektory). Poté využitím dědičnosti můžeme použít metodu **param** z rodičovské třídy **SNurbs** a plochu vykreslit.

Následující části se postupně věnují jednotlivým tělesům. Pro každé těleso je popsána metoda **spoctiParametry** včetně dalších metod, které používá. Následně jsou uvedeny kritická místa algoritmu. Na závěr jsou uvedeny ukázky výsledných těles a možnosti jejich úpravy.

#### Třída **CylinderNurbs** – kužel a válec

**Programové zpracování** Kužel i válec se zadávají pomocí tří bodů - středu dolní podstavy (**bodStredu**), bodu na obvodu podstavy (**bodPolomeru**) a středu horní podstavy

pro válec, resp. vrcholu pro kužel (**bodVysky**). Dále je možné zvolit úhel rozevření daného tělesa a počet rovnoběžek.

### Metoda `spoctiParametry`

**Vstup:** `bodStredu`, `bodPolomeru`, `bodVysky`, `typ` (válec, kužel), `pocetRovnobezek`

**Výstup:** řídicí body s váhami v poli `Sarray`, řádkový a sloupcový uzlový vektor – `knot1`, `knot2`, řádkový a sloupcový stupeň – `degree1`, `degree2`

Metoda se skládá z následujících kroků:

1. Určení základních vektorů tělesa – jednotkové vektory podstavy `vektorX`, `vektorY` a vektor výšky `vektorZ`.

```
vektorX = (bodPolomeru - bodStredu)/velikostX;
vektorZ = bodVysky - bodStredu;
vektorY = (vektorX x vektorZ)/velikostY // vektorový součin
```

2. Určení počtu oblouků v závislosti na zadaném úhlu rozevření.

```
if (uhelRozevreni <= 90.0) pocetOblouku = 1;
else
    if (uhelRozevreni <= 180.0) pocetOblouku = 2;
    else
        if (uhelRozevreni <= 270.0) pocetOblouku = 3;
        else pocetOblouku = 4;
```

3. Určení velikosti jednoho oblouku.

```
uhel = uhelRozevreni/pocetOblouku;
```

4. Výpočet hodnoty proměnné `radek` a určení délky uzlového vektoru `knot1` – proměnná `knotlength1` a pomocné váhy `w1`.

```
radek = 2*pocetOblouku + 1;
knot_length1 = radek + degree1 + 1;
w1 = cos(uhel/2.0);
```

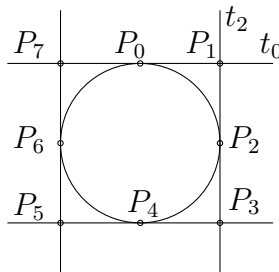
5. Nalezení řídicích bodů kružnicového oblouku podstavy.

Tato část se opakuje pro každé těleso, proto ji vysvětlíme pouze zde a u ostatních těles se na ni budeme odkazovat.

Nejdříve určíme krajní body úhlu o velikosti `uhel` z kroku 3. Známe střed kružnice (`bodStredu`) a její dva jednotkové navzájem kolmé vektory `vektorX`, `vektorY`. Počáteční bod úhlu je vstupní bod `bodPolomeru` a druhý bod vypočítáme z rovnice kružnice.

```
P2=BodStredu+velikostX*(cos(uhel)*vektorX+sin(uhel)*vektorY)
```

V těchto bodech určíme tečné vektory. Jejich průsečík určuje další řídicí bod, jak vidíme na případu kružnice (obr. 3.29). Váha nového bodu je součin váhy prvního řídicího bodu a kosinu poloviny velikosti úhlu tohoto oblouku (proměnná  $w_1$  z předchozího kroku).



Obrázek 3.29: Hledání řídicích bodů kružnice (360 stupňů)

V další části uděláme stejný výpočet postupně pro další oblouky. Poslední bod prvního oblouku je prvním bodem dalšího. Počet oblouků je určen ve druhém kroku. Řídicí body jsou ukládány do pole `Sarray`, proměnné `t0`, `t1` jsou tečny v krajních bodech oblouku. Pseudokód pro tento výpočet je:

```
index = 0;
p0[index] = bodPolomeru;
pomUhel = 0;
index = 0;
t0 = -sin(pomUhel)*vektorX + cos(pomUhel)*vektorY

for (i=1; i<=pocetOblouku; i++){ //kružn. oblouky
    pomUhel = pomUhel + uhel;
    // koncový bod a tečna oblouku o velikosti pomUhel
    p2 = bodStredu + velikostX*(cos(pomUhel)*vektorX +
        sin(pomUhel)*vektorY);
    t2 = -sin(pomUhel)*vektorX + cos(pomUhel)*vektorY;

    // hledám prostřední bod p1 na oblouku mezi p0, p2
    prusecikPrimek(p0, t0, p2, p1);
    // presun do pole ridich bodu
    Sarray[index]=p0;
    Sarray[index+1]=p1;
    Sarray[index+2]=p2;

    index = index + 2;
    if (i<pocetOblouku) {
        p0[index] = p2[index-2]; // posun na další oblouk
        t0 = t2;
    }
}
```

6. Generování řídicího vektoru v závislosti na počtu oblouků. Vnitřní hodnoty jsou generovány v závislosti na počtu oblouků podle předpisu daného pro kružnici.

```
switch(numberOfArcs){
    case 1: break;
    case 2: knot1=(0,0,0,0.5,0.5,1,1,1); break;
    case 3: knot1= (0,0,0,1/3,1/3,2/3,2/3,1,1,1) break;
    case 4: knot1= (0,0,0,0.25,0.25,0.5,0.5,0.75,0.75,1,1,1)
            break;
}
```

7. Odvození řídicích bodů horní podstavy metodou `horniPodstava`.
8. Podle počtu rovnoběžek zadaných uživatelem se dopočítají další řídicí body na svislých stěnách tělesa a provede se modifikace uzlového vektoru `knot2`. Stupeň `degree2` je roven jedné pro dvě rovnoběžky, pro více rovnoběžek se použije stupeň dva, abychom při změnách polohy řídicích bodů zachovali  $C^1$  spojitost.

```
if (pocetRovnobezek == 2){
    knot2 = (0,0,1,1);
}else
    pridejRovnobezky(pocetRovnobezek);
```

### Metody použité v metodě `spoctiParametry`

**Metoda `horniPodstava`** Pro válec provedeme posun o vektor `Z` řídicích bodů podstavy. Pro kužel se generuje pouze vrchol, což je vlastně zadaný bod `Vysky`. Abychom dostali výslednou síť bodů, je potřeba ho načíst tolikrát, kolik je bodů dolní podstavy.

**Metoda `pridejRovnobezky`** Metoda se volá, pokud uživatel zadá počet rovnoběžek větší než dvě. Omezení pro kužel i pro válec je interval  $\langle 2, 100 \rangle$ . Poté je nutné najít velikost posunutí bodů v závislosti na počtu rovnoběžek a modifikovat uzlový vektor, změnit počet sloupců a hodnotu délky uzlového vektoru. Tyto změny jsou řešeny následovně:

- Velikost posunutí.  
Dostaneme ji vydělením složek vektoru bočních stěn počtem rovnoběžek minus jedna. Body generujeme postupným přičítáním vypočítaného vektoru posunutí k předchozímu bodu.

```
posunuti = (Sarray[1] - Sarray[0])/(pocetRovnobezek-1);
for (j = 1; j<=pocetRovnobezek-1; j++)
    Sarray[j] = Sarray[j-1] + posunuti;
```



- Délka a modifikace uzlového vektoru.

NURBS křivka musí splňovat podmínku, že délka uzlového vektoru je rovna počtu bodů plus stupeň křivky plus jedna. Stupeň pro přímku je roven dvěma, tedy délka uzlového vektoru je rovna počtu rovnoběžek plus tři. Abychom zachovali plynulý průběh přímky, musí být uzlový vektor ekvidistantní, takže jako další část programu byl vytvořen generátor ekvidistantního uzlového vektoru `knot2` pro zadaný počet rovnoběžek.

```
knot_length2 = pocetRovnobezek + degree2 + 1;
knot2[0]=(0.0);
knot2[1]=(0.0);
knot2[2]=(0.0);
knot2[pocetRovnobezek]=(1.0);
knot2[pocetRovnobezek+1]=(1.0);
knot2[pocetRovnobezek+2]=(1.0);
for (k~= 3; k<=pocetRovnobezek-1; k++){
    knot2[k] = a/(pocetRovnobezek-2);
    a++;
}
```

**Metoda `prusecikPrimek`** Výpočet průsečíku přímek v prostoru je poměrně náročná operace. Obecně se jedná o výpočet tří rovnic o dvou neznámých. První možnost řešení je analytická. Přesný výpočet dosazovací metodou. Problémy vznikají, pokud je některý z koeficientů nulový. Druhou možností je použít některou numerickou metodu.

V programu bylo použito řešení rozšířením úlohy. Vzhledem k zadání leží přímky vždy v jedné rovině a jsou tedy vždy různoběžné. Jejich směrové vektory jsou `t0` a `t2`. Přidání vektoru `vektorZ` k jednomu z tečných vektorů vznikne rovina a úloha se změní na průsečík přímky s rovinou. Přístup je znázorněn na obr. 3.30.

Obecně:

Mějme danu přímku  $p : (P, \mathbf{v})$  a rovinu  $\alpha(Q, \mathbf{n})$

Průsečík přímky s rovinou se vypočítá jako:

$$X = P + \mathbf{v} \cdot \left( \frac{(\vec{PQ}, \mathbf{n})}{(\mathbf{v}, \mathbf{n})} \right) \quad (3.56)$$

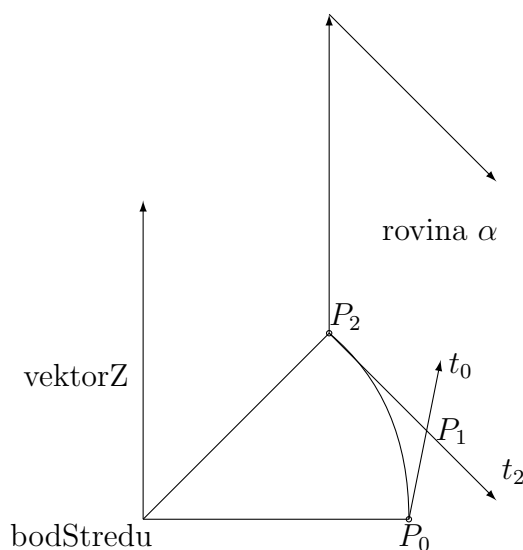
### Kritická místa

#### 1. Metoda `prusecikPrimek`

- Ve výsledném vzorci se jedná o podíl dvou skalárních součinů, tedy je to reálné číslo, kterým násobíme směrový vektor přímky.

#### 2. Vstupní podmínky

- Testy metody `param – Surftest1` a `Surftest2` musí být splněny.



Obrázek 3.30: Průsečík přímek v prostoru – proložení rovinou

- Vzhledem k požadavkům firmy se vykreslují kolmá tělesa, tedy vektory ze zadaných bodů jsou k sobě vzájemně kolmé.

$$(\text{bodStredu} - \text{bodPolomeru}, \text{bodStredu} - \text{bodVysky}) = 0$$

### 3. Omezení metod `pridejRovnobezky`, `rozevriTeleso`

- Počet rovnoběžek je v intervalu  $\langle 2, 100 \rangle$ .
- Při hledání ekvidistantního intervalu pro nové body je nutné psát číslo jedna jako 1.0, aby bylo typu `double`, jinak dojde k chybnému zaokrouhlování na celá čísla.
- Úhel rozevření je v intervalu  $\langle 1, 360 \rangle$ .

## Výsledné modely

Výsledné modely a ukázka jejich editace jsou vykresleny na obr. 3.31, 3.32, 3.33 a 3.34.

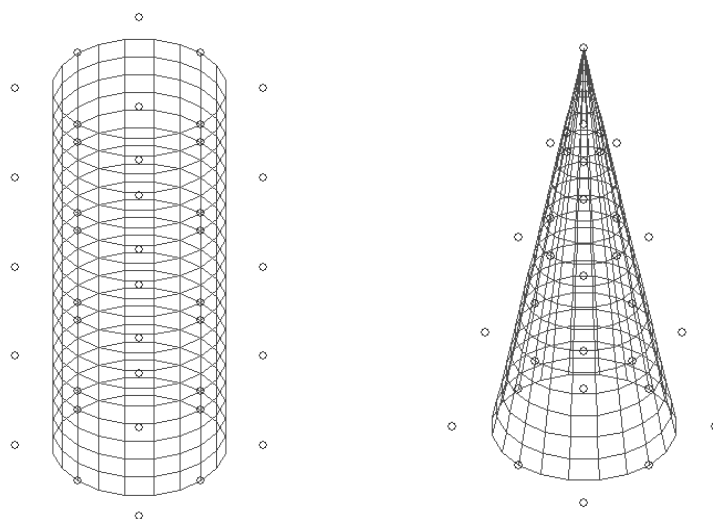
## Třída `SphereNurbs` – kulová plocha

**Programové zpracování** Uživatelským vstupem je bod zadávající střed koule (`bodStredu`), velikost poloměru (`polomer`) a úhel rozevření (`uhelRozevreni`).

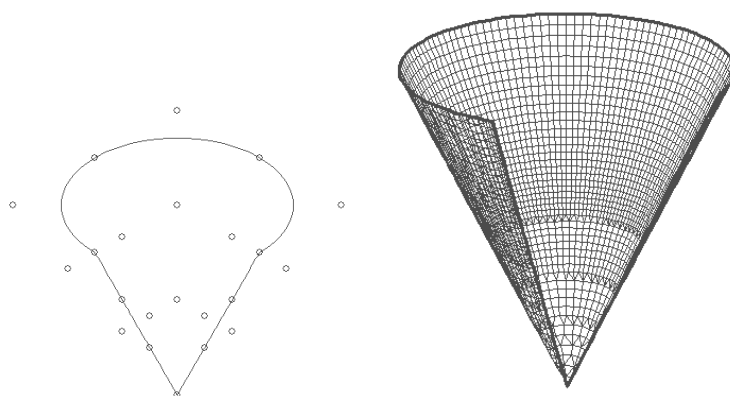
### Metoda `spoctiParamety`

**Vstup:** `bodStredu`, `polomer`, `uhelRozevreni`

**Výstup:** řídicí body s vahami v čtyřrozměrném poli `Sarray`, řádkový a sloupcový uzlový vektor `knot1`, `knot2`, stupně `degree1`, `degree2`



Obrázek 3.31: Válec a kužel s přidáním rovnoběžkami

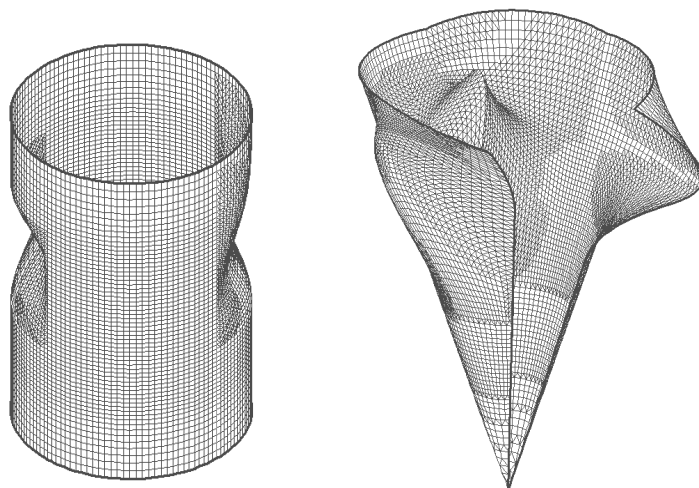


Obrázek 3.32: Kužel otevřený na 270 stupňů

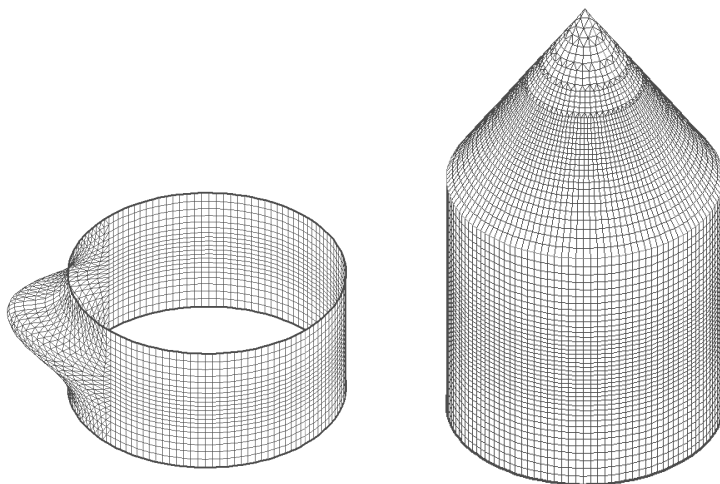
Prvním krokem je odvození základních řídicích bodů půlkružnice, která se bude otáčet, ze vstupních údajů. Poté se vypočítají další řídicí body a jejich váhy pro jednotlivé řádky v závislosti na velikosti úhlu rozevření.

1. Vygenerujeme základní body – metoda `generujZakladniBody`.
2. Určíme uzlový vektor `knot1` – metoda `generujKnot1`.  
`knot1 = (0.0,0.0,0.0,0.5,0.5,1.0,1.0,1.0)`
3. stanoví počet oblouků v závislosti na úhlu rozevření

```
if (uhelRozevreni<=90) pocetOblouku = 1;
else if (uhelRozevreni<=180) pocetOblouku = 2;
else if (uhelRozevreni<=270) pocetOblouku = 3;
else pocetOblouku = 4;
```

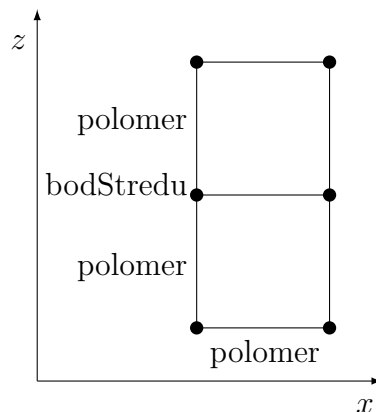


Obrázek 3.33: Modifikace NURBS kužele a válce I



Obrázek 3.34: Modifikace NURBS kužele a válce II

4. Určíme uzlový vektor `knot2` v závislosti na počtu oblouků analogicky podle bodu 6 (na str. 114) použitého u válce a kužele – metoda `generujKnot2(pocetOblouku)`.
5. Vypočítáme velikosti úhlu jednoho oblouku.  
`uhelJedenOblouk = uhelRozevreni/pocetOblouku;`
6. První a poslední body v jednotlivých sloupcích jsou shodné – metoda `generujBodyPolu(pocetOblouku);`.
7. Provedeme výpočet řídicích bodů pro jednotlivé řádky a určíme jejich váhy. Princip je podobný jako u metody `spoctiParametry` pro kužel a válec. Jednodušší je v určování základních vektorů. Základní body jsou umístěny v rovině  $xz$ , a proto dochází ke zjednodušení základních vektorů osového kříže – jsou to vektory



Obrázek 3.35: Generování základních bodů

$(1, 0, 0)$ ,  $(0, 1, 0)$ ,  $(0, 0, 1)$ . Metodu `spectiParametry` pro kulovou plochu můžeme zapísat pseudokódem následovně.

```
for (j = 1; j <= 3; j++){
    t0 = (0.0,1.0,0.0);
    index = 0;
    pomUhel = 0.0;
    for (i = 1; i <= pocetOblouku; i++){
        pomUhel = pomUhel + uhelJedenOblouk;
        Sarray[j][index+2].x = bodStredu.x + polomer * cos(pomUhel);
        Sarray[j][index+2].y = bodStredu.y + polomer * sin(pomUhel);
        Sarray[j][index+2].z = Sarray[j][index].z;
        Sarray[j][index+2].w = Sarray[j][index].w;

        t2[0]=(-1.0*sin(pomUhel), cos(pomUhel),0.0);
        prusecikPrimek(index, j, t0, cosUhel);
        index = index + 2; // prejdu na dalsi oblouk
        if (i < pocetOblouku)
            t0[q] = t2[q];
    }
}
```

### Kritická místa

#### 1. Vstupní podmínky

- Poloměr je kladné číslo.
- Úhel rozevření koule je v intervalu  $\langle 1, 360 \rangle$ .

#### 2. Metoda `generujBodyPolu`

První a poslední řádek tvoří stejné řídicí body – póly kulové plochy.

```

for ( i = 1; i < 2*pocet0blouku + 1; i++){
    Sarray[0][i].x = Sarray[0][0].x;
    Sarray[0][i].y = Sarray[0][0].y;
    Sarray[0][i].z = Sarray[0][0].z;
}

```

Jejich váhový vektor musí být pro celou kulovou plochu tvaru:

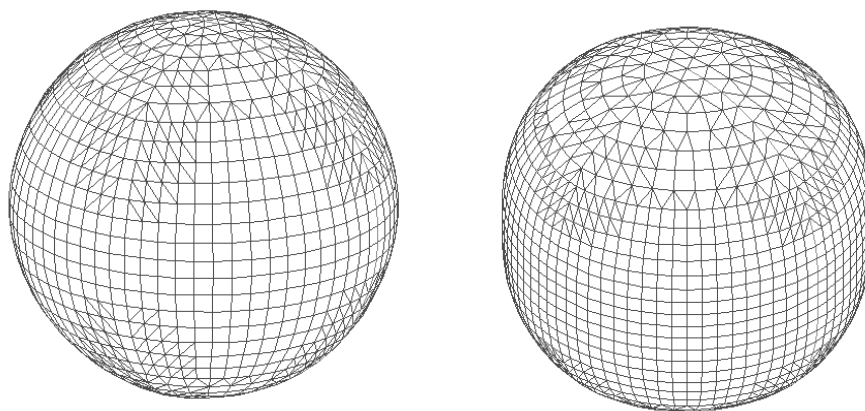
$$(1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1).$$

Pro rozevřenou kulovou plochu se použije pouze odpovídající část vektoru.

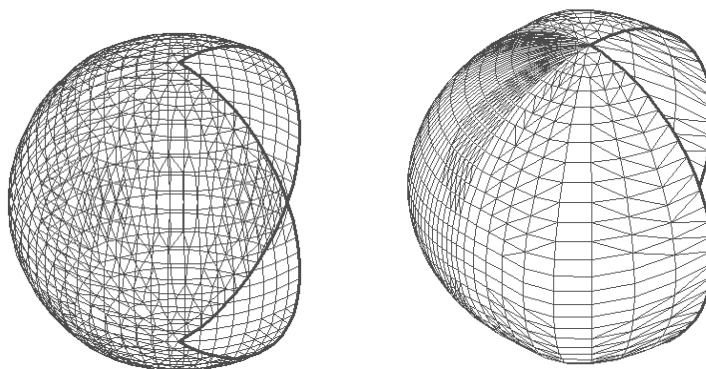
3. Metoda `prusecikPrimek`  
Stejná jako u třídy `CylinderNurbs`.
4. Převod úhlů na radiány  
Při používání funkcí kosinus a sinus je nutné převádět úhly ze stupňů na radiány.

### Výsledné modely

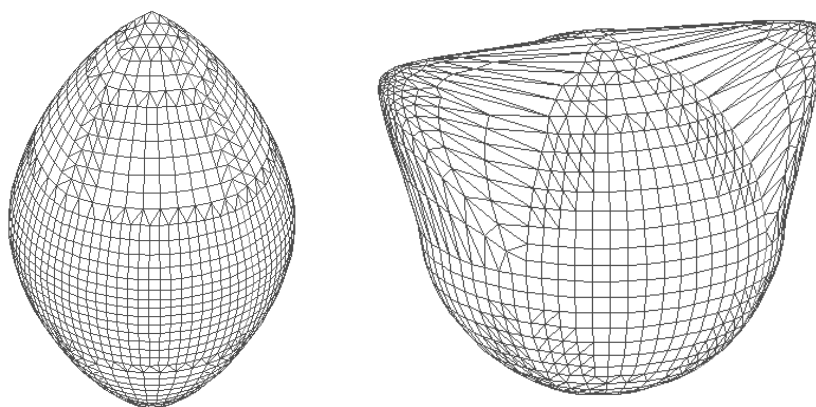
Ukázkové modely jsou vykresleny na obr. 3.36, 3.37, 3.38 a 3.39



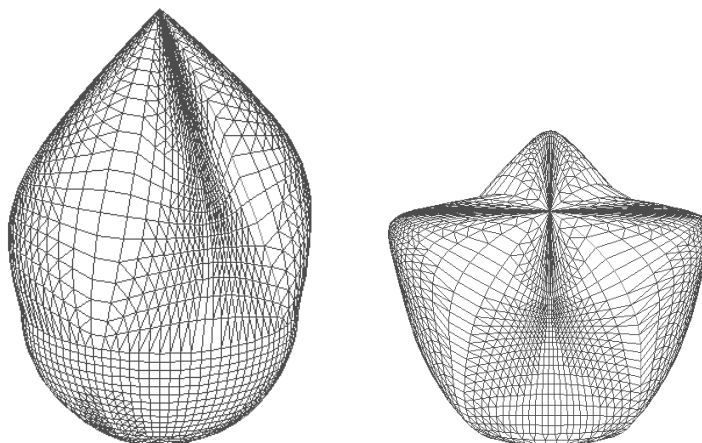
Obrázek 3.36: NURBS koule stupně dva a tři



Obrázek 3.37: Změna uzlového vektoru NURBS koule



Obrázek 3.38: Změna polohy řídicích bodů a váhy NURBS koule



Obrázek 3.39: Možnosti tvorby ze základního tvaru

### Třída PlaneNurbs – rovina

**Programové zpracování** Vstupem jsou tři obecné body – `bod1`, `bod2`, `bod3` – které určují obecnou rovinu. Dále je možno zadat počet rovnoběžek a poledníků, které vygenerují hustší síť řídicích bodů pro snadnější úpravy.

#### Metoda `spoctiParametry`

**Vstup:** `bod1`, `bod2`, `bod3`, `pocetPoledniku`, `pocetRovnobezek`, parametr  $(u, v)$

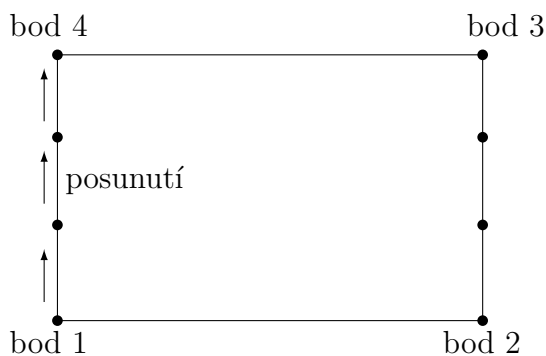
**Výstup:** řídicí body s vahami v čtyřrozměrném poli `Sarray`, řádkový a sloupcový uzlový vektor `knot1`, `knot2`

1. Vygenerujeme základní body metodou `generujZakladniBody`;
2. V závislosti na počtu poledníků se určí uzlový vektor `knot1` a spočítají další řídicí body metodou `pridejPoledniky`.
3. V závislosti na počtu rovnoběžek určíme uzlový vektor `knot2` a vygenerujeme další řídicí body metodou `pridejRovnobezky`.

Uzlové vektory se v obou případech vypočítají stejně. Prvních  $\text{stupeň}+1$  členů je rovno nule a posledních  $\text{stupeň}+1$  členů je rovno jedné. Délka uzlového vektoru se odvodí z počtu rovnoběžek či poledníků, aby bylo splněno pravidlo pro B-spline funkce. Prostřední uzly jsou rozděleny ekvidistantně.

#### Metody použité v metodě `spoctiParametry`

**Metoda `pridejRovnobezky`** Pro jednotlivé sloupcové body vygeneruje body podle proměnné `pocetRovnobezek`, kterou zadá uživatel. (viz obr. 3.40).



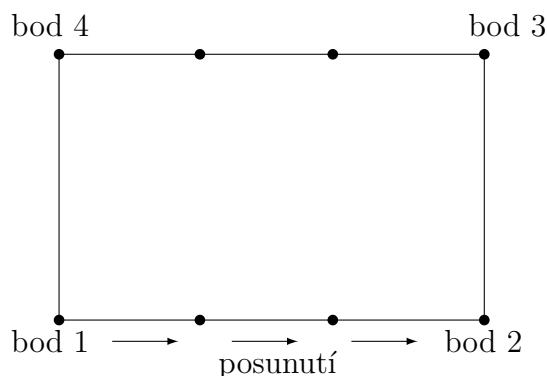
Obrázek 3.40: Generování bodů pro obecný počet rovnoběžek (4)

```
posunuti= (bod3 - bod2)/(pocetRovnobezek - 1);
for ( i = 0; i < pocetPoledniku; i++)
    for ( j = 1; j < pocetRovnobezek; j++)
        Sarray[i][j] = Sarray[i][j-1] + posunuti;
```



**Metoda pridejPoledniky**

Pro spodní a horní řádek vygeneruje řídicí body, viz obr.3.41.



Obrázek 3.41: Generování bodů pro obecný počet poledníků (4)

```
posunuti= (bod2 - bod1)/(pocetPoledniku - 1);
for ( j = 0; j < 2; j++)
  for (i = 1; i <= radek-1; i++)
    Sarray[i][j] = Sarray[i-1][j] + posunuti;
```

**Kritická místa**

## 1. Vstupní podmínky

- Vstupní body – bod1, bod2, bod3 jsou vzájemně různé.
- Vstupní body neleží na jedné přímce. K výpočtu využijeme vlastnost – pokud promítnutý bod leží na alespoň dvou průmětech přímky, leží na této přímce i v prostoru. K otestování, zda bod leží na přímce v rovině použijeme vztah:

$$(b_0 - a_0)(p_1 - a_1) - (b_1 - a_1)(p_0 - a_0) = 0, \quad (3.57)$$

kde body  $A = (a_0, a_1)$  a  $B = (b_0, b_1)$  určují přímku a testujeme bod  $P = (p_0, p_1)$ .

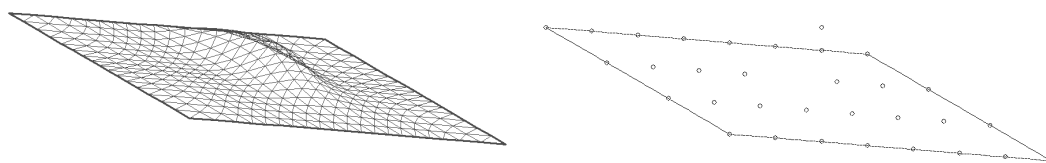
- Počet poledníků je v intervalu  $\langle 2, 100 \rangle$ .
- Počet rovnoběžek je v intervalu  $\langle 2, 100 \rangle$ .

## 2. Určení stupně degree1, degree2

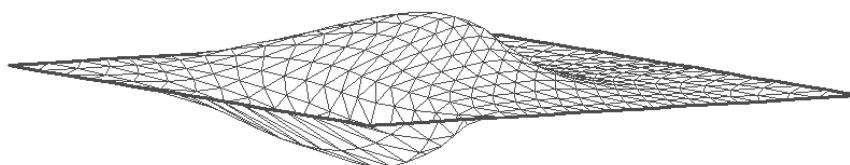
Pokud je počet rovnoběžek nebo poledníků roven dvěma, jedinou možností je volba stupně jedna. Počet bodů musí být vždy alespoň o jeden větší než je stupeň v daném směru. Pro ostatní přípustné hodnoty se volí stupeň roven dvěma, abychom zajistili spojitost při změně polohy řídicích bodů.

**Výsledné modely**

Na obr. 3.42, 3.43 je ukázka výsledných modelů NURBS rovin.



Obrázek 3.42: NURBS rovina upravená změnou bodu



Obrázek 3.43: NURBS rovina upravená změnou bodu

### Třída CubeNurbs

**Programové zpracování** Jako vstup od uživatele se přijímají tři body – dva tvoří úhlopříčku podstavy a třetí určuje výšku. Jeho poloha vzhledem k druhému bodu určuje, zda se bude jednat o kolmý nebo kosý hranol. Dále je to počet rovnoběžek a poledníků, které vygenerují hustší síť bodů k lepší modifikaci objektu. Pokud je počet rovnoběžek i poledníků roven dvěma, pak je stupeň plochy jedna. Při vyšším počtu poledníků a rovnoběžek se volí stupeň druhý, aby nevznikaly zlomy při posunu řídicích bodů.

#### Metoda spocitiParamety

**Vstup:** bodUhlopricky1, bodUhlopricky2, bodVysky, pocetRovnobezek, pocetPoledniku

**Výstup:** řídicí body s vahami v čtyřrozměrném poli `Sarray`, řádkový a sloupcový uzlový vektor `knot1`, `knot2`

Metoda je složena z následujících čtyř částí:

1. Generování čtyř základních bodů podstavy.  
`generujZakladniBody();`
2. Přidání bodů podle počtu poledníků, pokud je počet větší než dva.  
`if (radek > 2) pridejPoledniky(radek);`
3. Výpočet řídicích bodů na jednotlivých rovnoběžkách.  
`pridejRovnobezky(sloupec, radek, bodVysky);`
4. Nalezení uzlového vektoru `knot1` pro jednotlivé rovnoběžky, pokud je stupeň roven jedné (počet rovnoběžek i poledníků je roven dvěma), má uzlový vektor uvedený tvar.

```

if ( degree1 != 1)
    generujKnot1(degree1, radek);
else
    knot1 = (0.0, 0.0,0.25,0.5,0.75,1.0,1.0);

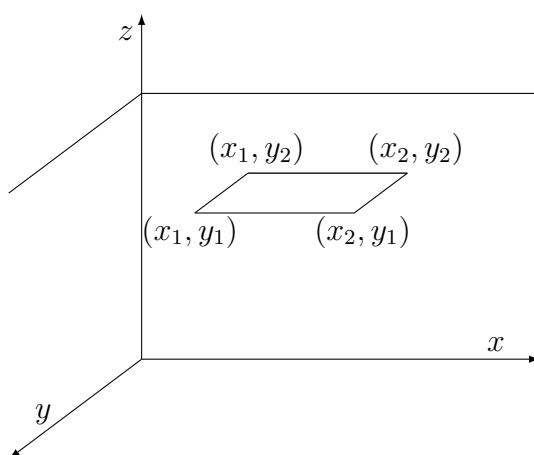
if ( degree2 != 1)
    generujKnot2(degree2, sloupec);
else
    knot2 = (0.0, 0.0,1.0,1.0);

```

### Metody použité v metodě spoctiParametry

**generujZakladniBody** Uživatel zadá dva body úhlopříčky a metoda určí dva zbývající. Pro jednoznačnost je nutné předpokládat, že daná úhlopříčka leží v rovině  $(xy)$  nebo v rovině s ní rovnoběžné.

Z obr. 3.44 je zřejmý výpočet zbývajících dvou vrcholů dolní podstavy. Zachová se vždy jedna ze souřadnic známých vrcholů. Velikost váhy je u všech bodů rovna jedné.



Obrázek 3.44: Výpočet řídicích bodů dolní podstavy

**metoda pridejPoledniky(radek);**

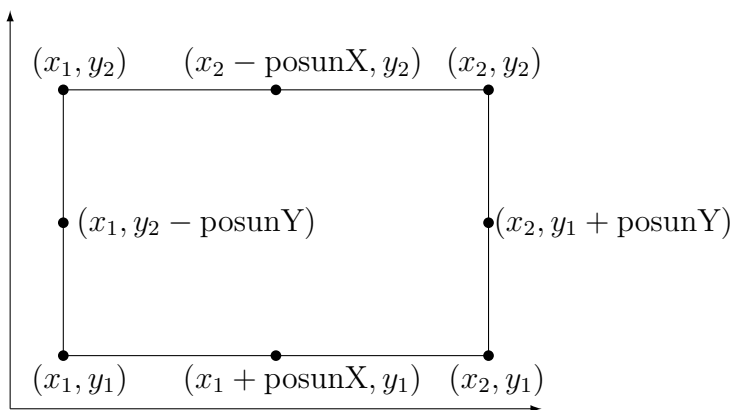
1. Spočítat velikost posunutí jednotlivých bodů vzhledem k počtu poledníků, které má hranol mít. Je nutné si uvědomit, že počet poledníků – proměnná **radek** – zde znamená počet poledníků na jedné hraně podstavy. Počet řídicích bodů podstavy je tedy jiný.

```

posunX = (Sarray[1][0].x - Sarray[0][0].x) / (radek-1);
posunY = (Sarray[2][0].y - Sarray[0][0].y) / (radek-1);

```

2. Pro jednotlivé hrany spodní podstavy spočítat řídicí body jako posunutí vhodné souřadnice předchozího bodu a velikost **posunX** nebo **posunY**. Váhy jsou u všech bodů rovny jedné (viz obr. 3.45). Je nutné si uvědomit, že pro dvě stěny má posunutí opačné znaménko a také, že poslední řídicí bod je roven prvnímu řídicímu bodu.

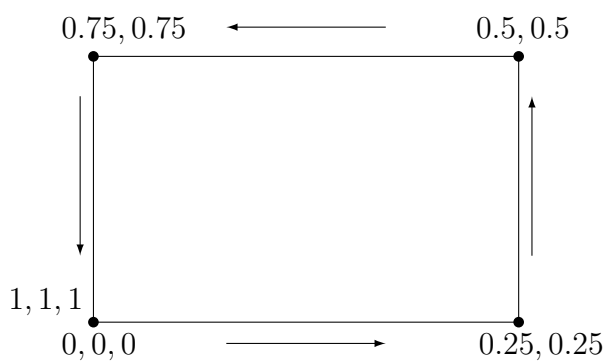


Obrázek 3.45: Výpočet řídicích bodů pro tři poledníky

**metoda** generujKnot1(degree1, radek)

Metoda vrací tvar uzlového vektoru `knot1`. Pro stupeň druhý a počet rovnoběžek větší než dva, musí uzlový vektor obsahovat vždy tyto členy, které způsobují to, že rohové řídicí body leží na ploše (viz obr. 3.46).

`knot1 = (0.0, 0.0, 0.0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1.0, 1.0, 1.0)`



Obrázek 3.46: Základní uzlový vektor pro hranol

Při volbě  $k$  poledníků dochází k přidání  $4k$  řídicích bodů (na každou hranu podstavy  $k$ ), a proto musí být uzlový vektor delší o  $4k$  hodnot. Ty se musí rovnoměrně rozmístit mezi výše uvedený vektor. Počet vložených čísel mezi jednotlivé hodnoty uzlového vektoru určuje proměnná `pocetVnitřníchUzlu`.

`pocetVnitřníchUzlu = ( 4 * radek - 5 * degree1 - 2) / 4;`

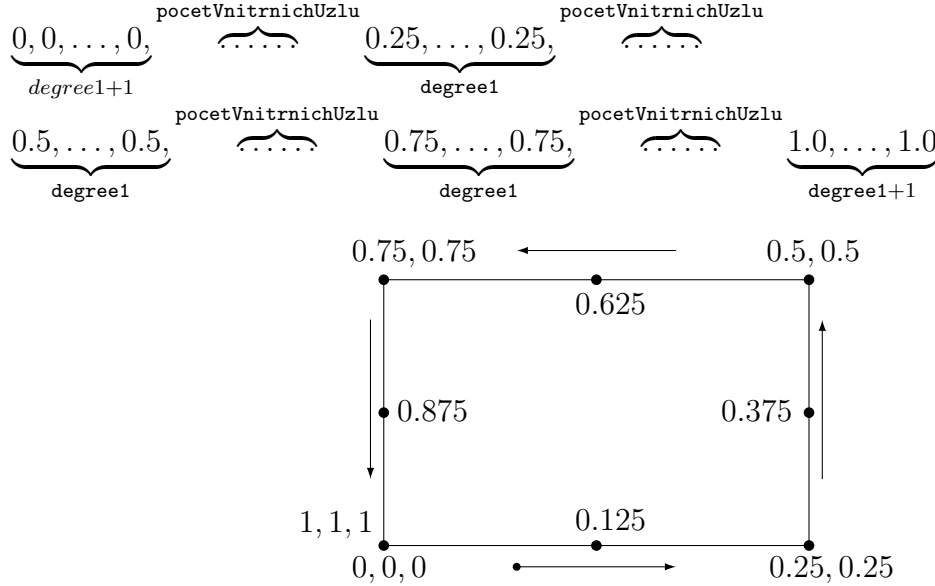
Pomocí `for` cyklů je potřeba naplnit uzlový vektor délky  $(5 * \text{degree1} + 1 + 4 * \text{pocetVnitřníchUzlu} + 1)$  známými hodnotami pro rohové řídicí body a ekvidistantními hodnotami pro přidání rovnoběžek. Ekvidistantní hodnoty se vypočítají podle:

$\text{knot1}[i] = \text{knot1}[i-1] + 0.25 / (\text{pocetVnitrnichUzlu} + 1);$

Na obr. 3.47 je případ pro tři poledníky. Uzlový vektor  $\text{knot1}$  má následující tvar:

$(0.0, 0.0, 0.0, 0.125, 0.25, 0.25, 0.375, 0.5, 0.5, 0.625, 0.75, 0.75, 0.875, 1.0, 1.0, 1.0)$

Obecně dostáváme:



Obrázek 3.47: Generování uzlového vektoru  $\text{knot1}$  pro tři poledníky

**metoda** `generujKnot2(degree2, sloupec)`

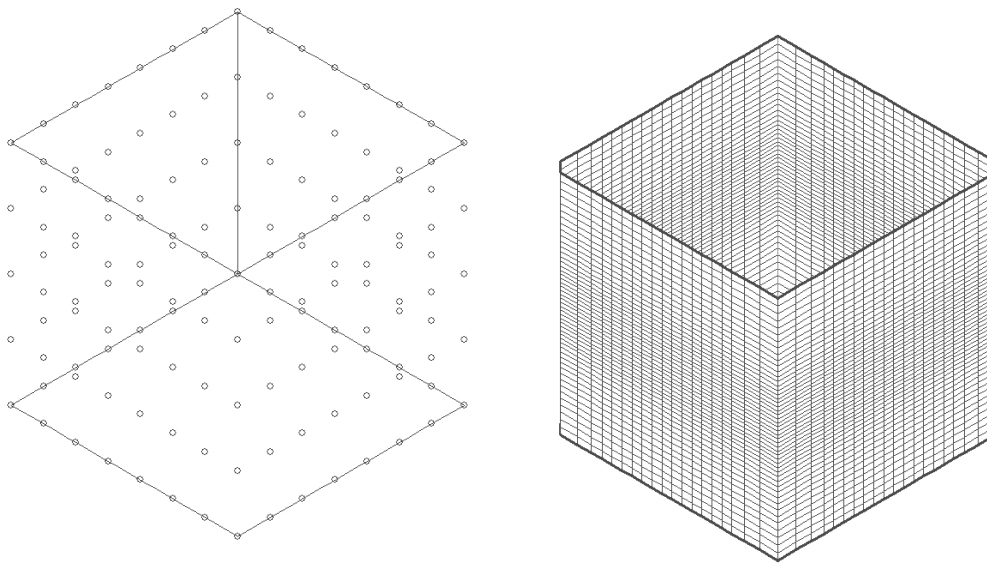
Uzlový vektor  $\text{knot2}$  musí obsahovat prvních a posledních  $\text{degree2}+1$  členů stejných, opět kvůli tomu, aby hrana začínala a končila v rohovém bodě. Vnitřní uzly jsou ekvidistantní. Uzlový vektor  $\text{knot2}$  se používá pro svislé hrany, není tedy nutné jako v případě  $\text{knot1}$  procházet všechny 4 vrcholy postupně, ale pohybujeme se vždy mezi dvěma body.

$$\underbrace{0, 0, \dots, 0}_{\text{degree2}+1}, \underbrace{\dots}_{\text{sloupec}-\text{degree2}}, \underbrace{1.0, \dots, 1.0}_{\text{degree2}+1}$$

### Kritická místa

#### 1. Vstupní podmínky

- Vstupní body – `bodUhlopricky1`, `bodUhlopricky2`, `bodVysky` jsou vzájemně různé.
- Body na úhlopříčce leží v rovině rovnoběžné s rovinou  $xy$ , tj. mají stejnou  $z$ -ovou souřadnici.
- Vstupní bod `bodVysky` leží mimo rovinu obsahující úhlopříčku, která je rovnoběžná s  $xy$ , tj. má jinou  $z$ -ovou souřadnici.



Obrázek 3.48: NURBS hranol s pěti rovnoběžkami a osmi poledníky

- Počet poledníků je v intervalu  $\langle 2, 100 \rangle$ .
  - Počet rovnoběžek je v intervalu  $\langle 2, 100 \rangle$ .
2. Určení stupně `degree1`, `degree2`  
Provede se opět v konstruktoru analogicky jako u roviny (viz Kritická místa u tvorby NURBS roviny)
  3. Určení proměnné `radek`  
V případě hranolu neodpovídá proměnná `pocetPoledniku` proměnné `radek`. Toto číslo určuje počet poledníků na jedné stěně, ale proměnná `radek` udává počet bodů v jednom sloupci. Po provedení metody `spoctiParametryHranol` musíme v konstruktoru změnit proměnnou `radek`.

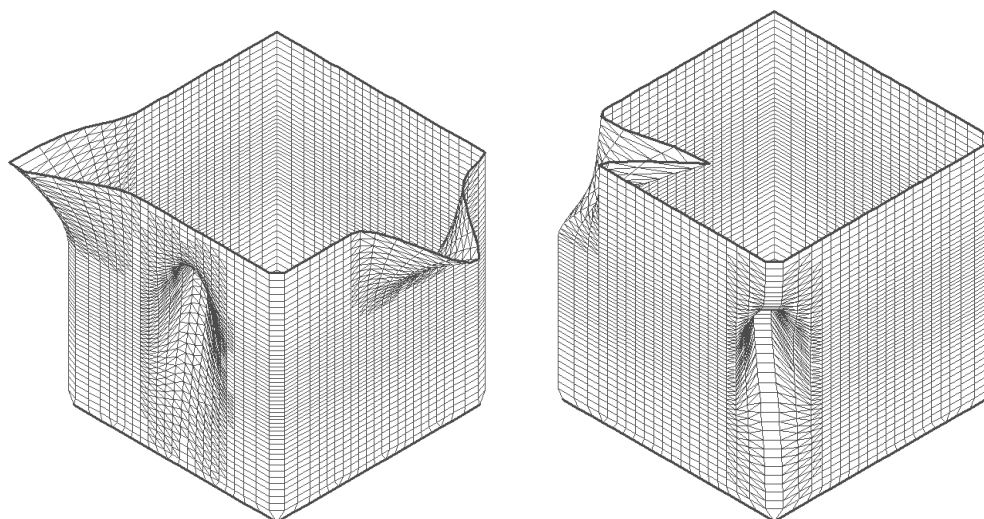
```
radek = 4*(pocetPoledniku - 1) + 1;
```

## Výsledné modely

Výsledné modely jsou na obr. 3.48, 3.49.

## Třída `AnuloidNurbs`

**Programové zpracování** Anuloid se zadává pomocí dvou bodů zadávající osu rotace anuloidu, dále středem a poloměrem kružnice rotace. Velikost úhlu rozevření anuloidu od 1 do 360 stupňů. Výhodou této konstrukce je to, že osa rotace může být libovolná přímka v prostoru. Uživatel nemusí používat další funkce k přesunutí anuloidu do požadované polohy.



Obrázek 3.49: Možnosti úpravy hranolu

**Metoda spocitiParametry**

**Vstup:** bod0sy1, bod0sy2, bodStredu, polomer, uhelRozevreni

**Výstup:** řídicí body s vahami v čtyřrozměrném poli Sarray, řádkový a sloupcový uzlový vektor knot1, knot2, stupně degree1, degree2

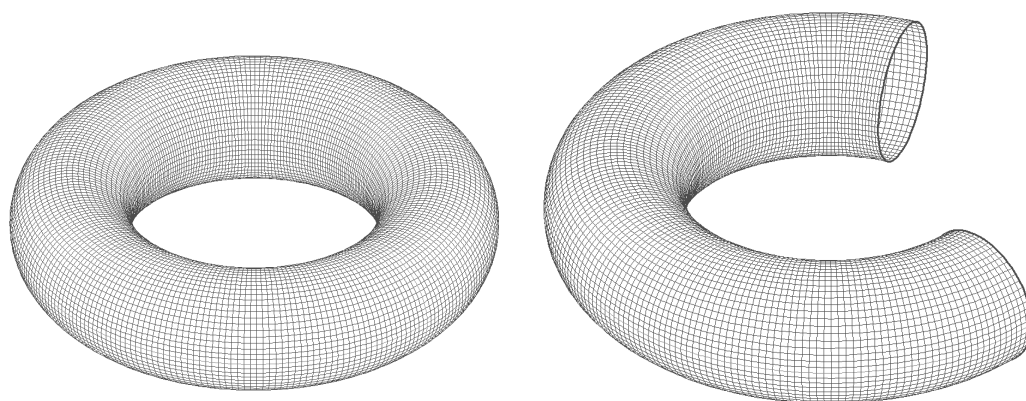
Nejdříve je nutné vygenerovat první kružnici se středem bodStredu a daným poloměrem (metoda generujZakladniBody). Poté se určí analogicky jako u kružnice uzlové vektory knot1 a knot2. V závislosti na úhlu rozevření se generují další body a jejich váhy. Stupeň pro sloupce i pro řádky je roven dvěma.

Metoda spocitiParametry provádí následující kroky:

1. Vygeneruje základní body – metoda generujZakladniBody.
2. Určuje uzlový vektor knot1 – metoda generujKnot1.  
 $\text{knot1} = (0.0, 0.0, 0.0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1.0, 1.0, 1.0)$
3. Stanoví počet oblouků v závislosti na úhlu rozevření.

```
if (uhelRozevreni<=90) pocetOblouku = 1;
else if (uhelRozevreni<=180) pocetOblouku = 2;
else if (uhelRozevreni<=270) pocetOblouku = 3;
else pocetOblouku = 4;
```

4. Určuje uzlový vektor knot2 v závislosti na počtu oblouků analogicky podle bodu 6 (str. 114) použitého u válce a kužele – metoda generujKnot2.
5. Výpočet velikosti úhlu jednoho oblouku.  
 $\text{uhelJedenOblouk} = \text{uhelRozevreni} / \text{pocetOblouku};$



Obrázek 3.50: NURBS anuloid (rozevření)

6. Výpočet řídicích bodů pro jednotlivé řádky a určení jejich vah. Princip je stejný jako u metody `spectiParametry` pro kužel a válec (viz postup a obr. 3.29 na str. 113).

### Kritická místa

1. Vstupní podmínky.
  - Poloměr je kladné číslo.
  - Úhel rozevření anuloidu je v intervalu  $\langle 1, 360 \rangle$ .
  - Bod středu kružnice rotace nesmí ležet na ose anuloidu. Pak by se jednalo o dvojnou kouli. Kulovou plochu ale vykreslujeme pomocí samostatné třídy `SphereNurbs`. Vzájemnou polohu přímky a bodu v prostoru řešíme konečným automatem.
  - Osa je zadána dvěma různými body.
2. Převod úhlů na radiány.  
Při používání funkcí kosinus a sinus je nutné převádět úhly ze stupňů na radiány.

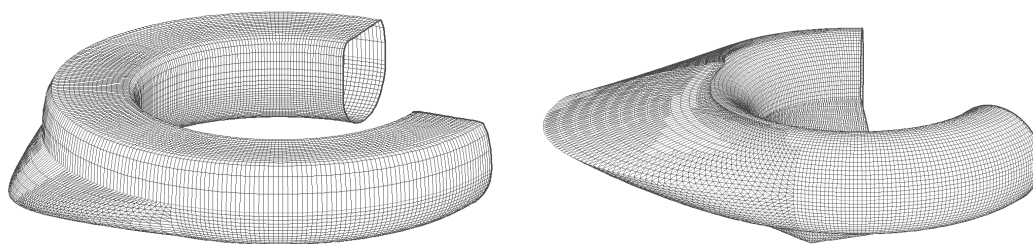
### Výsledné modely

Ukázku různých anuloidů a jejich modifikací naleznete na obr. 3.50, 3.51.

### Rotační tělesa

**Programové zpracování** Rotační těleso jako NURBS je zadáno obecnou osou rotace a libovolnou NURBS křivkou. Lze také zadat úhel rozevření tělesa od jednoho do 360 stupňů. Výhodou rotačních těles je možná editovatelnost výsledného modelu. Každému rotačnímu tělesu je totiž přiřazena síť řídicích bodů a uživatel má možnost měnit jejich polohu, váhu. Všechny tyto změny jsou pouze lokální.





Obrázek 3.51: NURBS anuloid – možnosti úpravy

**Metoda spoctiParametry**

**Vstup:** NURBS křivka (řídící body, váhový a uzlový vektor, stupeň), úhel rozevření

**Výstup:** řídící body v poli `Sarray` s vahami v poli `weights`, řádkový a sloupcový uzlový vektor – `knot1`, `knot2`, řádkový a sloupcový stupeň – `degree1`, `degree2`

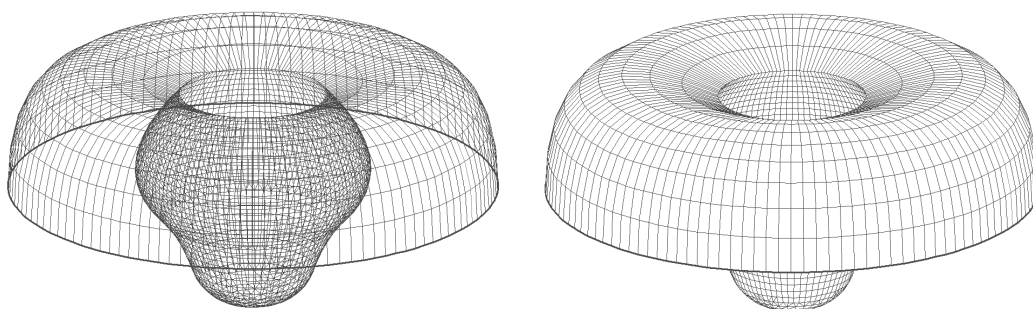
1. Inicializace – uživatelem zadané body jsou prvními body pro hledané kružnice.
2. Podle velikosti úhlu rozevření rozdělíme na oblouky a najdeme velikost základního úhlu (`angle`).

```
angle ∈ (0, 90 > → numberOfArcs = 1;
angle ∈ (90, 180 > → numberOfArcs = 2;
angle ∈ (180, 270 > → numberOfArcs = 3;
angle ∈ (270, 360 > → numberOfArcs = 4;
angle = angle/numberOfArcs;
```

3. Určíme uzlový vektor pro kružnice otáčení jednotlivých bodů vstupní křivky v závislosti na velikosti úhlu rozevření. Tento tvar vychází z vyjádření kružnicových oblouků jako NURBS.

```
switch(numberOfArcs){
    case 1: break;
    case 2: knot1=(0,0,0,0.5,0.5,1,1,1); break;
    case 3: knot1= (0,0,0,1/3,1/3,2/3,2/3,1,1,1) break;
    case 4: knot1= (0,0,0,0.25,0.25,0.5,0.5,0.75,0.75,1,1,1)
            break;
}
```

4. V dalším kroku vypočítáme souřadnice bodů, které vznikají otočením řídících bodů vstupní přímky o úhel `angle`. V obou těchto bodech nalezneme tečny a průsečík těchto tečen je další řídící bod. Tento postup opakujeme pro všechny oblouky příslušného kružnicového oblouku a je analogický pro tvorbu základních těles uvedenou v předchozích částech. Rozdílem je, že osa rotace může být v obecné poloze vzhledem ke vstupní křivce.



Obrázek 3.52: Rotační tělesa jako NURBS (s viditelností)

## Kritická místa

Nejdříve je nutné zkontrolovat vstupní hodnoty zadané uživatelem (viz část a). Ve výpočetní části se vyskytne problém, pokud některý z bodů křivky leží na ose rotace. V tomto případě degeneruje kružnice v jeden bod, ale ten je nutný zahrnout do výsledné sítě řídicích bodů devětkrát (viz definice NURBS kružnice). V metodách použitých pro generování řídicích bodů můžeme také narazit na problémy, které v další části podrobně probereme (viz část b, c).

### a. Kontrola vstupních hodnot

- osa rotace je zadána dvěma různými body,
- úhel rozevření tělesa je v intervalu  $\langle 1, 360 \rangle$ ,
- NURBS křivka je korektně zadána, tj. různé řídicí body každý s kladnou vahou, uzlový vektor je neklesající posloupnost kladných reálných čísel a splňuje podmínku vzhledem ke stupni a počtu řídicích bodů.

### b. metoda `PromitniBodNaPrimku(A,B,C)`

Metoda se používá při hledání středu kružnice, kolem kterého budeme bod otáčet. Provedeme projekci obecného bodu  $A$  na přímkou danou body  $B, C$ . Jelikož promítáme kolmo na danou přímku, je směrový vektor daný body  $B, C$  normálovým vektorem pro promítací rovinu jdoucí bodem  $A$ .

Rovnice přímky:  $X = B + (B - C)t$

Rovnice roviny:  $(B - C)X - (B - C)A = 0$

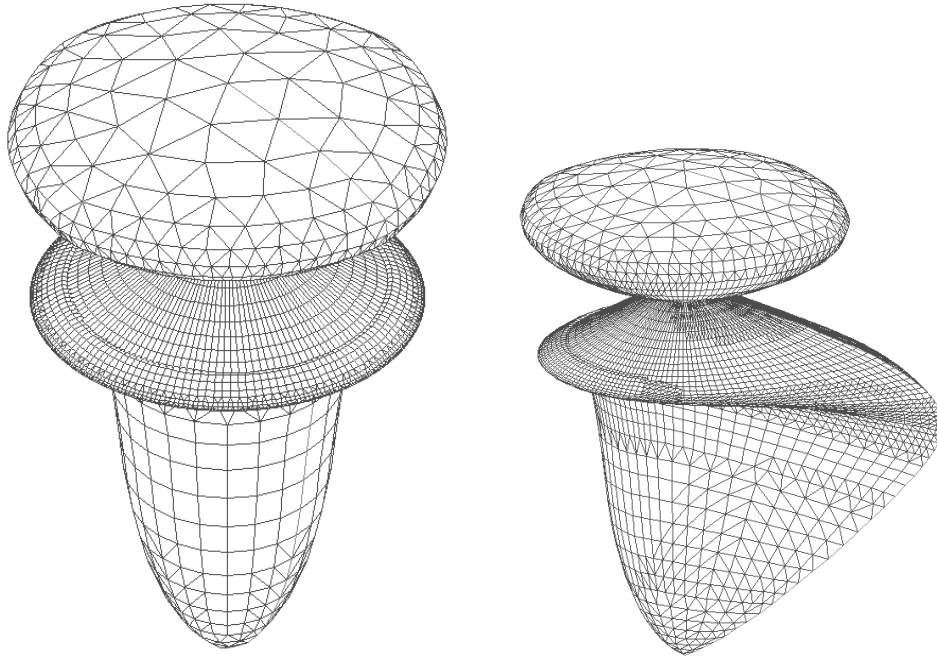
Po dosazení rovnice přímky do rovnice roviny dostáváme vztah pro výpočet parametru  $t$ . Dosazením parametru do rovnice přímky získáme výsledný průmět bodu  $A$ .

### c. metoda `PrusecikPrimek()`

Metoda, která je podrobně probrána u kužele a válce.

## Výsledné modely

Obr. 3.52 a 3.53 ukazují příklad obecných rotačních NURBS ploch a způsoby jejich úpravy.



Obrázek 3.53: Rotační tělesa – možnosti editace

### 3.5 Spojité navazování NURBS ploch

V technické praxi se často objevuje potřeba hladkého navazování NURBS ploch. K jedné, již vytvořené ploše, chceme přidat další tak, aby přechod mezi nimi byl  $C^1$  spojitý. Ke stanovení podmínek je nutné udělat podrobný rozbor celé situace. V literatuře je uvedena pouze základní podmínka o vztahu sousedních bodů, není zde vysvětlen vliv váhy či tvar uzlového vektoru. V této části bude podrobně probráno celé odvození.

Mějme danu NURBS plochu  $S(u, v)$  s řídicími body  $P_{ij}$  s vahami  $w_{ij}$ ,  $i = 0, 1, \dots, m$  a  $j = 0, 1, \dots, n$  a řádkovým uzlovým vektorem  $\mathbf{u}$  a sloupcovým vektorem  $\mathbf{v}$ . Dále je dán řádkový a sloupcový stupeň plochy  $p, q$ .

Druhou plochu chceme hladce navázat podél posledního sloupce první plochy. Označme ji  $S'(u, v)$ . Její řídicí body označme  $Q_{ij}$  s vahami  $w'_{ij}$ ,  $i = 0, 1, \dots, m$  a  $j = 0, 1, \dots, r$ . První zřejmou podmínkou je, že počet řádků musí být u obou ploch stejný. Také sloupcový uzlový vektor a sloupcový stupeň obou ploch se musí rovnat, neboť musí mít společnou křivku danou posledním sloupcem původní plochy. Tedy sloupcový uzlový vektor je  $\mathbf{v}$  a sloupcový stupeň je  $q$ . Zbývající řádkový vektor označíme  $\mathbf{u}'$  a řádkový stupeň  $s$ .

Kromě předchozích formálních podmínek musí platit základní vztah. Odpovídající body posledního a předposledního sloupce původní plochy leží na přímce spolu s body druhého sloupce připojené plochy. Váhy odpovídajících bodů těchto tří sloupců jsou stejné

$$w_{i,n-1} = w_{i,n} = w'_{i,0} = w'_{i,1}.$$

Body a váhy prvního sloupce navazující plochy jsou stejné jako řídicí body a váhy posledního sloupce původní plochy – společná hraniční křivka. Symbolicky zapsáno:

$$Q_{i,0} = P_{i,n}, w_{i,n} = w'_{i,0} \text{ pro } i = 0, 1, \dots, m. \quad (3.58)$$

$$P_{i,n-1}Q_{i,1} = k.P_{i,n-1}Q_{i,0} = k.P_{i,n-1}P_{i,n} \text{ pro libovolné } k \neq 0, k \in \mathbb{R}. \quad (3.59)$$

Rozepsáním dostáváme vyjádření bodů druhé sloupce připojené plochy:

$$Q_{i,1} = P_{i,n-1}(1 - k) + kQ_{i,0} \quad (3.60)$$

Z matematické analýzy platí věta o spojitým navázání. Nutnou a dostatečnou podmínkou je přítomnost tečné roviny, tedy existence parciálních derivací. V tomto případě je jedna derivace zřejmě shodná. Obě plochy mají společnou hraniční křivku danou stejnými řídicími body s váhami, stejným uzlovým vektorem i stupněm. Poněkud obtížnější je situace pro druhý směr – řádky.

Musíme dokázat, že při splnění výše uvedených podmínek v obecném bodě hraniční křivky existuje společná tečná rovina pro obě plochy. Budeme potřebovat derivace v prvním a posledním bodě NURBS křivky. Tu lze vyjádřit jako (uzlový vektor je na intervalu  $\langle 0, 1 \rangle$ ):

$$C'(0) = \frac{m}{1 - u_{p-1}}(Q_{i,1} - Q_{i,0})\frac{w_{i,1}}{w_{i,0}} = \frac{m}{1 - u_{p-1}}(Q_{i,1} - P_{i,n})\frac{w_{i,1}}{w_{i,0}} \quad (3.61)$$

$$C'(1) = \frac{m}{m - u_{m-p-1}}(P_{i,n} - P_{i,n-1})\frac{w_{i,n-1}}{w_{i,n}} \quad (3.62)$$

Z uvedených vzorců vidíme, že derivace v krajních bodech závisí pouze na vektoru určeném posledními dvěma nebo prvními dvěma body křivky. Dokážeme-li tedy, že tyto derivace jsou lineárně závislé, pak derivace v krajních bodech je stejná až na násobek a navázání obou ploch je spojitě.

Vezmeme tedy pro stejný parametr  $v$  obecný bod na poslední a předposlední křivce první plochy (sloupcové) a na první křivce (sloupcové) navazující plochy. Předpokládáme, že platí všechny uvedené podmínky výše. Vyjádříme obecný bod na těchto křivkách:

$$C_{n-1}^S(v) = \sum_{i=0}^m w_{i,n-1}P_{i,n-1}N_i^q(v) \quad (3.63)$$

$$C_n^S(v) = \sum_{i=0}^m w_{i,n}P_{i,n}N_i^q(v) = \sum_{i=0}^m w_{i,n}Q_{i,0}N_i^q(v) \quad (3.64)$$

$$C_1^{S'}(v) = \sum_{i=0}^m w'_{i,1}Q_{i,1}N_i^q(v) \quad (3.65)$$

Použitím předpokladů o rovnosti vah a z rovnice (3.60) dostáváme obecný bod na křivce z bodů prvního sloupce připojené plochy.

$$C_1^{S'}(v) = \sum_{i=0}^m w_{i,n}(P_{i,n-1}(1 - k) + kQ_{i,0})N_i^q(v) \quad (3.66)$$

Nyní stačí dokázat, že body  $C_{n-1}^S(v), C_n^S(v), C_1^{S'}(v)$  pro libovolné  $v \in \mathbf{v}$  leží na přímce. Pak jsou vektory lineárně závislé a podle vzorců pro derivace v krajních bodech jsou až na násobek (podíl vah a uzlových vektorů) stejné.

Musí být pro  $q \neq 0, q \in \mathbb{R}$ :

$$q \cdot C_{n-1}^S(v) C_n^S(v) = C_n^S(v) C_1^{S'}(v)$$

$$q(C_n^S(v) - C_{n-1}^S(v)) = C_1^{S'}(v) - C_n^S(v)$$

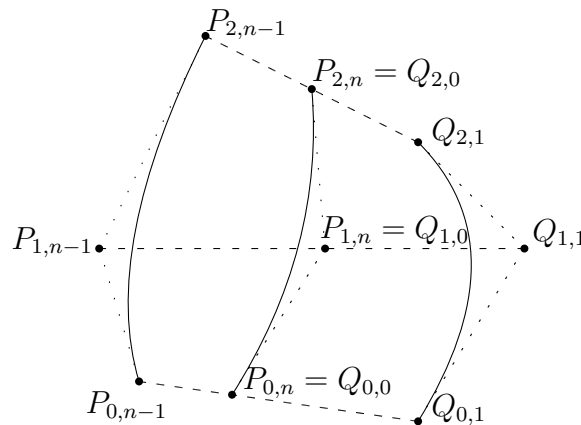
Vyjádríme:

$$\begin{aligned} C_n^S(v) - C_{n-1}^S(v) &= \sum_{i=0}^m w_{i,n} Q_{i,0} N_i^q(v) - \sum_{i=0}^m w_{i,n} P_{i,n-1} N_i^q(v) = \\ &= \sum_{i=0}^m w_{i,n} (Q_{i,0} - P_{i,n-1}) N_i^q(v) \end{aligned}$$

a

$$\begin{aligned} C_1^{S'}(v) - C_n^S(v) &= \sum_{i=0}^m w_{i,n} (P_{i,n-1} (1 - k) + k Q_{i,0}) N_i^q(v) - \sum_{i=0}^m w_{i,n} Q_{i,0} N_i^q(v) = \\ &= (k - 1) \sum_{i=0}^m w_{i,n} (Q_{i,0} - P_{i,n-1}) N_i^q(v) \end{aligned}$$

Oba výrazy jsou stejné až na násobek  $k - 1$ , na němž nezáleží. Proto jsou lineárně závislé a určují parciální derivaci v druhém směru pro obě plochy. Obě parciální derivace na hraniční křivce jsou stejné, proto pokud je splněn předpoklad, je navázání  $C_1$  spojitě. Na obr. 3.54 vidíme schéma jednoduchého navázání dvou ploch



Obrázek 3.54: Schéma hladkého napojení dvou ploch

### 3.5.1 Podmínky hladkého napojení NURBS ploch

Mějme dány dvě obecné NURBS plochy  $S, S'$ .

**Plocha  $S$**

síť  $m \times n$  bodů

řídící body  $P_{ij}$  s váhami  $w_{ij}$

řádkový a sloupcový uzlový vektor  $\mathbf{u}, \mathbf{v}$

řádkový a sloupcový stupeň  $p, q$

**Plocha  $S'$**

síť  $l \times r$  bodů

řídící body  $Q_{ij}$  s váhami  $w'_{ij}$

řádkový a sloupcový uzlový vektor  $\mathbf{u}', \mathbf{v}'$

řádkový a sloupcový stupeň  $p', q'$

Jestliže platí:

1.  $\mathbf{v}' = \mathbf{v}$
2.  $q = q'$
3.  $w_{i,n-1} = w_{i,n} = w'_{i,0} = w'_{i,1}$  pro  $i = 0, 1, \dots, m$
4.  $Q_{i,0} = P_{i,n}$  pro  $i = 0, 1, \dots, m$
5. pro libovolné  $k \neq 0, k \in \mathbb{R}$

$$P_{i,n-1}Q_{i,1} = k.P_{i,n-1}Q_{i,0} = k.P_{i,n-1}P_{i,n} \text{ pro } i = 0, 1, \dots, m$$

pak jsou plochy  $S$  a  $S'$  hladce napojeny.

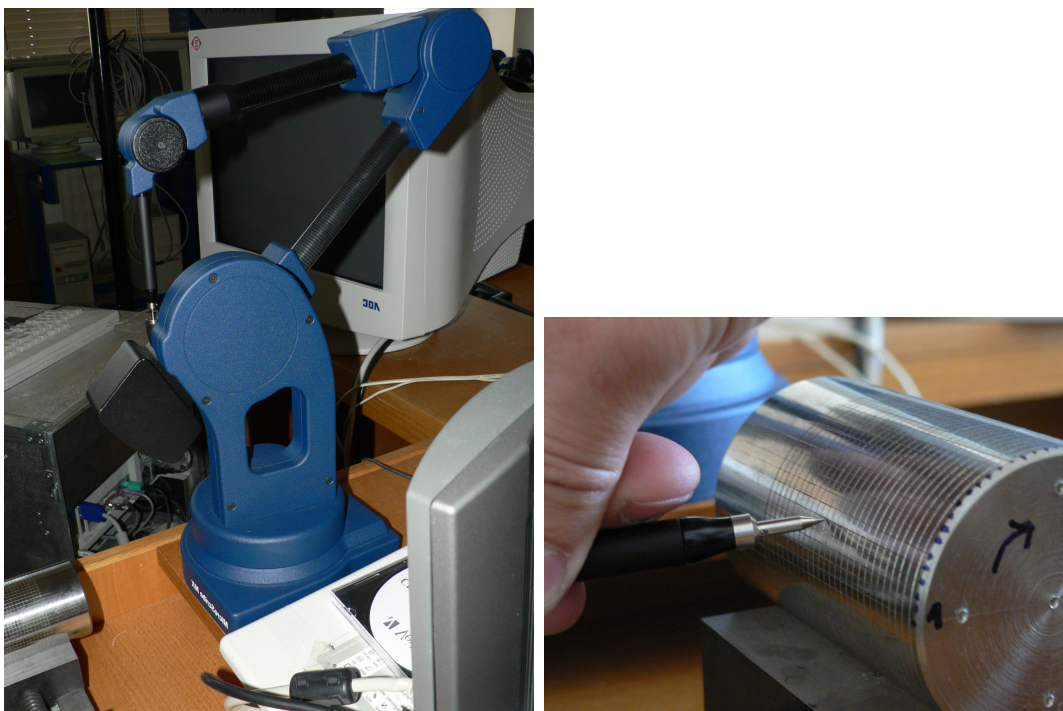
## 3.6 Využití interpolace pomocí NURBS křivek

### 3.6.1 Obecný postup

Ve spolupráci s ústavem strojírenské technologie – odbor technologie obrábění je řešen projekt pro medicínské využití. Jedná se o výrobu kloubních náhrad vhodných pro operace kolenních kloubů. Naše výsledky budou po testování použity v nemocnici u sv. Anny v Brně, kde se ročně provádí asi 100 těchto operací.

Naším úkolem je získat přesné vyjádření plochy kloubu a tu transformovat do souřadnic pro dráhu frézky, která kloub z kovového hranolu vyrobí. I když se jedná o naměřené hodnoty, které nejsou přesné, nemůžeme použít metodu nejmenších čtverců nebo další statistické metody. Plocha kloubu je totiž velmi členitá a není možné najít její jednoduchou reprezentaci. Proto bereme naměřené hodnoty jako hodnoty přesné a ty interpolujeme.

Metoda byla testována na půlválci. Nejdříve je pomocí přístroje Microscribe (obr. 3.55) nasnímaná síť bodů. Předmět musí být velmi dobře připevněn a pomocí snímací tužky (obr. 3.55) se zapisují souřadnice nasnímané pozice.



Obrázek 3.55: Přístroj Microscribe a snímání bodů válce

Tyto body jsou interpolovány pomocí NURBS křivek. Všechny váhy bodů nastavíme implicitně na hodnotu jedna. Budeme tedy pracovat pouze s neuniformními B-spline křivkami.

Ze vstupních bodů vypočítáme tvar uzlového vektoru pomocí středové či tětiové metody. V praxi budeme používat středovou metodu, která je vhodná, pokud ve vstupních datech existují výraznější přechody. Následně sestavíme matici pro nalezení řídicích bodů. Výsledný uzlový vektor a řídicí body vypíšeme do souboru a předáme ke zpracování frézku. NURBS křivku zobrazíme v programu Gnuplot, později bude vizualizace prováděna s použitím OpenGL.

### 3.6.2 Matematický aparát

Předpokládejme, že dostaneme množinu  $(n + 1)$  naměřených hodnot

$$\mathbf{Q}_k, k = 0, \dots, n.$$

Chceme sestavit neracionální neuniformní B-spline křivku stupně  $p$ , která bude interpolovat tyto body, tj.

$$\mathbf{Q}_k = C(\bar{u}_k) = \sum_{i=0}^n N_i^p(\bar{u}_k) \mathbf{P}_i. \quad (3.67)$$

Nejdříve musíme spočítat vhodné hodnoty  $\bar{u}_k$ . K jejich určení použijeme tětiovou nebo středovou metodu. Nechť  $d$  značí absolutní délku tětivy.

**Tětivová metoda:**

$$d = \sum_{k=1}^n |\mathbf{Q}_k - \mathbf{Q}_{k-1}|$$

Potom

$$\begin{aligned} \bar{u}_0 &= 0 \quad \bar{u}_n = 1 \\ \bar{u}_k &= \bar{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} \quad k = 1, \dots, n-1 \end{aligned} \quad (3.68)$$

**Středová metoda:**

$$d = \sum_{k=1}^n \sqrt{|\mathbf{Q}_k - \mathbf{Q}_{k-1}|}$$

Potom

$$\begin{aligned} \bar{u}_0 &= 0 \quad \bar{u}_n = 1 \\ \bar{u}_k &= \bar{u}_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{d} \quad k = 1, \dots, n-1 \end{aligned} \quad (3.69)$$

Pomocí tětivové nebo středové metody spočítáme parametry  $\bar{u}_0, \dots, \bar{u}_n$ . Abychom dostali v rovnici (3.67) řešitelnou soustavu, je nutné provést metodu nazvanou průměrování – *averaging*:

$$u_0 = \dots = u_p = 0 \quad u_{m-p} = \dots = u_m = 1,$$

$$u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} \bar{u}_i \quad j = 1, \dots, n-p. \quad (3.70)$$

Po výpočtu uzlových vektorů zbývá určit řídicí body. Rovnice (3.67) určuje  $(n+1) \times (n+1)$  lineárních rovnic neznámých  $P_i$ . Její prvky dostaneme výpočtem báзовých funkcí s parametry  $\bar{\mathbf{u}}$  pro uzlový vektor  $\mathbf{u}$ .

$$\begin{pmatrix} N_0^2(\bar{u}_0) & N_1^2(\bar{u}_0) & \dots & N_n^2(\bar{u}_0) \\ N_0^2(\bar{u}_1) & N_1^2(\bar{u}_1) & & N_n^2(\bar{u}_1) \\ \vdots & & \ddots & \\ N_0^2(\bar{u}_n) & N_1^2(\bar{u}_n) & \dots & N_n^2(\bar{u}_n) \end{pmatrix} \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_n \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_0 \\ \mathbf{Q}_1 \\ \vdots \\ \mathbf{Q}_n \end{pmatrix} \quad (3.71)$$

Algoritmus interpolace můžeme zapsat jako:

**Vstup:**  $n+1$  naměřených hodnot  $Q_k$ , požadovaný stupeň  $p$

**Výstup:** řídicí body  $P_i$ , délka uzlového vektoru  $m$ , uzlový vektor  $U$



```

CurveInterpol(n,Q,p,m,U,P){
  m = n+p+1; // delka uzloveho vektoru
  spoctiUpruh;
  SpoctiU;
  Inicializace pole A na nulove hodnoty;
  SpoctiKoefficientyMaticeA();
  P = GaussovaEliminace(A);
  vykresli krivku s vypoctenymi parametry
}

```

**Příklad:**

Máme dáno 5 naměřených hodnot bodů  $\mathbf{Q}_i$ ,  $i = 0, \dots, 4$ . Naším úkolem je podle předchozího postupu najít NURBS křivku, která tyto body interpoluje.

Body  $\mathbf{Q}_i$ : (0, 0, 0) (1, 1, 0) (3, 0, 0) (5, -1, 0) (8, 5, 0)

Nejdříve vypočítáme vektor  $\bar{\mathbf{u}}$ . Potom z něj metodou *averaging* klasický uzlový vektor pro křivku stupně dva. Sestavíme matici podle rovnice (3.71) a Gaussovou eliminací vypočítáme řídicí body  $P_i$ .

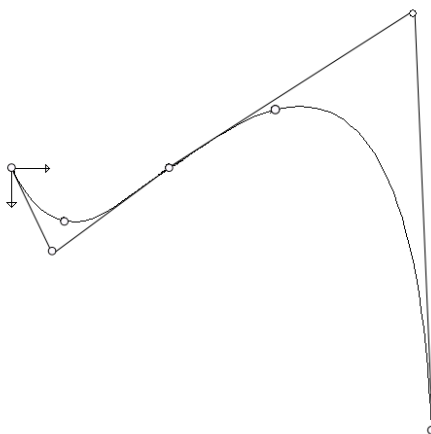
Výsledné hodnoty

Uzlový vektor: (0, 0, 0, 0.201059, 0.378601, 1, 1, 1)

řídicí body: (0, 0, 0) (0.786373, 1.57043, 0) (3.00158, -0.0251676, 0)

(7.64779, -2.953, 0) (8, 5, 0)

Výsledná interpolační křivka je na obr. 3.56).

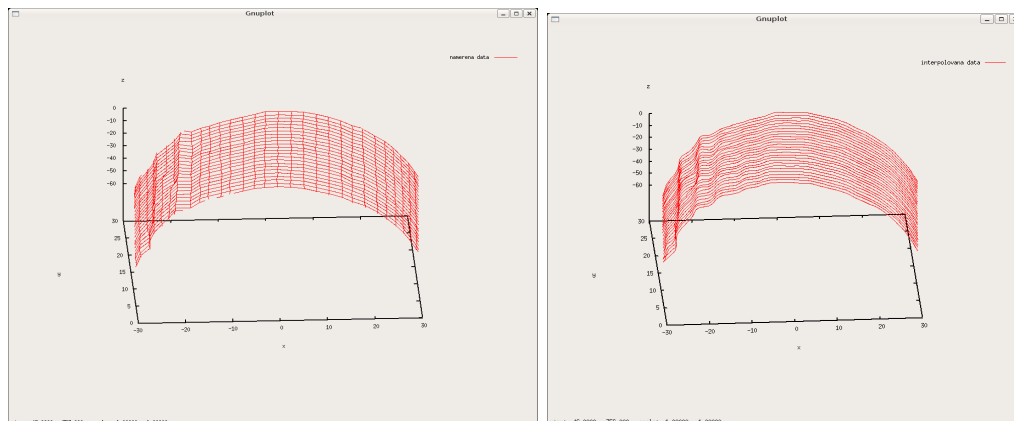


Obrázek 3.56: Interpolace pomocí NURBS – původní a nové řídicí body

### 3.6.3 Interpolace na válci

Přístrojem Microscribe byly naměřeny vstupní hodnoty (asi 1000 hodnot). Na ně potom po jednotlivých řádcích byla aplikována výše popsaná metoda. Výsledné křivky byly spojeny do jedné, která určila s daným (libovolně velkým) krokem body, určené pro frézku.

Na frézce byl válec znovu vyroben. Vzhledem k nepřesnostem při snímání bodů se projevilo malé zkřivení. To dokazuje funkčnost metody, která s velkou přesností interpoluje vložená data. Vstupní a vypočítané hodnoty testovacího půlválce jsou zobrazeny na obr. 3.57. A na obr. 3.58 je ukázka z výroby půlválce na frézce.



Obrázek 3.57: Vstupní a vypočítaná data pro testovací půlválec



Obrázek 3.58: Výroba výsledného válce na frézce

### 3.7 Nové metody modelování ploch použité v GIS

GIS (geografický informační systém, geographic information system) je informační systém se schopností práce s geodaty – tj. prostorovým vymezením objektů a prostorovými vztahy objektů. GIS popisuje okolní svět s využitím prostorové lokalizace.

Jedna z přesných a vyčerpávajících odborných definic GIS zní: Geografický informační systém je organizovaný souhrn počítačové techniky, programového vybavení, geografických dat a zaměstnanců navržený tak, aby mohl efektivně získávat, ukládat, aktualizovat, analyzovat, přenášet a zobrazovat všechny druhy geograficky vztahovaných informací.

V této sekci se budeme zajímat jen o část z tohoto procesu a to zobrazování a vyhodnocování zemského povrchu z naměřených dat LiDAR technologií pomocí T-spline ploch. Teoretický základ T-spline je popsán v části 2.4.

LiDAR (Light Detection and Ranging) technologie je jednou z možností, jak rychle získat informace o určité oblasti. Snímač umístěný ve vrtulníku či na křídle letadla vysílá laserový paprsek a jeho odrazy se zapisují. Poté jsou vypočítány přesné souřadnice naměřených bodů.

Výsledkem je mrak bodů, pro který je nutné další zpracování k identifikaci budov, stromů a povrchu. Nevýhodou LiDAR měření je omezená přesnost při snímání blízkých budov, chyby při snímání v GIS/INS nebo špatná odrazivost některých povrchů. I přesto je LiDAR technologie vynikající při přímém měření hloubky objektů.

### Výhody použití T-spline

- Není nutné mít pravidelnou síť bodů.
- Po vložení bodu (např. při dodatečných měřeních částí) není nutné překreslovat celou plochu.
- Lze provést lokální zjemnění sítě, kde je nutná větší přesnost zobrazení.
- Lze vytvořit ostré hrany – kraje vodních nádrží, budovy.
- Algoritmem T-spline simplification lze zredukovat počet řídicích bodů na méně než polovinu bez změny výsledné plochy.

Při zobrazování dat lze zvolit jeden ze dvou přístupů – aproximační nebo interpolační. Interpolační postup je již částečně popsán v Zheng et al. (2006). Nevýhodou tohoto postupu je vysoká paměťová i časová náročnost pro větší počet bodů. Zheng et al. (2006) vychází z pravidelné sítě bodů se stejnou vahou. Následně jsou generovány ekvidistantní uzlové vektory k jednotlivým bodům. Potom přichází na řadu interpolace, tj. výpočet soustavy rovnic, jejichž počet je roven počtu interpolovaných bodů. Zheng et al. (2006) přidává local refinement algoritmus pro zjemnění některých částí sítě.

V praxi jsou však vstupní body neuspořádané v mraku bodů a uzlový vektor není ekvidistantní. Také výpočet soustavy rovnic pro například milion bodů je již velmi časově náročné. Z tohoto důvodu byl vybrán aproximační přístup. Výsledná plocha neprochází naměřenými body, ale vzhledem k neekvidistantnímu uzlovému vektoru, který použijeme, bude přirozeněji popisovat terén.

Využití T-spline aproximace pro zobrazení mraku bodů dosud nebylo publikováno. V Lichy – Bekhahn (1999) je popsána metoda aproximace pouze pomocí B-spline ploch s úpravami řídicích bodů pro zobrazení ostrých přechodů mezi vodními plochami a zemí. Náš přístup byl poprvé publikován v (Procházka – Procházková (2007)).

### Algoritmus řešení:

1. Nalézt rovinné plochy (střechy, budovy) a oddělit je od ostatních bodů.
2. Uspořádat zbývající body.
3. Každému bodu přiřadit uzlové vektory potřebné pro T-spline síť.
4. Algoritmem T-spline simplification odstranit nadbytečné body bez změny tvaru výsledné plochy.

5. Zobrazit budovy a objekty pomocí rovinných ploch.
6. Vypočítat výslednou T-spline plochu a zobrazit ji.

### 3.7.1 Detekce objektů - střechy budov, objekty

Detekce budov a rekonstrukce z LiDAR dat je již velmi dobře zpracovaná, proto zde provedeme pouze nástin jednotlivých metod.

Většina známých přístupů převádí mrak bodů do hloubkových obrazů a poté známými metodami detekuje budovy jako tvary s přímými hranami (více v Vosselman (1999), Alharthy – J. (2002)). Další algoritmus v You et al. (2003) přizpůsobuje uživatelem vybrané objekty (hranoly, válce, kvadriky) LiDAR datům.

Rychlá a relativně přesná metoda je uvedena v Verma et al. (2006). Je založena na rozkladu složitějších struktur střech do jednodušších. Obecně je dáno  $n$  naměřených bodů:

$$L = \{P_i \mid i = 1, \dots, n\}.$$

Z nich spočítáme kovarianční  $(3 \times 3)$  matici pro  $\delta$ -okolí jednotlivých bodů.

$$N_i = \{P_j \mid P_j \in L : |P_i P_j| < \delta\}$$

$$M_i = \frac{1}{|N_i|} \sum_{P \in N_i} (P - \mu_i)(P - \mu_i)^T$$

kde  $\mu_i$  je průměr všech bodů v  $N_i$ .

Vzhledem k nepřesnostem při měření vlivem zaokrouhlování není hodnota matice  $M_i$  pro rovinné body rovna přesně dvěma. Proto se provede srovnání velikosti vlastních čísel matice. Poté se rozliší, zda se jedná o body na budovách či na zemi a jednotlivé roviny jsou potom topologicky uspořádány. Postup je vidět na obr. 3.59.

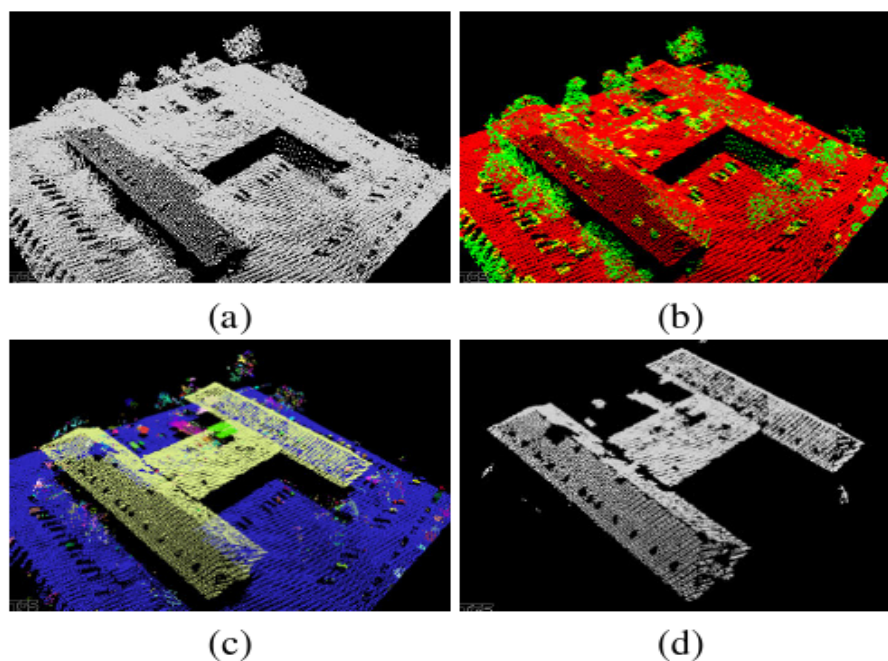
### 3.7.2 Uspořádání mraku bodů

Pro uspořádání mraku bodů jsme navrhli metodu nazvanou *y-cube*, která spočívá v rozdělení oblasti na jednotkové hranoly (velikost jednotky se zvolí vzhledem k velikosti snímané oblasti a kroku snímání) v ose  $y$  podle obr. 3.60. Následně setřídíme prvky v jednotlivých hranolech pomocí metody quick sort. V jazyce C ji obsahuje knihovna STL. Na obr. 3.61 je zobrazen vstupní mrak bodů a jeho uspořádání. Body jsou spojeny úsečkami a vytváří tedy kontrolní síť pro T-spline.

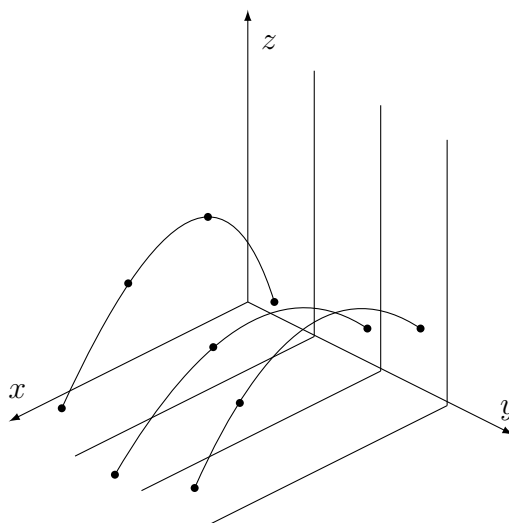
### 3.7.3 Výpočet uzlových vektorů

Body T-mesh jsou definovány dvěma uzlovými vektory, které popisují jejich lokální okolí. Uzlový vektor pro směr osy  $x$  určíme středovou metodou, která je popsána v části 3.6. Středová metoda byla vybrána, protože velmi dobře popisuje změny povrchu.

Uzlový vektor ve směru osy  $y$  byl zadán ekvidistantní. Při snímání mají naměřené body v jednotlivých hranolech přibližně stejné vzdálenosti. Druhým důvodem je to, že pro další úpravy (například vložení nového bodu, T-spline simplification apod.) by bylo časově náročné a programátorsky těžko zvládnutelné přepočítávat všechny uzlové vektory.



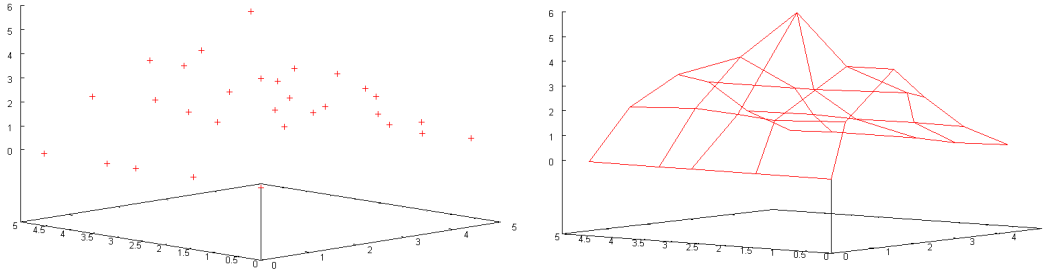
Obrázek 3.59: Vstupní mrak bodů a postupná klasifikace bodů, zdroj Verma et al. (2006)



Obrázek 3.60: Třídění bodů do hranolů – podle osy  $y$  (křivky jsou vykresleny pro názornost)

### 3.7.4 Zjednodušení plochy metodou T-spline simplification

V této části se aplikuje obecný algoritmus uvedený v části 2.4.3. Plocha vygenerovaná v předchozích bodech tvoří vstupní plochu pro algoritmus. Výpočet je časově náročnější, neboť testujeme matici  $D_{in}$  pro všechny body a v každém kroku počítáme přechodovou matici  $M_{in}$ . Tento postup je vhodný spíše pro menší datové soubory.



Obrázek 3.61: Mrak bodů a jeho uspořádání

### 3.7.5 Výpočet výsledné T-spline plochy a zobrazení

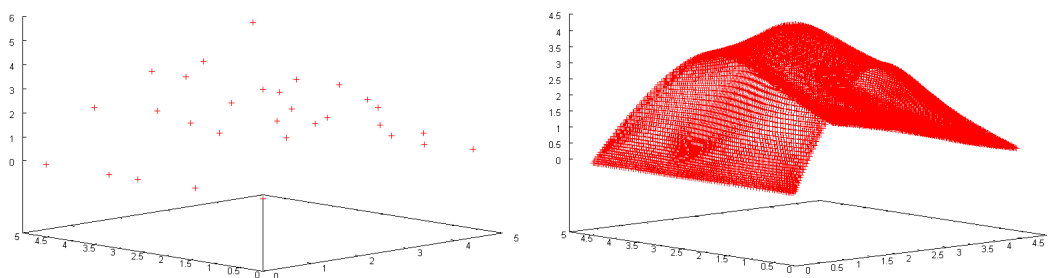
**Vstup:** posloupnost bodů  $\mathbf{P}$ , každý bod má přiřazený dva uzlové vektory odvozené z T-mesh –  $\mathbf{s}, \mathbf{t}$ , stupeň plochy – 3, parametry  $u, v$ .

**Výstup:** bod plochy  $S(s, t)$ .

1. Pro parametry  $s, t$  nalezneme body vlivu, tj. ty body, jejichž uzlové intervaly obsahují hodnoty  $s, t$ .
2. Pro vybrané body se vypočítají báze funkce  $N_{0i}^3(s), N_{0i}^3(t)$ .
3. Vyhodnotí se výraz  $\sum_i \mathbf{P}_i N_{0i}^3(s), N_{0i}^3(t)$ , který určuje souřadnice výsledného bodu.

Báze funkce  $N_i^3(s), N_i^3(t)$  nepočítáme deBoorovým algoritmem jako u NURBS. Použijeme přímý výpočet pouze s dosazením parametru a uzlového vektoru, neboť pro třetí stupeň je tvar polynomů  $N_{0i}^3$  vždy stejný. Pro třetí stupeň ploch je dán rovnicemi (3.72). Výsledná plocha pro mrak bodů rozříděný předchozím algoritmem je vykreslena na obr. 3.62.

$$N[\mathbf{t}_i](t) = \begin{cases} \frac{(t-t_0)^3}{(t_1-t_0)(t_3-t_0)(t_2-t_0)} & \text{pro } t \in \langle t_0, t_1 \rangle \\ \frac{(t-t_0)^2(t_2-t)}{(t_2-t_1)(t_3-t_1)(t_2-t_0)} + \frac{(t_3-t)(t-t_0)(t-t_1)}{(t_2-t_1)(t_3-t_1)(t_3-t_0)} + \frac{(t_4-t)(t-t_1)^2}{(t_2-t_1)(t_4-t_1)(t_3-t_1)} & \text{pro } t \in \langle t_1, t_2 \rangle \\ \frac{(t-t_0)(t_3-t)^2}{(t_3-t_2)(t_3-t_1)(t_3-t_0)} + \frac{(t_4-t)(t_3-t)(t-t_1)}{(t_3-t_2)(t_4-t_1)(t_3-t_1)} + \frac{(t_4-t)^2(t-t_2)}{(t_3-t_2)(t_4-t_2)(t_4-t_1)} & \text{pro } t \in \langle t_2, t_3 \rangle \\ \frac{(t_4-t)^3}{(t_4-t_3)(t_4-t_2)(t_4-t_1)} & \text{pro } t \in \langle t_3, t_4 \rangle \\ 0 & \text{jinak} \end{cases} \quad (3.72)$$

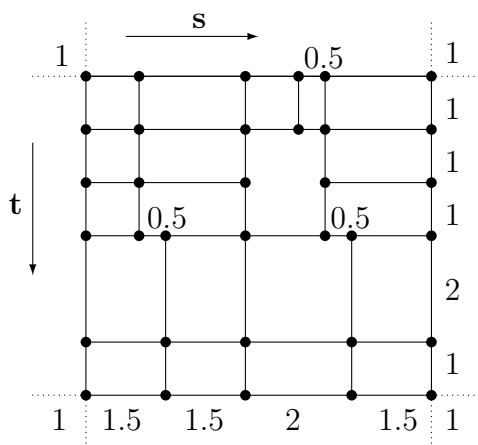


Obrázek 3.62: Výsledná T-spline plocha pro mrak bodů

### Příklady:

## 1. Obecná T-spline plocha

T-spline plocha je dána sítí znázorněnou na obr. 3.63 pomocí 34 řídicích bodů a uzlových vektorů odvozených z této sítě. Na obr. 3.64 je znázorněná výsledná plocha pro body, jejichž  $z$ -ová souřadnice je nulová a pro obecné  $z$ -ové souřadnice. Je zřejmé, že přímé zadávání ploch pomocí T-mesh a odhad tvaru plochy je velmi obtížné.



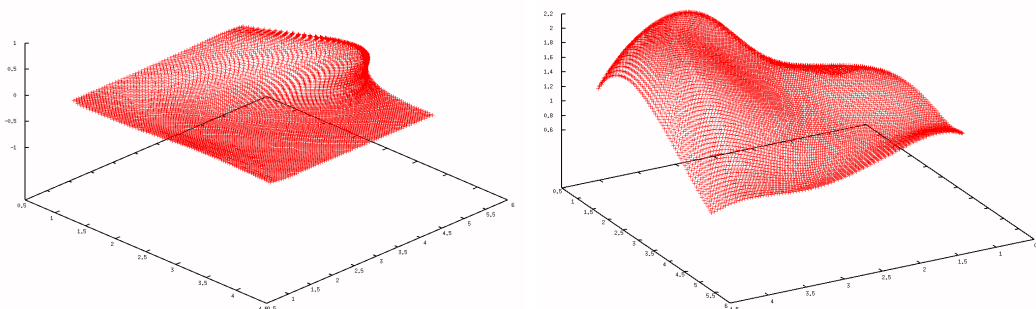
Obrázek 3.63: Obecná T-mesh

## 2. Vytvoření ostré hrany

Pomocí volby uzlového vektoru pro body sítě T-spline plochy lze modelovat ostré hrany a vrcholy. Tento způsob je ale vhodný pouze pro přírodní objekty - jezera, přehrady. Pro detekci budov, resp. střech je vhodnější použít některý algoritmus uvedený v části 3.7.1 a nahradit městská území polygonální reprezentací budov. Nastavení parametrů pro hrany není jednoduché a vedlo by při použití na desítkách objektů k velkému zpomalení programu a výsledek by nebyl tak dobrý, jako u již známých uveřejněných metod.

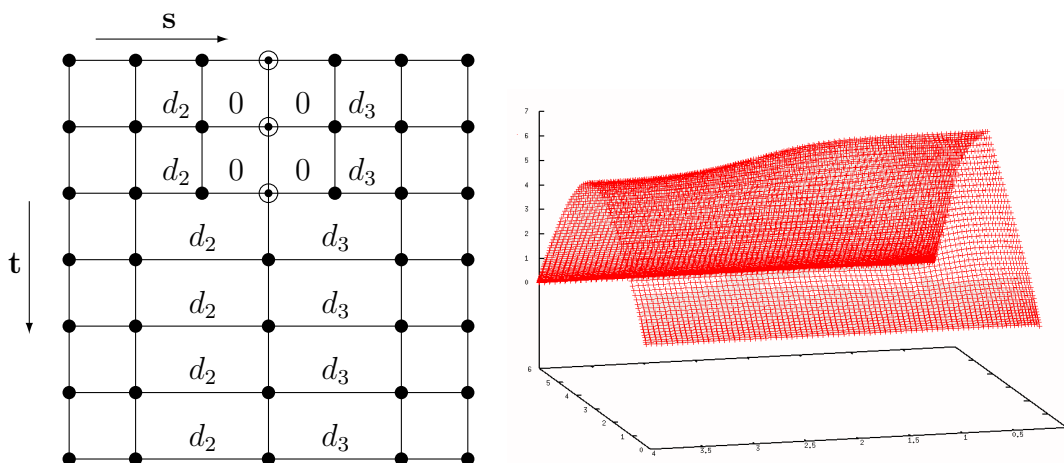
K vytvoření ostrých hran použijeme vlastnost spojitosti báзовých funkcí. Pokud se uzel opakuje v uzlovém vektoru, je spojitost v bodě odpovídajícímu tomuto parametru snížena o počet opakování.

Používáme stupeň třetí, proto je spojitost na celé ploše  $C^3$ . Pokud se bude jeden uzel

Obrázek 3.64: Obecná T-spline plocha ( $z = 0$ ,  $z$  obecné)

opakovat v uzlovém vektoru třikrát (podle vlastnosti 6 na str. 33), dostáváme spojitost  $C^0$ , která zapříčiní vznik hrotu.

Na obr. 3.65 je vytvořena síť pro plochu s hranou. Označené body způsobí snížení spojitosti. Jejich uzlový vektor  $\mathbf{s} = (d_1, d_1 + d_2, d_1 + d_2, d_1 + d_2, d_1 + d_2 + d_3)$  obsahuje součet  $d_1 + d_2$  třikrát, tedy spojitost je  $C^0$  – vzniká zde hrana, jak je vidět na výsledné ploše na obr. 3.65.



Obrázek 3.65: T-mesh a výsledná plocha s hranou

### 3. Vytvoření hrotu

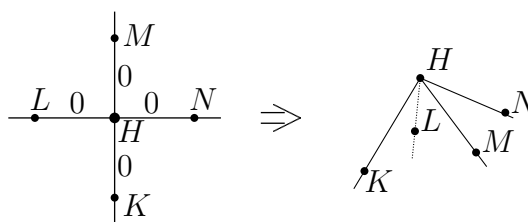
Pro znázornění hranatých těles je nutné znát zadání uzlů pro vytvoření vrcholu. Je nutné použít předchozí postup pro oba uzlové vektory (obr. 3.66) a ztotožnit některé řídicí body. Řešila jsem to způsobem, který je uveden na obr. 3.67 a lze jej aplikovat na obecný vrchol, pokud známe jeho souřadnice.

Na obr. 3.67 mají tři různé body stejné souřadnice bodu  $A$ . To je z toho důvodu, že plochu musíme procházet dvěma směry a ve svislé hraně obě cesty splynou. Totéž platí i pro body, které leží na úsečkách  $AH$ . Body, které leží v okolí vrcholu po uzlu délky nula mají vliv na tvar hrotu.

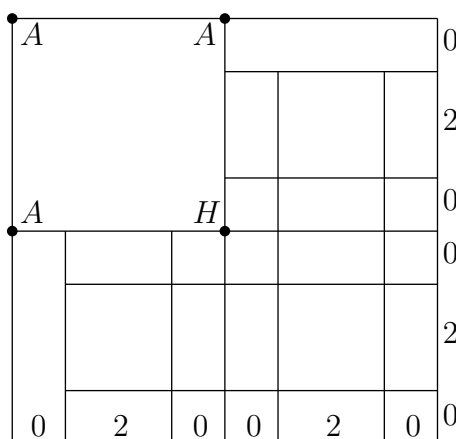
Část hranolu vytvořeného jako T-spline je zobrazena na obr. 3.68. Problémovým místem jsou spodní hrany. Plocha zde navazuje na body terénu, které již mají obecné



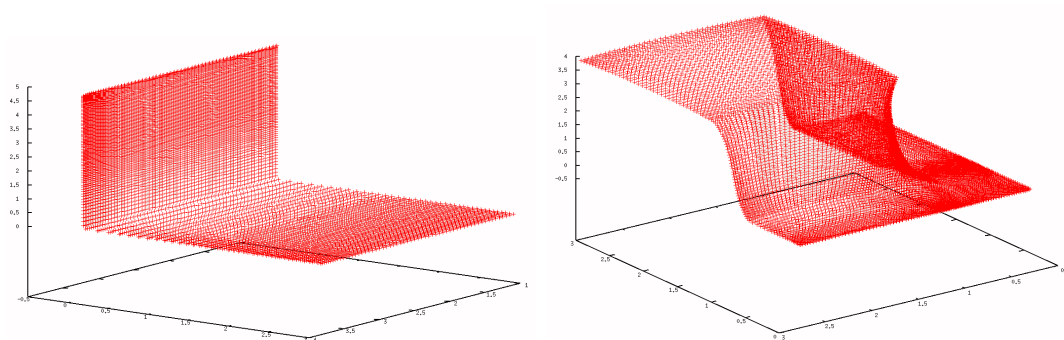
souřadnice. Podle definice T-spline jsou již započítávány do výpočtu, a proto zde dojde k hladkému přechodu. Řešením by bylo přidání dalších nulových bodů ve spodní části hranolu.



Obrázek 3.66: Tvorba hrotu pro T-spline



Obrázek 3.67: T-mesh pro tvorbu hrotu



Obrázek 3.68: Výsledná T-spline a) s hranou, b) s hrotem

### 3.8 NURBS programy

Ve své práci jsem se rozhodla srovnat některé významné produkty, které mají implementovány NURBS objekty, aby bylo patrné, jakým způsobem je popisovaná problematika

používána v praxi. Stručně popisují funkčnost několika 3D grafických studií (Autodesk Maya<sup>1</sup>, Form•Z<sup>2</sup>, Blender<sup>3</sup>) a knihoven (NURBS++, C Code) na Internetu pro práci s NURBS.

### 3.8.1 Open Source knihovny

#### NURBS++

Knihovna se nachází na Internetu<sup>4</sup> a byla dokončena v roce 2002. Obsahuje základní funkce týkající se NURBS křivek a ploch.

Modul NURBS křivky obsahuje:

- vytvoření křivky libovolného stupně pomocí řídicích bodů v homogenních i nehomogenních souřadnicích,
- výpočet derivací v bodě křivky,
- algoritmus zvýšení stupně, vložení či odebrání uzlu,
- metody interpolace a aproximace vstupních bodů,
- konstrukci kružnice,
- promítnutí bodu na křivku.

Modul NURBS plochy obsahuje:

- tvorbu plochy libovolného stupně v homogenních nebo nehomogenních souřadnicích,
- výpočet parciálních derivací,
- zvýšení stupně plochy či vložení uzlu,
- aproximace a interpolace bodů metodou nejmenších čtverců,
- funkci **sweep** – protažení křivky podél jiné křivky,
- průmět bodu na plochu.

Vstupem k vykreslení ploch jsou jednotlivé body zadané souřadnicemi, uzlové vektory a stupeň. Výstup může být VRML soubor, POV-Ray soubor nebo RIB soubor. Při výstupu do Post-Script souboru lze pracovat pouze s B-spline křivkami, tedy NURBS křivkami bez vah.

Ukázka tvorby NURBS křivky:

---

<sup>1</sup>[www.autodesk.com/maya](http://www.autodesk.com/maya)

<sup>2</sup><http://www.formz.com/>

<sup>3</sup><http://www.blender3d.org>

<sup>4</sup><http://libnurbs.sourceforge.net/index.shtml>

```

Vector_HPoint3Df pts(4) ;
pts[0] = HPoint3Df(0,0,0,1) ;
pts[1] = HPoint3Df(30,0,0,1) ;
pts[2] = HPoint3Df(60,30,0,1) ;
pts[3] = HPoint3Df(90,30,0,1) ;
Vector_float knot(8) ;
knot[0] = knot[1] = knot[2] = knot[3] = 0 ;
knot[4] = knot[5] = knot[6] = knot[7] = 1 ;
NurbsCurvef curve(pts,knot,3) ;

```

Modifikace transformační maticí je prováděna s pomocí třídy `MatrixRT<T>`, která obsahuje předdefinované matice pro posunutí, změnu měřítka, otáčení. Uvedený příklad udělá tuto transformaci: posunutí o vektor  $t_x, t_y, t_z$ , rotaci kolem osy  $z$ , poté kolem osy  $y$  a pak kolem osy  $x$ . Následně se provede změna měřítka metodou `scale`.

```

MatrixRT<float> Tx(rx,ry,rz,tx,ty,tz) ;
MatrixRT<float> Sx ;
Sx.scale(sx,sy,sz) ;
curve.transform(Tx*Sx) ;

```

### NURBS křivky v softwaru Maple

Práce v programu Maple je zatím možná pouze s NURBS křivkami. Na webových stránkách Maplesoft<sup>5</sup> je k dispozici knihovna pro práci s NURBS křivkami. Obsahuje tyto příkazy:

`Bspline, Curve, Hom, Open, RationalBspline, RationalCurve, Uniform`

Příkazy umí pouze vykreslit křivku danou zadanými body a uzlovým vektorem. Dále příkaz `Bspline` vypočítá báze funkce pro zadanou křivku a příkazem `plot` je lze vykreslit. NURBS plochy nejsou v softwaru Maple zahrnuty.

### C Code

Autor knihy *An Introduction to NURBS* David Rogers dal k dispozici na Internetu<sup>6</sup> krátké programy pro výpočet B-spline/NURBS křivek a ploch. Programy jsou napsané v programovacím jazyce C. Umí pouze základní úlohy – generování uzlových vektorů, výpočet báze funkcí a obecného bodu na křivce či ploše. Vstup i výstup je pouze prostřednictvím programového okna. Jedná se o doplněk výše uvedené knihy pro lepší pochopení v knize uvedených postupů. Autor uvádí, že algoritmy nejsou odladěné a také mohou být využívány pouze ke studijním účelům.

### 3.8.2 Form•Z

Komerčním softwarem americké firmy auto-des-sys je Form•Z. NURBS objekty se zde nazývají `nurbz`. V tomto softwaru však nejsou zadávány přímo jako NURBS plochy, ale

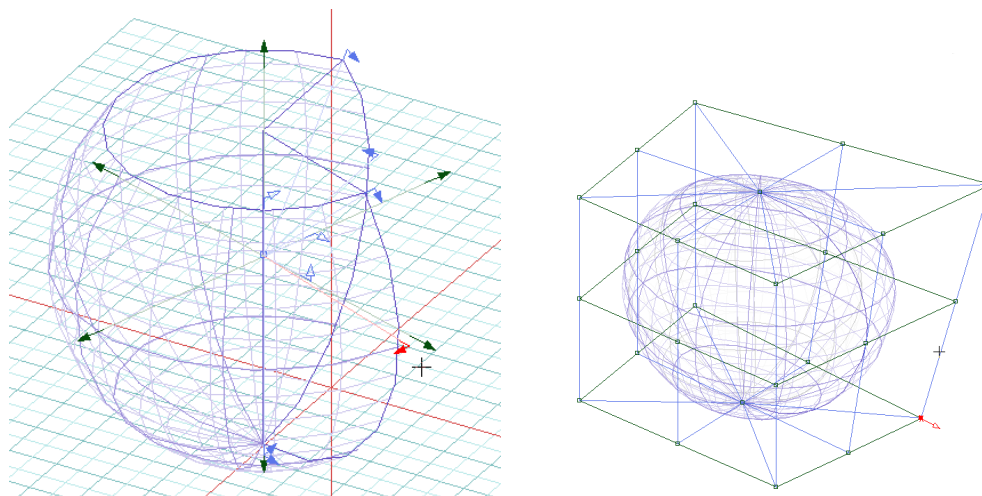
<sup>5</sup><http://www.maplesoft.com>

<sup>6</sup><http://www.nar-associates.com/nurbs/ccode.html>

jako analytické objekty – válec, kužel, krychle a také jako tzv. *objects of revolution* (rotační objekty) – koule, anuloid, paraboloid, hyperboloid. Pro každé těleso lze měnit poloměr, výšku a úhel rozevření. Pokud chcete pracovat s nurbz, je nutné již vytvořené objekty transformovat.

Přetransformované objekty lze upravovat pouze pomocí řídicích bodů. Nabízí se zde možnost změnit **LengthDegree**, **DepthDegree** – stupeň křivky v obou směrech. Z dalších funkcí lze zmínit **Loose and Tight Lofting**, které generují NURBS plochu mezi zadanými otevřenými či uzavřenými křivkami.

Za výhodu **Form•Z** lze považovat velmi jednoduché ovládání, což je k návrhu jednoduchých objektů dobré. Nevýhodou je nutnost předdefinovávat nakreslené objekty a nestabilita při neobvyklých úpravách.



Obrázek 3.69: Úpravy kulové plochy (software Form Z)

### 3.8.3 Blender

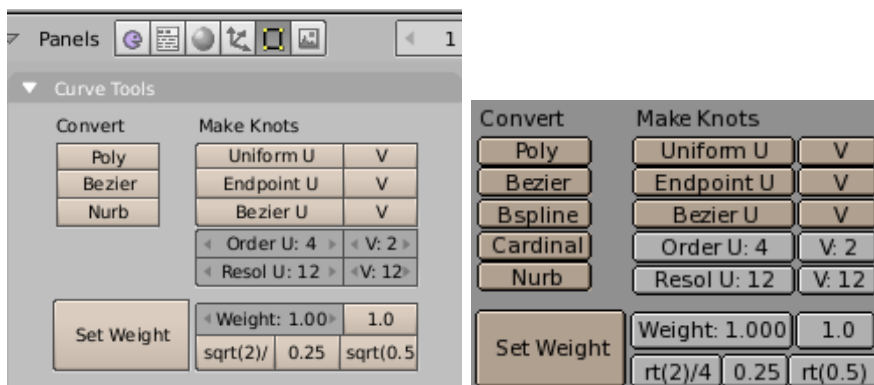
Blender je produktem nizozemské společnosti Not a Number Technologies. Je to kvalitní vizualizační program pro tvorbu 3D grafiky. Je určený pro modelování, rendering a animaci. Je to program, který je používán i profesionály. Jeho velkou výhodou je to, že se jedná o freeware. Je multiplatformní, užívá se v systémech Windows, MacOS, Solaris, Linux, FreeBSD. Je také modulární díky otevřenému API a skriptování v jazyce Python.

Všechna data scény se ukládají do jediného souboru s příponou .blend. Blender čte i zapisuje formáty tga, jpg, png, Iris, iff, avi, gif, tiff, mov a další. Poskytuje podporu importu a exportu do formátů dxf, Inventor a vrml souborů. Také lze vytvořit spustitelné soubory (exe) s interaktivními 3D aplikacemi.

#### NURBS objekty

K modelování objektů slouží mesh, NURBS křivky a plochy. Křivky se zadávají řídicím polygonem. V nabídce lze zvolit, zda kreslit klasickou aproximační křivku či interpolační křivku. Další možností je úprava vah. Jedinou předdefinovanou křivkou je kružnice. Mezi

NURBS tělesa patří válec, koule a anuloid. Obecně lze zadat NURBS plochu a poté editovat její body pomocí pozice či váhy.



Obrázek 3.70: Editace NURBS křivky a plochy

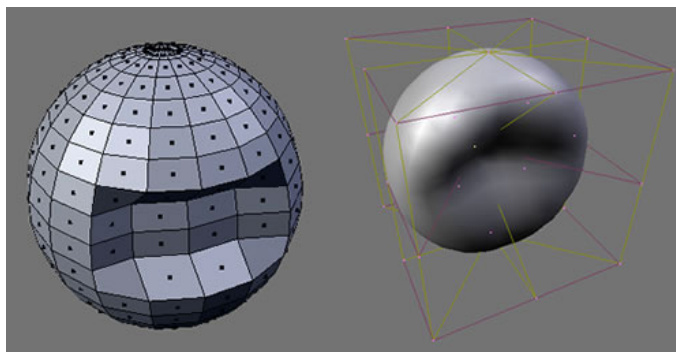
### 3.8.4 Autodesk Maya – learning edition

Software Maya používají profesionální návrháři i grafici. Práce v ní je založena převážně na NURBS objektech.

#### Podporované formáty souborů

##### 1. nativní

- Maya binary file: definuje geometrii, osvětlení, renderování a další vlastnosti scény.
- Maya ASCII file: Tento formát je určen pro uživatele, kteří chtějí buď editovat soubory Mayi nebo psát překladače do/z Mayi (např. export do VRML). Obsahuje tytéž definice scény, jako binární soubor, pouze je čitelnější.
- Maya interchange file format (IFF): Jedná se o výstupní formát Mayi, do kterého ukládá Maya vyrenderované obrazy v 8 nebo 16 bitovém RGB. Navíc je možné ukládat i alfa-kanál a 32 bitovou hloubkovou mapu.



Obrázek 3.71: Srovnání kulové plochy mesh a NURBS (Blender)

- Move files (.mov): ASCII soubor obsahující kanálová data.

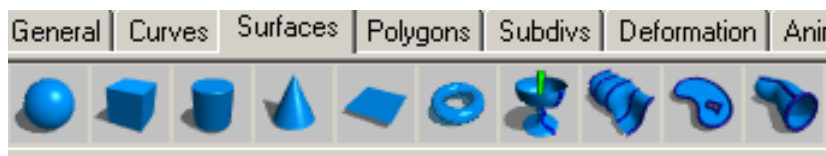
## 2. pomocí plug-in

Na jejich základě je postaven export a import do různých formátů. Maya verze 4.5.1 jich měla 14 včetně nativních. Maya verze 5 k nim přidala další čtyři. Mezi nejpodstatnější patří VRML2, OpenInventor, IGES, DXF a u Mayi 5 i Flash, Adobe Illustrator a zapouzdřený PostScript.

## NURBS objekty

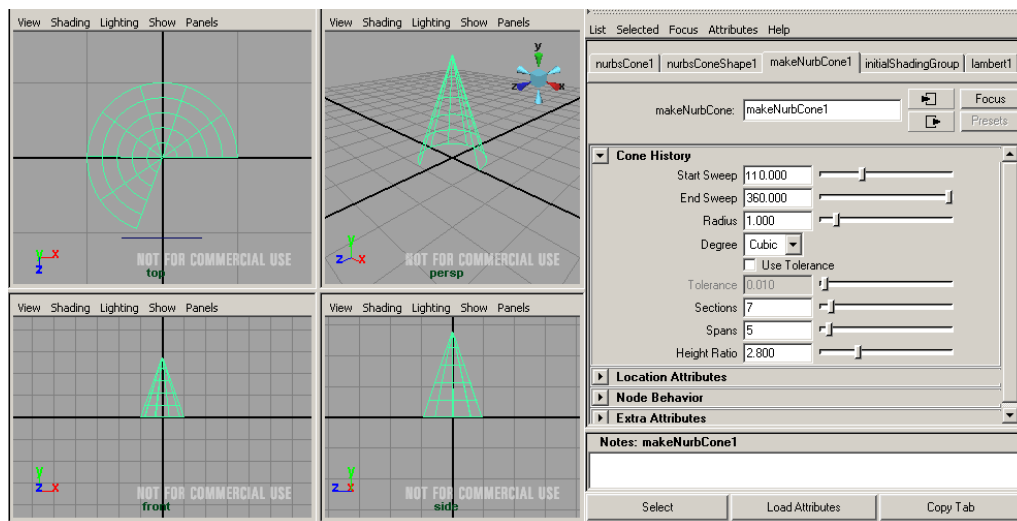
Kreslení NURBS křivek se provádí zadáváním řídicích bodů. Poté je možné křivku upravovat pomocí těchto bodů nebo přímo pozicí bodu křivky. Pro vytvoření hrotu je nutné násobné zadání řídicího bodu. Uzlový vektor se generuje automaticky tětiovou metodou nebo ekvidistantně. Upravovat váhy není možné.

Další důležitou částí jsou NURBS tělesa. V základní paletě jsou předdefinována tato tělesa: koule, anuloid, válec, kužel, rovina, hranol (viz obr. 3.72).



Obrázek 3.72: Základní nabídka NURBS těles

Tělesa jsou vždy vykreslena v základní poloze a v základní velikosti. Ke každému se objeví ovládací panel, kde je možné změnit polohu a tvar transformačními funkcemi – posun, rotace, měřítko, zkosení. Také je možné pracovat s parametry objektů – počet poledníků a rovnoběžek, uzlový vektor. Návrh kužele s ovládacími panely je zobrazen na obr. 3.73.



Obrázek 3.73: Ovládací panely pro kužel

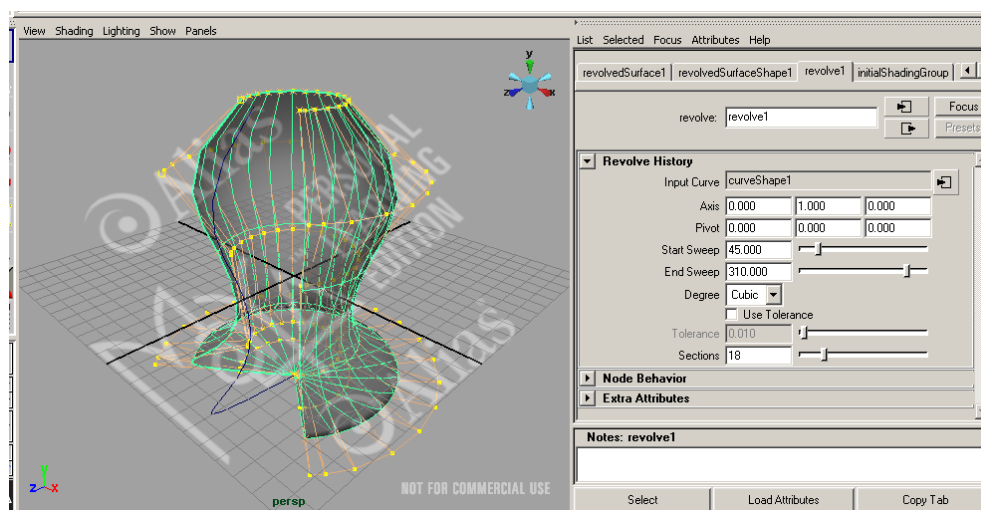
Možnosti modifikace NURBS objektů jsou:

- zobrazení řídicích bodů i konvexního obalu,
- zjemňování sítě – přidání rovnoběžek a poledníků,
- rozevření tělesa – možné v obou směrech (není možné u hranolu a roviny),
- změna stupně – pro kužel a válec pouze stupně jedna a tři, u ostatních těles pouze do stupně 5,
- přidání řídicích bodů – zjemnění uzlových vektorů.

Pro rotační tělesa vytvoříme základní křivku a poté funkce **NURBS Rotate** vytvoří rotační NURBS těleso rotací křivky kolem osy  $z$  (obr. 3.74). Ovládání pro rotační tělesa je velmi podobné klasickým NURBS tělesům

Speciálním případem plochy je hranol. Ten není zadán jako jedna plocha, ale je tvořen čtyřmi samostatnými stěnami, které lze nezávisle na sobě upravovat. Tento způsob má své nevýhody při práci s body na bočních hranách. Musí se upravovat na obou rovinách samostatně, což je obtížné, pokud chceme uzavřený hranol.

Další využití NURBS je prokládání křivek plochou a funkce **sweep** a **loft**. Při tvorbě obecné plochy musíme zadat čtyři hraniční křivky, které speciální funkcí spojit a poté je možné vytvořit NURBS plochu. Funkce **sweep** nakreslí dráhu jedné křivky po druhé. Funkce **loft** slouží k proložení libovolných křivek obecnou plochou.



Obrázek 3.74: Vytvoření rotačních NURBS těles

### 3.8.5 RFEM 3D

Jak již bylo řečeno, RFEM 3D je softwarový produkt, který využívá výsledků této práce. V loňském roce byly do prostředí přidány NURBS křivky a analytický způsob výpočtu jejich derivací (viz obr. 3.76).

NURBS plochy a tělesa popsaná v této práci jsou nyní v softwaru RFEM 3D v testovací fázi a budou zahrnuty v dalším release vydaném v letošním roce. Způsoby zadávání jednotlivých objektů jsou popsány v kapitole Výsledky a předpokládané zadávací okno je na obr. 3.77. Každé těleso jde upravovat změnou řídicích bodů a vah či změnou tvaru uzlových vektorů.

Dokonaleji zpracované NURBS objekty obsahuje pouze Maya a Blender. Existuje zde totiž možnost úpravy nejen pomocí řídicích bodů a parametrů, ale přímo deformaci plochy pomocí techniky FFD (Free-Form Deformation) poprvé popsané v Sederberg – Parry (1986). Další rozšíření NURBS modulu v RFEM 3D by mohlo vést právě tímto směrem.

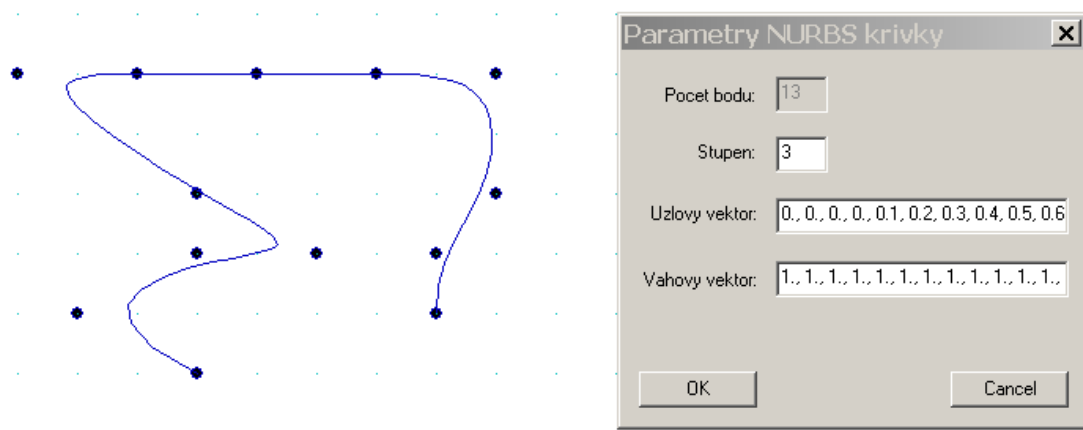
Dalším rozdílem je úprava rozevření tělesa. Ve studiu Maya lze měnit rozevření kdykoliv i po editaci tělesa, v RFEM 3D se úhel rozevření a počet rovnoběžek zadává pouze při vytvoření tělesa. Kromě těchto funkcí má nový modul v RFEM stejné vlastnosti a možnosti editace. Na obr. 3.78, 3.79 je ukázka již uskutečněných projektů vytvořených v RFEM 3D. Jedná se o fotbalový stadion v Mnichově a projekt patrových garáží v Ratingenu. Většina projektů se uskutečnila v Německu<sup>7</sup>.

Uvedené knihovny a toolboxy na Internetu jsou vhodné pouze pro jednoduché testování vlastností a chování NURBS objektů. Neobsahují předdefinovaná NURBS tělesa a vytvořené objekty nelze upravovat. Také nejsou propojeny s uživatelským prostředím. Toto všechno vytvořená třída v softwaru RFEM bez problémů zvládá.

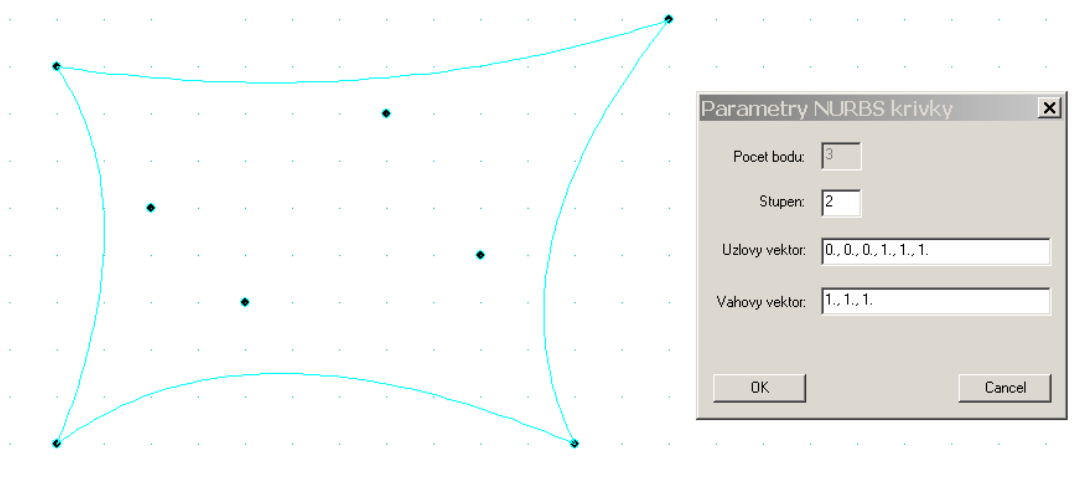
Obrázek 3.75: Zadávání NURBS těles

<sup>7</sup><http://www.dlubal.com>





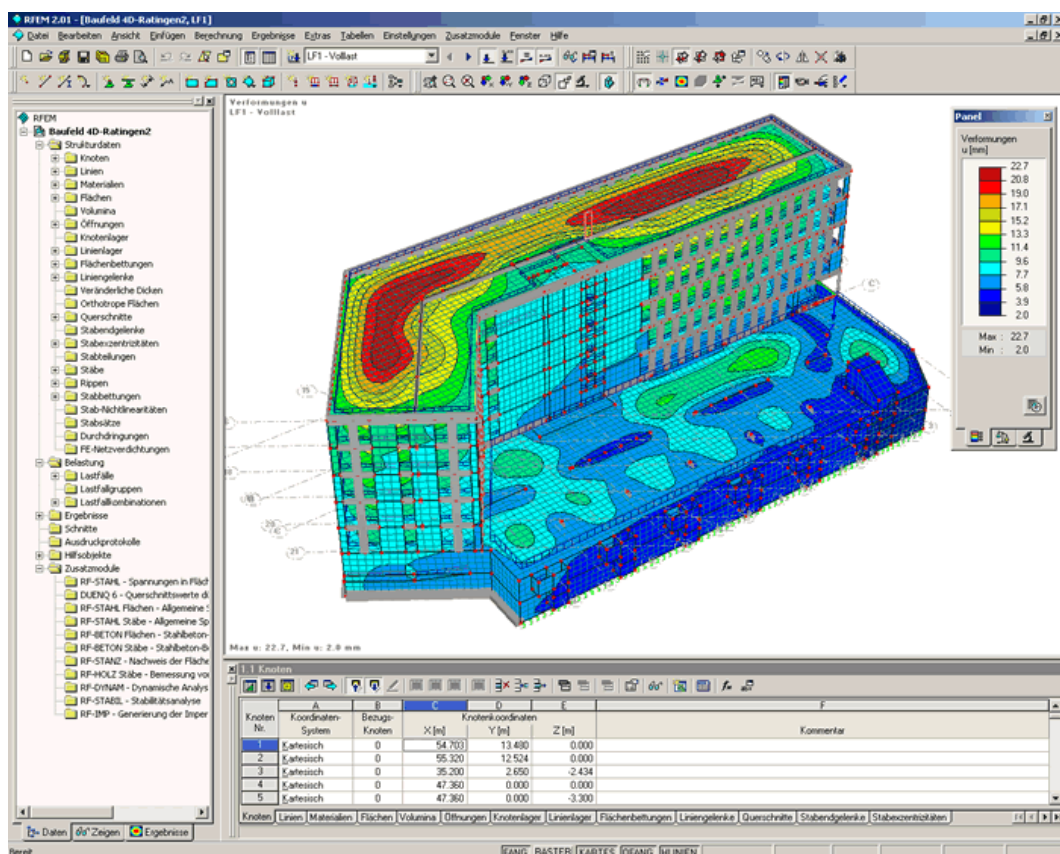
Obrázek 3.76: Zadání NURBS křivky



Obrázek 3.77: Zadání NURBS plochy hraničními křivkami



Obrázek 3.78: Fotbalový stadion v Mnichově



Obrázek 3.79: RFEM 3D – projekt garáže v Ratingenu

# Kapitola 4

## Závěr

Tato práce je zaměřena na oblast počítačové grafiky týkající se *free-form* objektů. Zvolený přístup není pouze implementační, ale také zkoumající, prohlubující a hledající nové poznatky a uplatnění. K tomu, aby šla práce novým směrem, je nutné shrnout dosavadní poznatky a dobře jim porozumět, což jsem popsala v kapitole 1 a 2.

K čemu by byly technikům nejmodernější způsoby zobrazování, kdyby nerozuměli jejich vnitřní struktuře, kdyby sebemenší odchylka od pevně daného zadání způsobila kolaps programu. Proto jsem se snažila popsat metody řešení s ohledem na vnitřní porozumění a kritická místa. Při určování vlivu uzlového vektoru jsou popsány a vykresleny báze funkce a ukázána jejich souvislost s řídicími body, při výpočtu derivací je nalezen speciální případ, kde derivace existuje i přes matematickou nespojitost, při programování NURBS objektů jsou podrobně popsány omezující podmínky a kritická místa.

V rámci doktorského studia jsem spolupracovala s firmou Fem Consulting, která je pobočkou německé firmy vytvářející software RFEM 3D pro návrh stavebních konstrukcí či strojních modelů. V release tohoto programu v roce 2006 byly implementovány NURBS křivky a výpočet derivací analytickou metodou. V současné době probíhá testovací fáze pro NURBS plochy a tělesa, která budou přidána v další verzi programu. Naše spolupráce bude nadále pokračovat na T-spline plochách, jejichž implementace se předpokládá v závěru roku 2007. Uvažujeme i o zahrnutí techniky Free Form Deforming.

Během svého studia jsem měla dojem, že jazyk čistě matematický a jazyk technicky praktický, nejsou blízcí kolegové, ale vzdálení příbuzní, kteří si příliš nerozumí. Na pohled jednoduchému problému týkajícímu se tenzorového součinu se věnuje spousta geometrií prostým konstatování, že existuje. Proto je v práci zahrnuta část týkající se NURBS ploch z pohledu tenzorového součinu a podrobně vysvětleno, proč je který přístup nekorektní a navržen tzv. kvazitenzorový součin, který je korektní pro oba jazyky.

Využití NURBS není pouze v počítačové grafice a designu. Ve spolupráci s ústavem strojírenské technologie – odbor technologie obrábění pracujeme na vývoji programu pro generování dráhy bodů, která by co nejlépe interpolovala vstupní data. Ve spolupráci s nemocnicí u sv. Anny v Brně budeme testovat tuto metodu na datech pro výrobu kolenních kloubních náhrad a dále spolupracovat na zlepšování a možnému uvedení softwaru do praxe.

Geografické informační systémy jsou jedním z dalších uplatnění matematických ploch. Aproximace či interpolace naměřených dat (například LiDAR technologií) je stěžejním

problémem při zobrazování zemského povrchu. B-spline přístup je v této oblasti již plně využíván. Ve své práci jsem navrhla nový způsob aproximace pomocí T-spline s využitím již známých metod pro další úpravy (detekce budov, stromů, hran).

Matematické modelování ploch je důležitou součástí počítačové grafiky i praktických oborů. Ve své práci jsem se pokusila shrnout dosavadní znalosti a najít jejich nové uplatnění. Spolupráce s komerční firmou či použití ve zdravotnictví potvrzuje, že zde uvedené poznatky jsou užitečné a použitelné v praxi. Věřím, že i můj další výzkum bude založen na poznatcích, které jsem nabyla při studiu na této disertační práci.

# Literatura

- ALEXANDR, L. *Výuka počítačové grafiky cestou www*. Diplomová práce. VUT Brno, 1999.
- ALHARTHY, A. – J., B. Heuristic filtering and 3d feature extraction from lidar data. *PCV02*. 2002.
- ASZÓDI, B. – CZUCZOR, S. – SZIRMAY-KALOS, L. NURBS fairing by knot vector optimization. *Full papers WSCG*. 2004.
- BARTELS, R. – FORSEY, D. Surface fitting with hierarchical splines. *ACM Trans. Graphics*. 1995, 14, 2, s. 134–161.
- BERCOVIER, M. – GOLDENTHAL, R. Spline Curve Approximation and Design Over the Knots Using Genetic Algorithms. *EUROGEN*. 2003.
- BERCOVIER, M. – GOLDENTHAL, R. Spline Curve Approximation and Design by Optimal Control Over the Knots. *Computing*. 2004, 72, s. 53–64.
- CLINE, A. Spline spaces. [online], Naposledy navštíveno 20. 2. 2007. Dostupné z: [www.cs.utexas.edu/users/cline/CS383D/Spline\\_spaces.pdf](http://www.cs.utexas.edu/users/cline/CS383D/Spline_spaces.pdf).
- BOOR, C. *A Practical Guide to Splines*. New York : Springer-Verlag, 1976. ISBN 0-387-90356-9.
- DOUPOVEC, M. *Diferenciální geometrie a tenzorový počet*. VUT Brno, 1999.
- FOLEY et al. *Computer Graphics – principles and practice*. New York : Addison–Wesley, 2005. ISBN 0-201-84840-6.
- GOLDMAN, R. The Ambient Spaces of Computer Graphics and Geometric Modeling. *Computer Graphics and Applications, IEEE*. 2000.
- HORÁK, P. *Algebra a teoretická aritmetika*. Brno : Rektorát Masarykovy univerzity Brno, 1991. ISBN 80-210-0320-0.
- JANYŠKA, J. – SEKANINOVÁ, A. *Analytická geometrie kuželoseček a kvadrik*. Brno : Masarykova Univerzita v Brně, 2001. ISBN 80-210-2604-9.
- JUHASZ, I. – HOFFMANN, M. Constrained shape modification of cubic B-spline curves by means of knots. *Computer-Aided Design*. 2004, 36, s. 437–445.

- JUHASZ, I. – HOFFMANN, M. The Effect of Knot Modifications on the Shape of B-spline Curves. *Journal for Geometry and Graphics*. 2001, 5, s. 111–119.
- KAZINNIK, R. – ELBER, G. Orthogonal Decomposition of Non-Uniform B-spline Spaces using Wavelets. *Eurographics 97*. 1997, 16, 3.
- LI, W. – RAY, N. – LÉVY, B. Automatic and Interactive Mesh to T-spline Conversion. *Eurographics Symposium on Geometry Processing*. 2006.
- LICHY, C. – BEKHAHN, V. B-Spline Surface Modelling with Adaptive de Boor Grids in Hydroinformatics. *ISSES 1999*. 1999.
- LOMTATIDZE, L. *Historický vývoj pojmu křivka*. PhD thesis, Masarykova univerzita v Brně, 2005.
- LYCHE, T. Knot removal for spline curves and surfaces. *Multivariate Approximation and Application*. 1993, s. 152–87.
- LYCHE, T. – MORKEN, K. – QUAK, E. Theory and algorithms for non-uniform spline wavelets. *Multivariate Approximation and Application*. 2001, s. 152–87.
- MARTIŠEK, D. *Matematické principy grafických systémů*. Brno : Littera, 2002. ISBN 80-85763-19-2.
- MOTL, L. – ZAHRADNÍK, M. Pěstujeme lineární algebru. [online], Naposledy navštíveno 10. 3. 2007. Dostupné z: <http://www.kolej.mff.cuni.cz/~lmotm275/skripta/mzahrad/node150.html>.
- PASEKA, J. Počítačová geometrie. [online], Naposledy navštíveno 20. 1. 2007. Dostupné z: <ftp://www.math.muni.cz/pub/math/people/Paseka/lectures/>.
- PIEGL, L. Modifying the shape of rational B-splines Part 1: curves. *Computer Aided Design*. 1989, 21, s. 509–18.
- PIEGL, L. On NURBS:Survey. *IEEE Computer Graphics and Applications*. 1991.
- PIEGL, L. – TILLER, W. *The NURBS Book*. Berlin : Springer-Verlag, 2002. ISBN 3-540-61545-8.
- PROCHÁZKA, D. – PROCHÁZKOVÁ, J. T-spline – nová metoda modelování povrchů. *GIS Ostrava*. 2007, 1, s. 85. ISSN 1213-2454.
- PROCHÁZKOVÁ, J. Derivatives of B-spline function. *25. konference o počítačové grafice a geometrii*. 2005, 25.
- PROCHÁZKOVÁ, J. – PROCHÁZKA, D. The Application of NURBS Surfaces in Engineering Surface. *Sborník konference Moderní matematické metody v inženýrství*. 2006a.
- PROCHÁZKOVÁ, J. – PROCHÁZKA, D. Implementation of Cylinder and Cone as NURBS Surfaces in Engineering Software. *Articles of Symposium on Computer Geometry SCG 2006*. 2006b.

- PROCHÁZKOVÁ, J. – PROCHÁZKA, D. Implementation of NURBS curve derivatives in engineering practice. *WSCG 2007 Poster Papers*. 2007.
- RAMSHAW, L. *Blossoms are polar form*. DEC Systems, 1989.
- SCHRODER, P. – SWELDENS, W. Spherical wavelets: Efficiently representing functions on the sphere. *SIGGRAPH*. 1995, s. 161–172.
- SEDERBERG, T. W. et al. T-splines and T-NURCCS. *ACM Transactions on Graphics*. 2003a, 22, 3, s. 477–484.
- SEDERBERG, T. An Introduction to B-Spline Curves. [online], Naposledy navštíveno 12. 4. 2007. Dostupné z: [tom.cs.byu.edu/~455/bs.pdf](http://tom.cs.byu.edu/~455/bs.pdf).
- SEDERBERG, T. – PARRY, S. Free-Form Deformation of Solid Models. *ACM Computer Graphics*. 1986, 20, 4.
- SEDERBERG, T. et al. T-splines and T-NURCCs. *ACM Trans. Graphic*. 2003b.
- SEDERBERG, T. et al. T-splines Simplification and Local Refinement. *ACM Trans. Graphic*. 2004.
- SEKANINA, M. *Geometrie II*. Praha : SPN Praha, 1988. ISBN r88U.
- SHENE, C. Introduction to Computing with Geometry Notes. [online], Naposledy navštíveno 11. 12. 2006. Dostupné z: <http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/notes.html>.
- TERZOPOULOS, D. – QIN, H. Dynamic NURBS with Geometric Constraints for Interactive Sculpting. *ACM Transactions on Graphics*. 1994, 13, s. 103–136.
- TERZOPOULOS, D. – QIN, H. D-NURBS a physics-based framework for geometric design. *IEEE Transactions of Visualization and Computer Graphics*. 1996, 2, 1, s. 85–96.
- TILLER, W. Rational B-splines for Curve and Surface Representation. *IEEE Computer Graphics and Applications*. 1983.
- TILLER, W. – PIEGL, T. A Menagerie of Rational B-Spline Circles. *IEEE Computer Graphics and Applications*. 1989.
- VERMA, V. – KUMAR, R. – HSU, S. 3D Building Detection and Modeling from Aerial LIDAR Data. *IEEE Computer Vision and Pattern Recognition*. 2006, 2, s. 2213–2220.
- VOSSELMAN, V. Building reconstruction using planar faces in very high density data. *International Archives of Photogrammetry and Remote Sensing*. 1999.
- WANG, Y. – ZHENG, J. – SEAH, H. Conversion between T-splines and hierarchical B-splines. *Articles of the Eighth IASTED International Conference COMPUTER GRAPHICS AND IMAGING*. 2005, 8, s. 8–13.
- YOU, S. et al. Urban site modeling from Lidar data. *ICCSA*. 2003, 3, s. 579–88.

ZHENG, J. – WANG, Y. – SEAH, H. S. Adaptive T-spline Surface Fitting to Z-Map Models. *GRAPHITE*. 2006.

ŽÁRA, J. et al. *Moderní počítačová grafika*. Computer Press, 2004.

ČADEK, J. Lineární algebra a geometrie. [online], Naposledy navštíveno 12. 4. 2007.  
Dostupné z: <ftp://www.math.muni.cz/pub/math/people/Cadek/la3/SKRIPTA.pdf>.



# Použité značení

$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$	- množina všech přirozených, celých, racionálních, reálných a komplexních čísel
$\mathbb{R}^n$	- kartézská mocnina množiny $\mathbb{R}$
$T$	- číselné těleso
$\det(A)$	- determinant matice $A$
$\mathbf{u}, \overrightarrow{PQ}, Q - P$	- vektor
$ u ,  \overrightarrow{PQ} $	- velikost vektoru
$V_n$	- vektorový prostor dimenze $n$
$\mathcal{A}_n$	- afinní prostor dimenze $n$
$\mathcal{E}_n$	- euklidovský prostor dimenze $n$
$P_n$	- projektivní prostor dimenze $n$
$P = (p_1, \dots, p_n)$	- bod euklidovského prostoru $\mathcal{E}_n$
$(\mathbf{u}, \mathbf{v})$	- skalární součin vektorů
$\mathbf{u} \otimes \mathbf{v}$	- tenzorový součin vektorů
$\mathbf{U} \otimes \mathbf{V}$	- tenzorový součin vektorových prostorů
$N_i^n(t)$	- B-spline funkce stupně $n$
$C(t)$	- křivka v rovině či prostoru
$S(u, v), T(u, v)$	- vyjádření plochy