



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

# APROXIMACE PLOCH POMOCÍ B-SPLINE A T-SPLINE

SURFACE APPROXIMATION USING B-SPLINE AND T-SPLINE

## BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

## AUTOR PRÁCE

AUTHOR

Tereza Pálková

## VEDOUCÍ PRÁCE

SUPERVISOR

Mgr. Jana Procházková, Ph.D.

BRNO 2023



# Zadání bakalářské práce

Ústav:	Ústav matematiky
Studentka:	<b>Tereza Pálková</b>
Studijní program:	Matematické inženýrství
Studijní obor:	bez specializace
Vedoucí práce:	<b>Mgr. Jana Procházková, Ph.D.</b>
Akademický rok:	2022/23

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

## Aproximace ploch pomocí B–spline a T–spline

### Stručná charakteristika problematiky úkolu:

V rámci bakalářské práce se student seznámí se základní teorií týkající se B–spline reprezentace ploch a jejího zobecnění T–spline. B–spline plochy pracují s pravidelnou mřížkou řídicích bodů a uzlovými vektory, které ji charakterizují. Mají výborné vlastnosti – lokální kontrolovatelnost, možnost přidání vah. T–spline plochy jsou definovány nad nepravidelnou sítí řídicích bodů a je dána pouze lokální reprezentace uzlových vektorů. Rozšiřují možnosti použití aproximačních ploch i pro neuspořádaná data získaná například 3D skenováním.

Student bude mít možnost vyzkoušet si práci s různými nástroji pro 3D skenování (mobil, tablet, 3D skener) a poté vlastní modely použít pro testování a porovnání obou metod aproximace.

### Cíle bakalářské práce:

1. Nastudovat problematiku B–spline aproximačních ploch.
2. Seznámit se s rozšířením B–spline ploch na T–spline, popsat jejich vlastnosti a algoritmy.
3. Vytvořit program na proložení dat pomocí B–spline a T–spline ploch.
4. Otestovat a srovnat možnosti použití na reálných datech získaných 3D skenováním.

### Seznam doporučené literatury:

PIEGL, L. A., TILLER, W. The NURBS book. Berlin: Springer-Verlag, c1995. ISBN 3-540-55069-0.

SEDERBERG, T. W., ZHENG, J., BAKENOV, A., NASRI, A. T-splines and T-NURCCs. ACM Transactions on Graphics. 2003, 22(3), 477-484. ISSN 0730-0301. Dostupné z: doi:10.1145/882262.882295

SEDERBERG, T. W., CARDON, D. L., FINNIGAN, G. T., NORTH, N. S., ZHENG, J., LYCHE, T. T-spline Simplification and Local Refinement. ACM Transactions on Graphics. 2004, 23(3), 276-283. ISSN 0730-0301. Dostupné z: doi:10.1145/1015706.1015715

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2022/23

V Brně, dne

L. S.

---

prof. RNDr. Josef Šlapal, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **Abstrakt**

Bakalářská práce se zaměřuje na B-splajn reprezentaci ploch a její zobecnění T-splajn. Práce je členěna do tří hlavních částí. V úvodu pokládá teoretické základy obou ploch a uvádí jejich matematické vlastnosti. Druhá část se zabývá implementací algoritmů v softwaru Matlab. Poslední část obsahuje aplikaci algoritmů na reálná data získaná 3D skenováním a srovnání obou ploch z hlediska využití.

## **Abstract**

The bachelor thesis focuses on B-spline representation of surfaces and its generalization T-spline. The work is divided into three main parts. The introduction lays down the theoretical foundations of both surfaces and presents their mathematical properties. The second part deals with the implementation of algorithms in Matlab software. The final part contains the application of the algorithms on real data obtained by 3D scanning and a comparison of both surfaces in terms of utilization.

## **klíčová slova**

B-splajn křivka, B-splajn plocha, T-splajn plocha

## **keywords**

B-spline curve, B-spline surface, T-spline surface

PÁLKOVÁ, Tereza. *Aproximace ploch pomocí B-spline a T-spline* [online]. Brno, 2023 [cit. 2023-04-16]. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/145790>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav matematiky. Vedoucí práce Jana Procházková.



Prohlašuji, že jsem bakalářskou práci *Aproximace ploch pomocí B-spline a T-spline* vypracovala samostatně pod vedením Mgr. Jany Procházkové, Ph.D. s použitím materiálů uvedených v seznamu literatury.

Tereza Pálková





Děkuji především své vedoucí Mgr. Janě Procházkové, Ph.D. za cenné rady, čas a trpělivost, se kterou mi pomohla všechno pochopit a zpracovat. Dále bych ráda poděkovala své rodině a přátelům, kteří mi byli důležitou oporou po dobu tvorby této práce.

Tereza Pálková



# Obsah

<b>1 Úvod</b>	<b>2</b>
1.1 Bézierovy křivky a plochy . . . . .	2
1.2 Coonsovy kubiky . . . . .	4
1.3 B-Splajny . . . . .	5
1.3.1 B-splajn funkce . . . . .	5
1.3.2 B-splajn křivky . . . . .	9
1.3.3 B-splajn plochy . . . . .	10
1.4 T-Splajny . . . . .	13
1.4.1 PB-Splajn plochy . . . . .	13
1.4.2 T-Mesh . . . . .	13
1.4.3 Výpočet bázových funkcí . . . . .	14
1.4.4 T-splajn plochy . . . . .	16
<b>2 Programová implementace</b>	<b>18</b>
2.1 Algoritmus pro výpočet B-splajn křivky . . . . .	18
2.2 Algoritmus pro výpočet B-splajn plochy . . . . .	19
2.3 Rychlost a přesnost algoritmu pro výpočet B-splajn ploch . . . . .	21
2.4 Algoritmus pro výpočet T-splajn plochy . . . . .	22
2.5 Rychlost a přesnost algoritmu pro výpočet T-splajn ploch . . . . .	24
<b>3 Aplikace na reálná data</b>	<b>25</b>
3.1 B-splajny . . . . .	25
3.1.1 Získání a úprava dat . . . . .	25
3.1.2 Uspořádání bodů do mřížky . . . . .	26
3.1.3 Aplikace deBoorova algoritmu . . . . .	27
3.2 T-splajny . . . . .	28
3.3 Využití B-splajn a T-splajn ploch a jejich srovnání . . . . .	29
<b>4 Závěr</b>	<b>31</b>

# 1 Úvod

Na přelomu padesátých a šedesátých let 20. století začali matematici věnovat pozornost právě kubickým polynomiálním křivkám. Požadavky přicházely především od designérů z automobilového průmyslu, ale také od návrhářů trupů lodí a letadel. Konkrétně firmy Citroen, Renault a Boeing stály u zrodu nové matematické disciplíny. V jejich konstruktérských a návrhářských kancelářích se do té doby využíval pro návrh konstrukcí složitějších tvarů (například karoserií aut) tzv. splajn. Jednalo se o tenký a dlouhý pásek kovu, který byl tvarován pomocí olověných závaží. Šlo o promyšlený postup, který byl velmi rychlý a názorný. Neposkytoval ale žádný další popis navrženého tvaru, což působilo značné problémy. Pro vyjádření navržených tvarů lodních a letadlových trupů a karoserií tehdy moderních aut už totiž nestačily základní poznatky geometrie, které jim byly známy.

Do problematiky tedy vstoupila matematika se snahou o popis obecných objektů. Cílem bylo jednoduše matematicky popsat navržené tvary a docílit jednoduché manipulace při jejich úpravách. K tomu posloužily právě kubické polynomiální křivky, které navíc nejlépe kopírovaly tvary vytvořené kovovým ohebným pravítkem (splajnem). I když se na kubických polynomech všichni shodli, vzniklo téměř nezávisle na sobě několik přístupů k dané problematice. Všechny využívají popis pomocí tzv. řídicího polygonu, který je složen z řídicích bodů křivky. Tím je zaručena jednoduchá upravitelnost křivek pomocí manipulace s těmito body. Mezi nejznámější předchůdce B-splajnů patří Fergusonovy a Bézierovy křivky a plochy, které prochází prvním a posledním řídicím bodem. Liší se pouze koeficienty polynomu a způsobem zadání tečného vektoru. Dále jsou to Coonsovy kubiky, pomocí nichž skládáme tzv. Coonsovy kubické B-splajny.

V roce 2003 se v matematickém světě objevily T-splajn plochy, které nahadily B-splajny tam, kde nestačily, tedy na nepravidelných mřížkách bodů.

## 1.1 Bézierovy křivky a plochy

Tyto křivky popsali nezávisle na sobě dva matematici. V letech 1959-1962 Pierr Bézier pracující pro firmu Renault na vývoji programu pro návrh křivek a ploch s názvem UNISURF a Paul de Casteljaou ve stejném roce 1959 pro Citroen. Druhý zmíněný si však svou práci nedal patentovat, proto se křivky nazývají Bézierovy. Tato část čerpá z [1] a [3].

Bézierova křivka třetího stupně, neboli Bézierova kubika (obrázek 1), je nejvyužívanější v počítačové grafice. Proto budeme dále v práci pracovat převážně s ní.

Bézierovu kubiku můžeme parametricky vyjádřit rovnicí:

$$Q(t) = \sum_{i=0}^3 P_i B_i(t), t \in \langle 0, 1 \rangle, \quad (1.1)$$

kde  $P_i(t)$  jsou 4 zadané řídicí body a  $B_i(t)$  jsou kubické bázové polynomy nazývané Bernsteinovy polynomy ve tvaru:

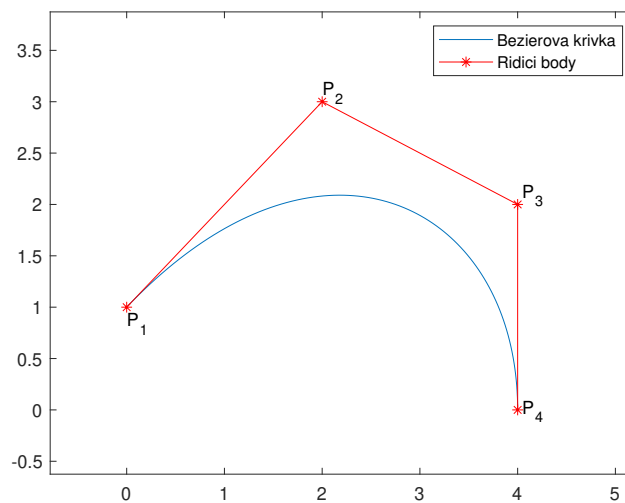
$$\begin{aligned} B_0(t) &= (1-t)^3 \\ B_1(t) &= 3t(1-t)^2 \\ B_2(t) &= 3t^2(1-t) \\ B_3(t) &= t^3 \end{aligned} \quad (1.2)$$

Jelikož jsou Bernsteinovy polynomy ve tvaru binomického rozvoje, je snadné Beziérovu kubiku zobecnit na stupeň  $n$  a její tvar se téměř nezmění:

$$Q(t) = \sum_{k=0}^n P_k B_k(t), t \in \langle 0, 1 \rangle, \quad (1.3)$$

Pro  $i$ -tý Bernsteinův polynom stupně  $n$  potom platí:

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \text{ pro } i = 0, 1, \dots, n. \quad (1.4)$$

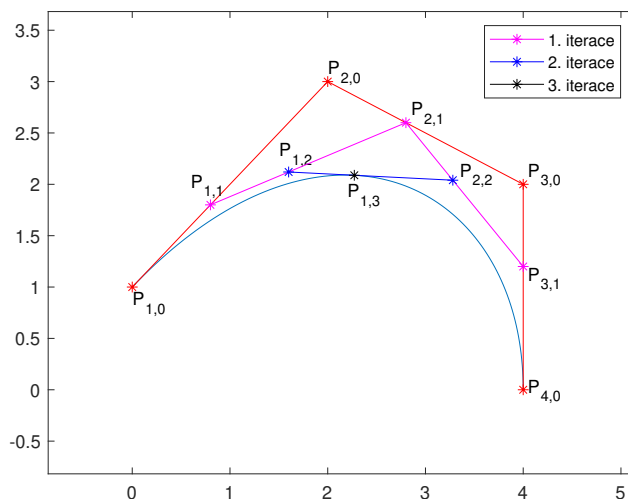


Obrázek 1: Beziérová křivka

Pro získávání bodů na Béziově křivce slouží rekurzivní algoritmus De Casteljau, pro který musíme znát pouze řídicí polygon a hodnotu parametru  $t$ , pro který budeme bod generovat. Algoritmus lze zapsat:

$$P_i^k = (1-t)P_{i-1}^{k-1} + tP_i^{k-1} \quad \text{kde } i = 0, \dots, n-k; \quad k = 0, \dots, n, \quad (1.5)$$

kde  $n$  je počet řídicích bodů. Výsledný bod na křivce je po  $k+1$  iteracích bod  $P_i^3$ . Na obrázku 2 je pomocí algoritmu De Castiljau získán bod pro parametr  $t = 0.4$ .



Obrázek 2: Algoritmus De Casteljau

## 1.2 Coonsovy kubiky

Pojem Coonsovy kubiky byl poprvé představen profesorem Stevenem Ansonem Coonsem, průkopníkem počítačové grafiky z MIT, v roce 1967. Více např. v [2] a [3].

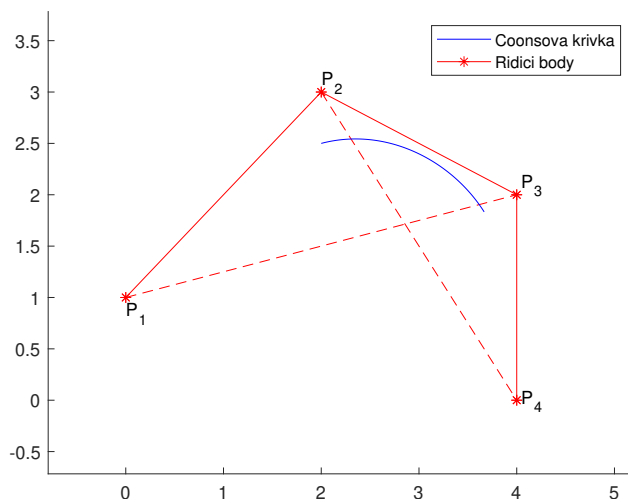
Coonsova kubika je zadána čtyřmi řídicími body  $P_1$ ,  $P_2$ ,  $P_3$  a  $P_4$ . Tyto body budeme vnímat jako dva trojúhelníky  $P_1, P_2, P_3$  a  $P_2, P_3, P_4$ . Křivka potom začíná v tzv. antitěžišti náležícím bodu  $P_2$  v prvním trojúhelníku a končí v antitěžišti náležícím bodu  $P_3$  v druhého trojúhelníku (viz obrázek 3).

Coonsovu kubiku můžeme parametricky vyjádřit rovnicí:

$$Q(t) = \frac{1}{6} \sum_{k=0}^3 P_k C_k(t), t \in \langle 0, 1 \rangle, \quad (1.6)$$

kde  $C_i(t)$  jsou kubické bázové polynomy Coonsových křivek ve tvaru:

$$\begin{aligned} C_0(t) &= (1-t)^3 \\ C_1(t) &= 3t^3 - 6t^2 + 4 \\ C_2(t) &= -3t^3 + 3t^2 + 3t + 1 \\ C_3(t) &= t^3 \end{aligned} \quad (1.7)$$



Obrázek 3: Coonsova křivka

### 1.3 B-Splajny

Splajn je po částech definovaná polynomální funkce se spojitou derivací do řádu křivky. Její hlavní výhodou je lokální kontrolovatelnost (tedy když změníme jeden bod, nedojde ke změně celého tvaru) a hladkost. V praxi jsou často využívány kubické splajny, které nejlépe vystihují tvar pružného kovového pravítka, jak bylo uvedeno v úvodu. Ke konstrukci splajnů se dají využít obě dvě výše uvedené kubiky. Nejpoužívanější je však kubika Coonsova, pomocí které tvoříme tzv. Coonsovy kubické B-splajny.

U Bézierových křivek musíme pro větší přesnost požadovaného tvaru přidávat řídicí body a to často vede ke zvýšení stupně polynomu a práce s nimi je potom velmi obtížná. Jelikož jsou B-splajn křivky složeny z několika kubických křivek, dokážeme s nimi zkonstruovat jakýkoliv tvar s poměrně velkou přesností za použití polynomů třetího stupně. Následující kapitola čerpá z [4, 5, 6].

#### 1.3.1 B-splajn funkce

Bernsteinovy polynomy, zmiňované u Bézierových kubik, slouží v podstatě jako váhy, které jsou jednotlivým bodům přidělovány. Bázové funkce B-splajn křivek fungují stejně, mají však dvě důležité vlastnosti:

1. Jejich definiční obor je rozdělen na podintervaly, tzv. uzly.
2. Bázové funkce jsou nenulové na celém intervalu.

**Definice 1.1.** Nechť  $U = \{u_0, \dots, u_m\}$  je neklesající posloupnost reálných čísel, kde  $u_i \leq u_{i+1}$ ,  $i = 0, \dots, m - 1$ . Hodnoty  $u_i$  nazýváme **uzly** a vektor  $U$  je **uzlový vektor**.

Rekurentně zadaná bázová funkce B-splajn křivek stupně  $p$  se značí  $N_{i,p}(u)$  a je definována jako

$$N_{i,0}(u) = \begin{cases} 1, & \text{pokud } u_i \leq u < u_{i+1} \\ 0, & \text{jinak} \end{cases} \quad (1.8)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (1.9)$$

Jedná se o tzv. *Cox-de Boorovu rekurentní formuli*.

*Poznámka.* Výraz  $\frac{0}{0}$ , kterého můžeme při výpočtu dosáhnout, definujeme jako 0.

*Poznámka.* Nejčastěji budeme pracovat s tzv. *uniformními* uzlovými vektory, kde  $u_0 = 0$ ,  $u_m = 1$  a ostatní uzly jsou mezi nimi ekvidistantně rozloženy.

**Příklad 1.2.** Mějme uzlový vektor  $U = \{0, 0, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1, 1, 1\}$ . Nyní si určíme B-splajn báze funkce stupně 0, 1 a 2. (Opakování krajních hodnot zadaného uzlového vektoru vysvětlíme v definici 1.4.)

### Bázové funkce nultého stupně

$$N_{0,0} = N_{1,0} = 0$$

$$N_{2,0} = \begin{cases} 1, & \text{pro } u \in \langle 0, \frac{1}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

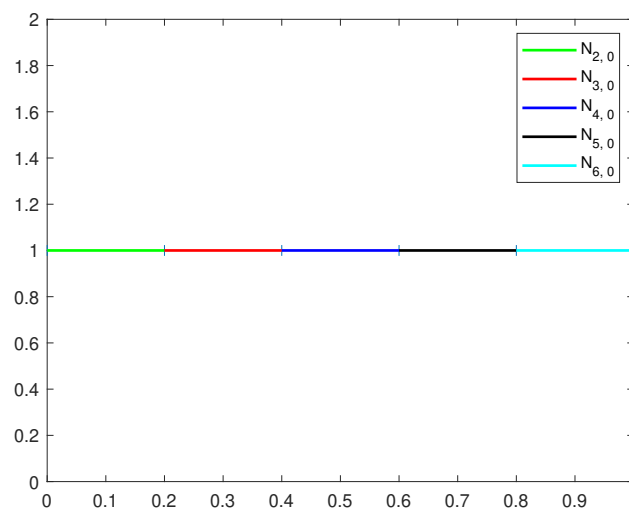
$$N_{3,0} = \begin{cases} 1, & \text{pro } u \in \langle \frac{1}{5}, \frac{2}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{4,0} = \begin{cases} 1, & \text{pro } u \in \langle \frac{2}{5}, \frac{3}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{5,0} = \begin{cases} 1, & \text{pro } u \in \langle \frac{3}{5}, \frac{4}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{6,0} = \begin{cases} 1, & \text{pro } u \in \langle \frac{4}{5}, 1 \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{7,0} = N_{8,0} = 0$$



Obrázek 4: Bázové funkce nultého stupně



## Bázové funkce prvního stupně

$$N_{0,1}(u) = \frac{u-0}{0-0}N_{0,0}(u) + \frac{0-u}{0}N_{1,0}(u) = 0$$

$$N_{1,1}(u) = \frac{u-0}{0-0}N_{1,0}(u) + \frac{0.2-u}{0.2-0}N_{2,0}(u) = \begin{cases} 1-5u, & \text{pro } u \in \langle 0, \frac{1}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{2,1}(u) = \frac{u-0}{0.2-0}N_{2,0}(u) + \frac{0.4-u}{0.4-0.2}N_{3,0}(u) = \begin{cases} 5u, & \text{pro } u \in \langle 0, \frac{1}{5} \rangle \\ 2-5u, & \text{pro } u \in \langle \frac{1}{5}, \frac{2}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

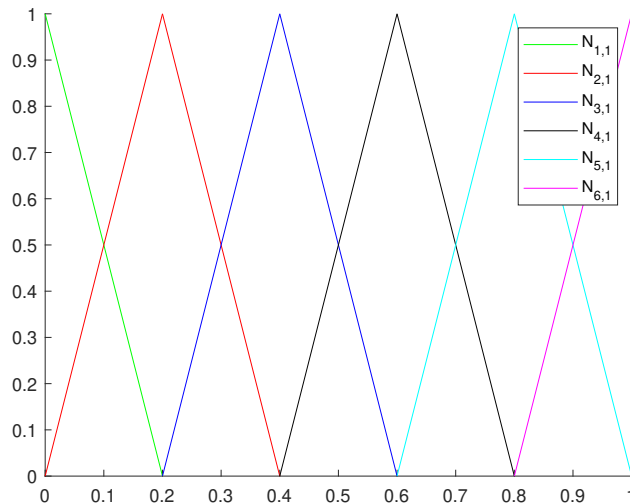
$$N_{3,1}(u) = \frac{u-0.2}{0.4-0.2}N_{3,0}(u) + \frac{0.6-u}{0.6-0.4}N_{4,0}(u) = \begin{cases} 5u-1, & \text{pro } u \in \langle \frac{1}{5}, \frac{2}{5} \rangle \\ 3-5u, & \text{pro } u \in \langle \frac{2}{5}, \frac{3}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{4,1}(u) = \frac{u-0.4}{0.6-0.4}N_{4,0}(u) + \frac{0.8-u}{0.8-0.6}N_{5,0}(u) = \begin{cases} 5u-2, & \text{pro } u \in \langle \frac{2}{5}, \frac{3}{5} \rangle \\ 4-5u, & \text{pro } u \in \langle \frac{3}{5}, \frac{4}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{5,1}(u) = \frac{u-0.6}{0.8-0.6}N_{5,0}(u) + \frac{1-u}{1-0.8}N_{6,0}(u) = \begin{cases} 5u-3, & \text{pro } u \in \langle \frac{3}{5}, \frac{4}{5} \rangle \\ 5-5u, & \text{pro } u \in \langle \frac{4}{5}, 1 \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{6,1}(u) = \frac{u-0.8}{1-0.8}N_{7,0}(u) + \frac{1-u}{1-1}N_{8,0}(u) = \begin{cases} 5u-4, & \text{pro } u \in \langle \frac{4}{5}, 1 \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{7,1}(u) = N_{8,1}(u) = 0$$



Obrázek 5: Bázové funkce prvního stupně

### Bázové funkce druhého stupně

$$N_{0,2}(u) = \frac{u-0}{0-0}N_{0,1}(u) + \frac{0.2-u}{0.2-0}N_{1,1}(u) = \begin{cases} (1-5u)^2, & \text{pro } u \in \langle 0, \frac{1}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{1,2}(u) = \frac{u-0}{0.2-0}N_{1,1}(u) + \frac{0.4-u}{0.4-0}N_{2,1}(u) = \begin{cases} 10u - 37.5u^2, & \text{pro } u \in \langle 0, \frac{1}{5} \rangle \\ 2 - 10u + 12.5u^2, & \text{pro } u \in \langle \frac{1}{5}, \frac{2}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

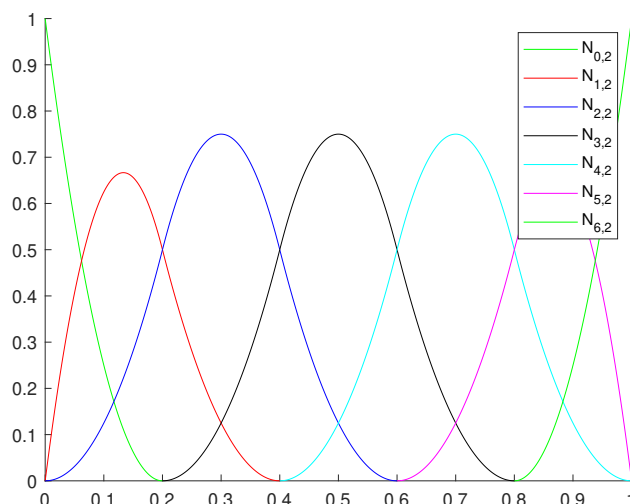
$$N_{2,2}(u) = \frac{u-0}{0.4-0}N_{3,1}(u) + \frac{0.6-u}{0.6-0.2}N_{4,2}(u) = \begin{cases} 12.5u^2, & \text{pro } u \in \langle 0, \frac{1}{5} \rangle \\ 15u - 25u^2 - 1.5, & \text{pro } u \in \langle \frac{1}{5}, \frac{2}{5} \rangle \\ 4.5 - 15u + 12.5u^2, & \text{pro } u \in \langle \frac{2}{5}, \frac{3}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{3,2}(u) = \frac{u-0.2}{0.6-0.2}N_{3,1}(u) + \frac{0.8-u}{0.8-0.4}N_{4,1}(u) = \begin{cases} 12.5u^2 - 5u + 0.5, & \text{pro } u \in \langle \frac{1}{5}, \frac{2}{5} \rangle \\ 25u - 25u^2 - 5.5, & \text{pro } u \in \langle \frac{2}{5}, \frac{3}{5} \rangle \\ 8 - 20u + 12.5u^2, & \text{pro } u \in \langle \frac{3}{5}, \frac{4}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{4,2}(u) = \frac{u-0.4}{0.8-0.4}N_{4,1}(u) + \frac{1-u}{1-0.6}N_{5,1}(u) = \begin{cases} 12.5u^2 - 10u + 2, & \text{pro } u \in \langle \frac{2}{5}, \frac{3}{5} \rangle \\ 35u - 25u^2 - 11.5, & \text{pro } u \in \langle \frac{3}{5}, \frac{4}{5} \rangle \\ 12.5u^2 - 25u + 12.5, & \text{pro } u \in \langle \frac{4}{5}, 1 \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{5,2}(u) = \frac{u-0.6}{1-0.6}N_{5,1}(u) + \frac{1-u}{1-0.8}N_{6,1}(u) = \begin{cases} 12.5u^2 - 15u + 4.5, & \text{pro } u \in \langle \frac{3}{5}, \frac{4}{5} \rangle \\ 50u - 37.5u^2 - 27.5, & \text{pro } u \in \langle \frac{4}{5}, 1 \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{6,2}(u) = \frac{u-0.8}{1-0.8}N_{6,1}(u) + \frac{1-u}{1-1}N_{7,1}(u) = \begin{cases} 25u^2 - 40u + 16, & \text{pro } u \in \langle \frac{4}{5}, 1 \rangle \\ 0, & \text{jinak} \end{cases}$$



Obrázek 6: Bázové funkce druhého stupně

### Vlastnosti bázových funkcí

1.  $N_{i,p}(u) = 0$  pro  $u \notin \langle u_i, u_{i+p+1} \rangle$ .
2. V každém intervalu  $\langle u_j, u_{j+1} \rangle$  je nejvýše  $p + 1$  bázových funkcí nenulových. Konkrétně se jedná o funkce  $N_{j-p,p}, \dots, N_{j,p}$ .
3.  $N_{i,p}(u) \geq 0$  pro každé  $i, p$  a  $u$  (nezápornost).
4. Pro libovolný interval  $\langle u_i, u_{i+1} \rangle$  platí, že  $\sum_{j=i-p}^i N_{j,p}(u) = 1$  pro všechna  $u \in \langle u_i, u_{i+1} \rangle$ .
5. Uvnitř uzlového intervalu existují všechny derivace funkce  $N_{i,p}(u)$ .
6. Funkce  $N_{i,p}(u)$  dosahuje právě jedné maximální hodnoty (kromě případu  $p = 0$ ).

### 1.3.2 B-splajn křivky

**Definice 1.3.** B-splajn křivku stupně  $p$  definujeme jako

$$\mathbf{C}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i, \quad (1.10)$$

kde  $\mathbf{P}_i$  jsou řídicí body a  $N_{i,p}(u)$  jsou B-splajn bázové funkce stupně  $p$  definované na uzlovém vektoru

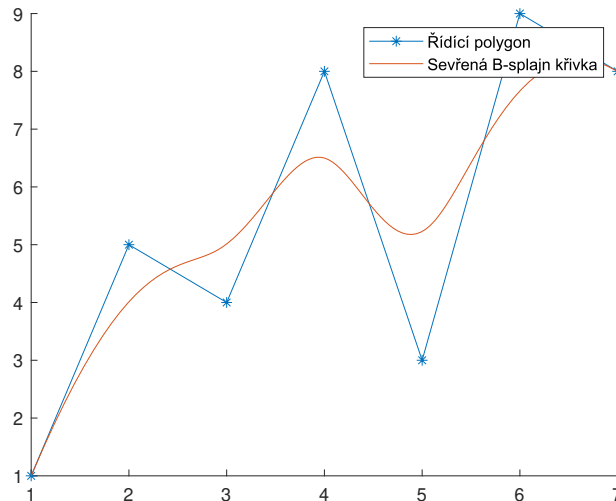
$$U = \{u_0, \dots, u_m\}, \quad (1.11)$$

který obsahuje  $(m+1)$  uzlů.

**Definice 1.4.** B-splajn křivku nazveme **sevřenou**, jestliže má první a poslední uzel uzlového vektoru násobnost  $p + 1$ . Platí, že  $\mathbf{C}(u_0) = \mathbf{P}_0$ ,  $\mathbf{C}(u_m) = \mathbf{P}_m$  a spojnice prvních a posledních dvou řídicích bodů jsou zároveň tečnami B-splajn křivky (viz obrázek 7).

*Poznámka.* Dále se budeme zabývat pouze sevřenými křivkami a budeme předpokládat, že  $u_0 = 0$  a  $u_m = 1$ .

*Poznámka.* Opakování vnitřních uzlů je povoleno pouze do hodnoty  $p + 1$ , kde  $p$  je zmiňovaný stupeň křivky. Násobnost snižuje spojitost křivky v bodě a pro větší hodnoty opakování by tedy bod nebyl zahrnut do křivky.



Obrázek 7: Sevřená B-splajn křivka s řídicím polygonem

**Definice 1.5.** Polygon tvořený body  $\mathbf{P}_i$  nazveme **řídicí polygon**.

#### Důležité vlastnosti B-splajn křivek

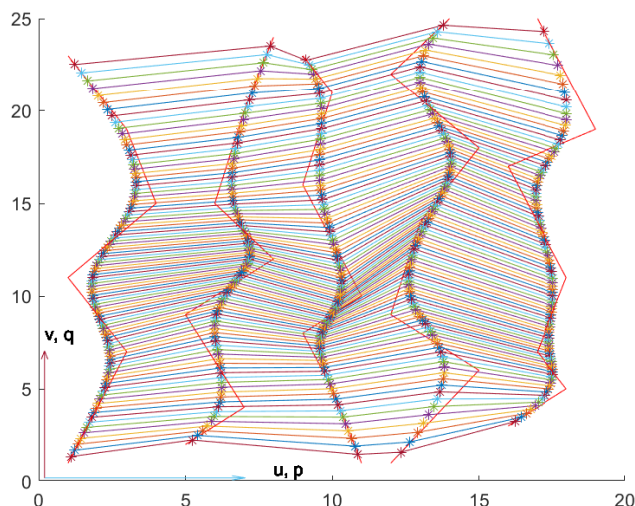
1. Nechť  $\mathbf{C}(u)$  je po částech polynomická křivka stupně  $p$ , která má  $n + 1$  řídicích bodů a  $m + 1$  uzlů. Potom platí vztah:

$$m = n + p + 1$$

2. Křivka prochází prvním a posledním řídicím bodem, tedy  $\mathbf{C}(0) = \mathbf{P}_0$  a  $\mathbf{C}(1) = \mathbf{P}_n$ .
3. Při změně bodu  $\mathbf{P}_i$  dojde ke změně křivky pouze na intervalu  $\langle u_i, u_{i+p+1} \rangle$ . Z této vlastnosti také vyplývá, že  $N_{i,p}(u) = 0$  pro  $u \notin \langle u_i, u_{i+p+1} \rangle$ .
4. Křivka  $\mathbf{C}(u)$  má derivace do řádu  $p$  uvnitř intervalu dvou uzlů a je nejméně  $p - k$  krát diferencovatelná v uzlovém bodě násobnosti  $k$ .

### 1.3.3 B-splajn plochy

Nyní rozšíříme výše uvedené znalosti do prostoru a představíme B-splajn plochu (obrázek 9). Konstrukci provedeme na pravidelné mřížce  $m \times n$  bodů. Algoritmus pro výpočet B-splajn křivky tedy provedeme postupně v obou směrech. Vypočítané body ve směru uzlového vektoru  $v$  převedeme na řídicí (viz obrázek 8) a výpočet opakuje ve směru vektoru  $u$ .



Obrázek 8: Výpočet v prvním směru

**Definice 1.6.** Mějme danu pravidelnou síť  $m \times n$  bodů a dále dva stupně  $p$  a  $q$ . **B-splajn plochu** definujeme vztahem

$$\mathbf{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j}, \quad (1.12)$$

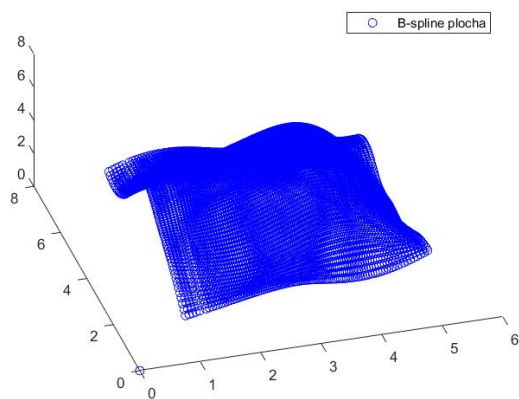
kde  $\mathbf{P}_{i,j}$  jsou řídicí body a  $U, V$  dva uzlové vektory

$$U = \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1}\}, \quad (1.13)$$

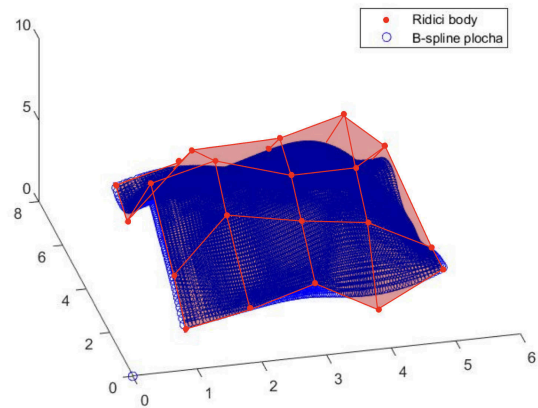
$$V = \{\underbrace{0, \dots, 0}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{1, \dots, 1}_{q+1}\},$$

$U$  má  $(r+1)$  a  $V(s+1)$  uzlů. Dále platí rovnice  $r = n + p + 1$  a  $s = m + q + 1$ .

*Poznámka.* Pro naše účely budou mít křivky v obou směrech uzlových vektorů stejný stupeň, tedy  $p = q$ . Nejčastěji potom budeme využívat stupně tři:  $p = q = 3$  (vysvětlení je podáno v úvodu práce (kapitola 1)).



(a) B-splajn plocha

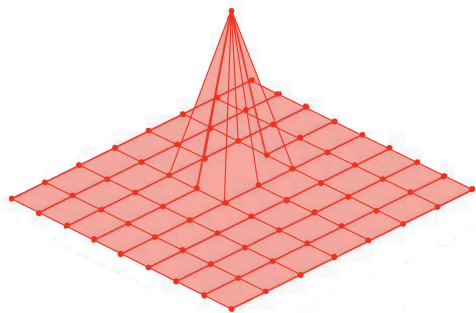


(b) B-splajn plocha s řídicí mřížkou

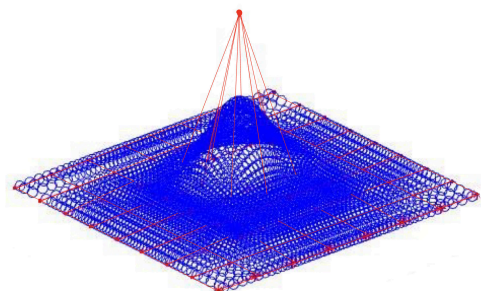
Obrázek 9: B-splajn plocha

### Důležité vlastnosti B-splajn ploch

1. Plocha prochází čtyřmi rohovými řídicími body, tedy  $\mathbf{S}(0, 0) = \mathbf{P}_{0,0}$ ,  $\mathbf{S}(1, 0) = \mathbf{P}_{n,0}$ ,  $\mathbf{S}(0, 1) = \mathbf{P}_{0,m}$  a  $\mathbf{S}(1, 1) = \mathbf{P}_{n,m}$ .
2. Při změně bodu  $\mathbf{P}_{i,j}$  dojde ke změně plochy pouze na obdélníku  $\langle u_i, u_{i+p+1} \rangle \times \langle v_j, v_{j+q+1} \rangle$ . Tuto vlastnost lze sledovat na obrázku 10.
3. Plocha  $\mathbf{S}(u, v)$  je  $p - k$  ( $q - k$ ) krát diferencovatelná ve směru vektoru  $u$  ( $v$ ) v uzlovém bodě násobnosti  $k$ .
4. Řídící síť bodů je po částech rovinná plocha. Proto čím nižší je stupeň B-splajn plochy, tím lepší je aproximací sítě bodů.



(a) Změněná řídicí mřížka



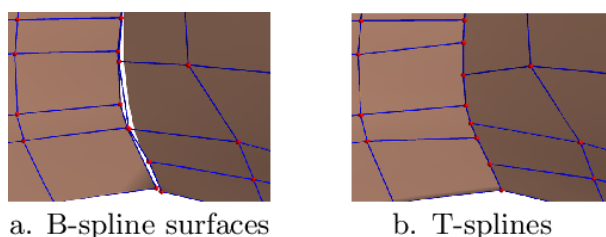
(b) Změněná B-splajn plocha

Obrázek 10: Lokální kontrolovatelnost B-splajn ploch

## 1.4 T-Splajny

T-splajn plochy jsou nástupci B-splajn a NURBS ploch. Představeny byly v roce 2003 v [7] a patentovány 2007. Jde tedy o poměrně moderní technologii, která má proti svým předchůdcům mnoho výhod. Nejdříve si zde uvedeme čtyři základní výhody.

1. Narozdíl od B-splajn ploch nepotřebují T-splajny ke konstrukci pravidelnou síť bodů (Touto vlastností se budeme zabývat v kapitole 1.4.2).
2. V porovnání s NURBS plochami potřebují výrazně menší počet řídicích bodů. Díky tomu je jednodušší také tvarování plochy. Více v [8].
3. Vytváří mnohem plynulejší napojení 2 ploch (viz. obrázek 11).
4. Výhodou je snadné přidávání bodů. Nemusíme totiž doplnit celý sloupec nových bodů, aby byla zachována pravidelná mřížka u B-splajnů.



Obrázek 11: Porovnání napojení B-splajn a T-splajn ploch [7]

Další informace týkající se T-splajn ploch lze nalézt v [7, 9].

### 1.4.1 PB-Splajn plochy

Abychom byli schopni definovat T-splajn plochy matematicky, podívejme se nejdřív na takzvané "Point Based" plochy. Jak z názvu vyplývá, jedná se o plochy, které jsou založeny pouze na jednotlivých bodech, nikoliv na topologickém vztahu mezi nimi (tzv. *mřížce*).

PB-splajn plochu potom můžeme vyjádřit parametricky rovnicí:

$$P(u, v) = \frac{\sum_{i=1}^n P_i B_i(u, v)}{\sum_{i=1}^n B_i(u, v)}, \quad (1.14)$$

kde  $P_i$  jsou řídicí body a  $B_i(u, v)$  jsou bázové funkce definovány jako součin

$$B_i(u, v) = N_{i0}^3(u) N_{i0}^3(v), \quad (1.15)$$

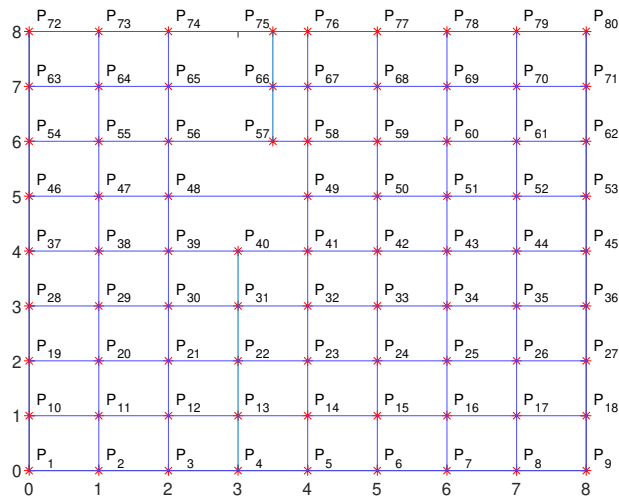
kde  $N_{i0}^3(u)$  je bázová funkce třetího stupně svázaná s vektorem  $u$  a  $N_{i0}^3(v)$  s vektorem  $v$ .

Z definice tedy vyplývá, že k vytvoření PB-splajn plochy musíme znát ke každému řídicímu bodu také 2 uzlové vektory.

### 1.4.2 T-Mesh

Tím se dostáváme k definici T-splajn ploch. Jediné, čím se od zmiňovaných PB-splajnů liší je, že nebudeme mít řídicí body rozmístěny náhodně, ale uspořádaný v řídicí mřížce nazývané *T-mesh* (obrázek 12).

Jak jsme si uvedli v úvodu (část 1.4), má tohle uspořádání mnoho výhod oproti B-splajn plochám. T-mesh například podporuje vznik *T-spojů*, které můžeme na obrázku 13 vidět u bodů  $P_{40}$  a  $P_{57}$

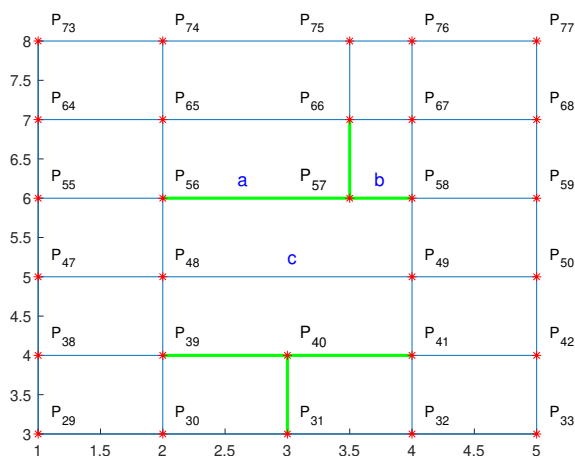


Obrázek 12: T-mesh mřížka

*Poznámka.* Pokud by se v T-mesh nenacházel žádný T-spoj, jednalo by se o mřížku B-splajn ploch.

#### Pravidla pro tvorbu T-mesh:

1. Součet uzlových intervalů na opačných stranách musí být stejný. Například v obrázku 13:  $c = a + b$ .
2. Pokud vzniknou v mřížce naproti sobě dva T-spoje, které mohou být propojeny jednou hranou, musí být tato hrana do sítě zahrnuta.



Obrázek 13: T-mesh mřížka detail

#### 1.4.3 Výpočet báзовých funkcí

Jelikož se v následující části budeme zabývat T-splajn plochami, budeme potřebovat obecné B-splajn báзовé funkce stupně tři, které nyní odvodíme ze vztahů (1.8) a (1.9).

**Příklad 1.7.** Mějme uzlový vektor  $U = \{u_1, u_2, u_3, u_4, u_5\}$ . Nyní určíme B-splajn báзовé funkce stupně 0, 1, 2 a 3.



### Bázové funkce nultého stupně

$$N_{1,0} = \begin{cases} 1, & \text{pro } u \in \langle u_1, u_2 \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{2,0} = \begin{cases} 1, & \text{pro } u \in \langle u_2, u_3 \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{3,0} = \begin{cases} 1, & \text{pro } u \in \langle u_3, u_4 \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{4,0} = \begin{cases} 1, & \text{pro } u \in \langle u_4, u_5 \rangle \\ 0, & \text{jinak} \end{cases}$$

### Bázové funkce prvního stupně

$$N_{1,1}(u) = \frac{u - u_1}{u_2 - u_1} N_{1,0}(u) + \frac{u_3 - u}{u_3 - u_2} N_{2,0}(u) = \begin{cases} \frac{u-u_1}{u_2-u_1}, & \text{pro } u \in \langle u_1, u_2 \rangle \\ \frac{u_3-u}{u_3-u_2}, & \text{pro } u \in \langle u_2, u_3 \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{2,1}(u) = \frac{u - u_2}{u_3 - u_2} N_{2,0}(u) + \frac{u_4 - u}{u_4 - u_3} N_{3,0}(u) = \begin{cases} \frac{u-u-2}{u_3-u_2}, & \text{pro } u \in \langle 0, \frac{1}{5} \rangle \\ \frac{u_4-u}{u_4-u_3}, & \text{pro } u \in \langle \frac{1}{5}, \frac{2}{5} \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{3,1}(u) = \frac{u - u_3}{u_4 - u_3} N_{3,0}(u) + \frac{u_5 - u}{u_5 - u_4} N_{4,0}(u) = \begin{cases} \frac{u-u_3}{u_4-u_3}, & \text{pro } u \in \langle u_3, u_4 \rangle \\ \frac{u_5-u}{u_5-u_4}, & \text{pro } u \in \langle u_4, u_5 \rangle \\ 0, & \text{jinak} \end{cases}$$

### Bázové funkce druhého stupně

$$N_{1,2}(u) = \frac{u - u_1}{u_3 - u_1} N_{1,1}(u) + \frac{u_4 - u}{u_4 - u_2} N_{2,1}(u) = \begin{cases} \frac{(u-u_1)^2}{(u_2-u_1)(u_4-u_1)(u_3-u_1)}, & \text{pro } u \in \langle u_1, u_2 \rangle \\ \frac{u-u_1}{u_3-u_1} \cdot \frac{u_3-u}{u_3-u_2} + \frac{(u_3-u_1)(u_2-u_1)}{u_4-u} \cdot \frac{u-u_2}{u_3-u_2}, & \text{pro } u \in \langle u_2, u_3 \rangle \\ \frac{(u_4-u)^2}{(u_4-u_2)(u_4-u_3)}, & \text{pro } u \in \langle u_3, u_4 \rangle \\ 0, & \text{jinak} \end{cases}$$

$$N_{2,2}(u) = \frac{u - u_2}{u_4 - u_2} N_{2,1}(u) + \frac{u_5 - u}{u_5 - u_3} N_{3,1}(u) = \begin{cases} \frac{(u-u_2)^2}{(u_4-u_2)(u_3-u_2)}, & \text{pro } u \in \langle u_2, u_3 \rangle \\ \frac{u-u_2}{u_4-u_2} \cdot \frac{u_4-u}{u_4-u_3} + \frac{(u_4-u_2)(u_3-u_2)}{u_5-u} \cdot \frac{u-u_3}{u_4-u_3}, & \text{pro } u \in \langle u_3, u_4 \rangle \\ \frac{(u_5-u)^2}{(u_5-u_3)(u_5-u_4)}, & \text{pro } u \in \langle u_4, u_5 \rangle \\ 0, & \text{jinak} \end{cases}$$

### Bázové funkce třetího stupně

$$N_{1,3}(u) = \begin{cases} \frac{(u-u_1)^3}{(u_2-u_1)(u_4-u_1)(u_3-u_1)}, & \text{pro } u \in \langle u_1, u_2 \rangle \\ \frac{(u-u_1)^2(u_3-u)}{(u_3-u_2)(u_4-u_2)(u_3-u_1)} + \frac{(u_4-u)(u-u_1)(u-u_2)}{(u_3-u_2)(u_4-u_2)(u_4-u_1)} + \frac{(u_5-u)(u-u_2)^2}{(u_3-u_2)(u_5-u_2)(u_4-u_2)}, & \text{pro } u \in \langle u_2, u_3 \rangle \\ \frac{(u-u_1)(u_4-u)^2}{(u_4-u_3)(u_4-u_2)(u_4-u_1)} + \frac{(u_5-u)(u_4-u)(u-u_2)}{(u_4-u_3)(u_5-u_2)(u_4-u_2)} + \frac{(u_5-u)^2(u-u_3)^2}{(u_4-u_3)(u_5-u_3)(u_5-u_2)}, & \text{pro } u \in \langle u_3, u_4 \rangle \\ \frac{(u_5-u)^3}{(u_5-u_4)(u_5-u_3)(u_5-u_2)}, & \text{pro } u \in \langle u_4, u_5 \rangle \end{cases}$$

### 1.4.4 T-splajn plochy

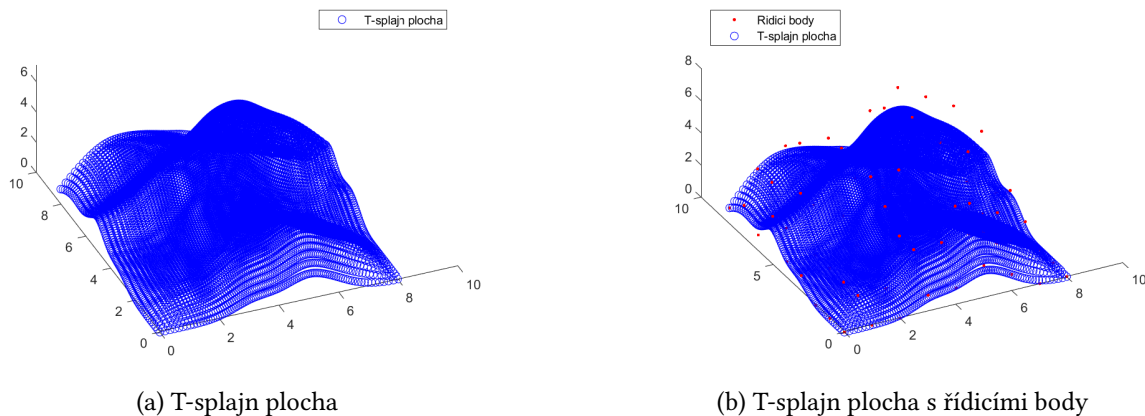
**Definice 1.8.** Mějme dānu sít bodů uspořādu do T-mesh a dále stupeň  $p$ . **T-splajn plochu** (obrázek 14) definujeme vztahem

$$\mathbf{T}(u, v) = \sum_{i=1}^n N_{i,0}^p(u) N_{i,0}^p(v) \mathbf{P}_i, \quad (1.16)$$

kde  $N(u), N(v)$  jsou bázové funkce dāny obecně vztahy:

$$N(t) = \begin{cases} \frac{(t-t_0)^3}{(t_1-t_0)(t_3-t_0)(t_2-t_0)}, & \text{pro } t \in \langle t_0, t_1 \rangle \\ \frac{(t-t_0)^2(t_2-t)}{(t_2-t_1)(t_3-t_1)(t_2-t_0)} + \frac{(t_3-t)(t-t_0)(t-t_1)}{(t_2-t_1)(t_3-t_1)(t_3-t_0)} + \frac{(t_4-t)(t-t_1)^2}{(t_2-t_1)(t_4-t_1)(t_3-t_1)}, & \text{pro } t \in \langle t_1, t_2 \rangle \\ \frac{(t-t_0)(t_3-t)^2}{(t_3-t_2)(t_3-t_1)(t_3-t_0)} + \frac{(t_4-t)(t_3-t)(t-t_1)}{(t_3-t_2)(t_4-t_1)(t_3-t_1)} + \frac{(t_4-t)^2(t-t_2)^2}{(t_3-t_2)(t_4-t_2)(t_4-t_1)}, & \text{pro } t \in \langle t_2, t_3 \rangle \\ \frac{(t_4-t)^3}{(t_4-t_3)(t_4-t_2)(t_4-t_1)}, & \text{pro } t \in \langle t_3, t_4 \rangle \end{cases} \quad (1.17)$$

*Poznámka.* Dále budeme uvažovat stupeň plochy  $p = 3$  ze stejných důvodů, které byly uvedeny u B-splajn ploch.



Obrázek 14: T-splajn plocha

Pro vznik algoritmu na výpočet těchto ploch je pro každý bod nutné znát dva uzlové vektory, ve směru  $u$  a  $v$ . Ty se vygenerují pro každý řádek a sloupec podle počtu bodů a stupně křivky (v našem případě je stupeň roven třem) a přiřadí se jednotlivým bodům podle jejich pořadí. Tabulka na obrázku 15 znázorňuje body a jim odpovídající uzlové vektory podle T-mesh znázorněné na obrázku 12.

	body $P$	$u$	$v$
$P_1$	0 0 0	0 0 0 0 1/6	0 0 0 0 1/6
$P_2$	1 0 0	0 0 0 1/6 2/6	0 0 0 0 1/6
...	...	...	...
$P_9$	8 0 0	5/6 1 1 1 1	0 0 0 0 1/6
$P_{10}$	0 1 0	0 0 0 0 1/6	0 0 0 1/6 2/6
$P_{11}$	1 1 0	0 0 0 1/6 2/6	0 0 0 1/6 2/6
...	...	...	...
$P_{18}$	8 1 0	5/6 1 1 1 1	0 0 0 1/6 2/6
$P_{19}$	0 2 0	0 0 0 0 1/6	0 0 1/6 2/6 3/6
$P_{20}$	1 2 0	0 0 0 1/6 2/6	0 0 1/6 2/6 3/6
...	...	...	...
$P_{27}$	8 2 0	5/6 1 1 1 1	0 0 1/6 2/6 3/6
$P_{28}$	0 3 0	0 0 0 0 1/6	0 1/6 2/6 3/6 4/6
$P_{29}$	1 3 0	0 0 0 1/6 2/6	0 1/6 2/6 3/6 4/6
...	...	...	...
$P_{36}$	8 3 0	5/6 1 1 1 1	0 1/6 2/6 3/6 4/6
$P_{37}$	0 4 0	0 0 0 0 1/6	1/6 2/6 3/6 4/6 5/6
$P_{38}$	1 4 0	0 0 0 1/6 2/6	1/6 2/6 3/6 4/6 5/6
...	...	...	...
$P_{45}$	8 4 0	5/6 1 1 1 1	1/6 2/6 3/6 4/6 5/6
$P_{46}$	0 5 0	0 0 0 0 1/6	2/6 3/6 4/6 5/6 1
$P_{47}$	1 5 0	0 0 0 1/6 3/6	2/6 3/6 4/6 5/6 1
$P_{48}$	2 5 0	0 0 1/6 3/6 4/6	2/6 3/6 4/6 5/6 1
$P_{49}$	4 5 0	0 1/6 3/6 4/6 5/6	2/6 3/6 4/6 5/6 1
$P_{50}$	5 5 0	1/6 3/6 4/6 5/6 1	2/6 3/6 4/6 5/6 1
...	...	...	...
$P_{53}$	8 5 0	5/6 1 1 1 1	2/6 3/6 4/6 5/6 1
$P_{54}$	0 6 0	0 0 0 0 1/6	3/6 4/6 5/6 1 1
$P_{55}$	1 6 0	0 0 0 1/6 5/12	3/6 4/6 5/6 1 1
$P_{56}$	2 6 0	0 0 1/6 5/12 6/12	3/6 4/6 5/6 1 1
$P_{57}$	3.5 6 0	0 1/6 5/12 6/12 4/6	2/6 3/6 5/6 1 1
$P_{58}$	4 6 0	1/6 5/12 6/12 4/6 5/6	3/6 4/6 5/6 1 1
...	...	...	...
$P_{62}$	8 6 0	5/6 1 1 1 1	3/6 4/6 5/6 1 1
$P_{63}$	0 7 0	0 0 0 0 1/6	4/6 5/6 1 1 1
$P_{64}$	1 7 0	0 0 0 1/6 5/12	4/6 5/6 1 1 1
$P_{65}$	2 7 0	0 0 1/6 5/12 6/12	4/6 5/6 1 1 1
$P_{66}$	3.5 7 0	0 1/6 5/12 6/12 4/6	3/6 5/6 1 1 1
$P_{67}$	4 7 0	1/6 5/12 6/12 4/6 5/6	4/6 5/6 1 1 1
...	...	...	...
$P_{71}$	8 7 0	5/6 1 1 1 1	4/6 5/6 1 1 1
$P_{72}$	0 8 0	0 0 0 0 1/6	5/6 1 1 1 1
$P_{73}$	1 8 0	0 0 0 1/6 5/12	5/6 1 1 1 1
$P_{74}$	2 8 0	0 0 1/6 5/12 6/12	5/6 1 1 1 1
$P_{75}$	3.5 8 0	0 1/6 5/12 6/12 4/6	5/6 1 1 1 1
$P_{76}$	4 8 0	1/6 5/12 6/12 4/6 5/6	5/6 1 1 1 1
...	...	...	...
$P_{80}$	8 8 0	5/6 1 1 1 1	5/6 1 1 1 1

Obrázek 15: Body a uzlové vektory

## 2 Programová implementace

### 2.1 Algoritmus pro výpočet B-splajn křivky

Pro výpočet B-splajn křivky se využívá tzv. **deBoorův algoritmus**. Pro každý parametr  $u$  určí bod na B-splajn křivce.

Funkce `vratindex` vrátí násobnost  $s$  parametru  $u$  v uzlovém vektoru (pokud parametr neodpovídá žádnému uzlu,  $s = 0$ ). Dále  $h$ , což udává, kolikrát je třeba parametr do uzlového vektoru vložit, aby jeho násobnost odpovídala stupni křivky, tedy  $h = \text{stupen} - s$ . Nakonec vrátí hodnotu  $l$ , což je informace a o tom, v jaké části uzlového vektoru se parametr nachází.

Dále dojde k výpočtu bazové funkce, kterou v algoritmu značíme  $a$ , a váženého průměru bodů. Tím získáme nový bod na křivce. První `for` cyklus provádí počet iterací na základě spočítaného  $h$  (hodnoty jsou posunuty kvůli softwaru Matlab, který nepovoluje indexování od nuly.), druhý postupně prochází uzlový vektor.

Hraniční parametry 0 a 1 nejsou v algoritmu ošetřeny a proto nastavíme 0 do počátečního a 1 do posledního řídicího bodu.

Nakonec funkce `vykresleni` přeindexuje body tak, aby bylo možné jejich vykreslení a následně je graficky zobrazí.

#### Vstupy

- Matice řídicích bodů  $P$
- Stupeň křivky (obvykle volíme 3)

#### Výstupy

- Matice bodů B-splajn křivky  $Q$
- Vykreslení křivky (obr. 16)

#### Zdrojový kód 1: DeBoorův algoritmus

```
function [Q]= BsplajnKrivka (P_path , stupen)
    P = importdata (P_path);
    P=P (: ,: ,1);
    [pocetbodu ,~]= size (P);
    uzlovyVektor=uzlvek (pocetbodu , stupen );
    Q=zeros (11 ,3);

    for u=0.01:0.01:0.99
        [s ,h ,l]= vratindex (u ,uzlovyVektor , stupen );

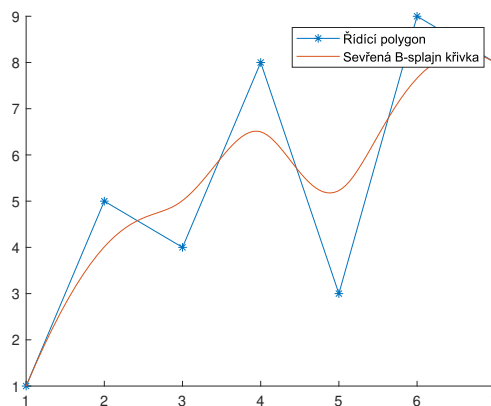
        % vypocet bazovych funkci a bodu na krivce
        for r=2:h+1
            for i=l-stupen+r-1:l-s
                a(i , r)=(u-uzlovyVektor ( i ))
                    / ( uzlovyVektor ( i+stupen-r+2)-uzlovyVektor ( i ));
                P(i , 1 , r)= (1-a(i , r))*P(i-1 ,1 ,r-1)+a(i , r)*P(i ,1 , r-1);
            end
        end
    end
```

```

        P(i, 2, r) = (1 - a(i, r)) * P(i - 1, 2, r - 1) + a(i, r) * P(i, 2, r - 1);
        P(i, 3, r) = (1 - a(i, r)) * P(i - 1, 3, r - 1) + a(i, r) * P(i, 3, r - 1);
    end
end

% ukladani vypoctenych bodu
Q(u*100, :, 1) = P(l - s, :, stupen - s + 1);
% prvni a posledni bod
Q(1, :, 1) = P(1, :, 1);
Q(101, :, 1) = P(7, :, 1);
end
vykresleni(P, Q)
end

```



Obrázek 16: B-splajn křivka

## 2.2 Algoritmus pro výpočet B-splajn plochy

Nyní rozšíříme předchozí případ na pravidelnou mřížku  $m \times n$  bodů. Opět provedeme deBoorův algoritmus, jen nyní určíme body B-splajn křivek nejdříve pro všechny sloupce řídicích bodů ve směru uzlového vektoru `uzlovyVektor2` a tedy pro parametr  $v$ . Získané body nyní využijeme jako nové řídicí body, k čemuž použijeme funkci `PrevedVypocitanBodyNaRidici`, která je správně přeindexuje. Algoritmus nyní provádíme znovu ve směru uzlového vektoru `uzlovyVektor1`. Tím získáme B-splajn plochu, jejíž příklad je na obrázku 17.

### Vstupy

- Matice řídicích bodů  $P$  (počet matic závisí na počtu sloupců řídicích bodů, což odpovídá proměnné `pocetBodu2`)
- Stupeň křivky (obvykle volíme 3 a v obou směrech bude stejný)

### Výstupy

- Vykreslení plochy (obr. 17)

Zdrojový kód 2: DeBoorův algoritmus pro plochu

---

```

function []= BsplajnPlocha (stupen , krok)
% definice promennych
[P, pocetbodu1 , pocetbodu2] = getData ();
uzlovyVektor2=uzlvek (pocetbodu1 , stupen );      %osa y
uzlovyVektor1=uzlvek (pocetbodu2 , stupen );      %osa x

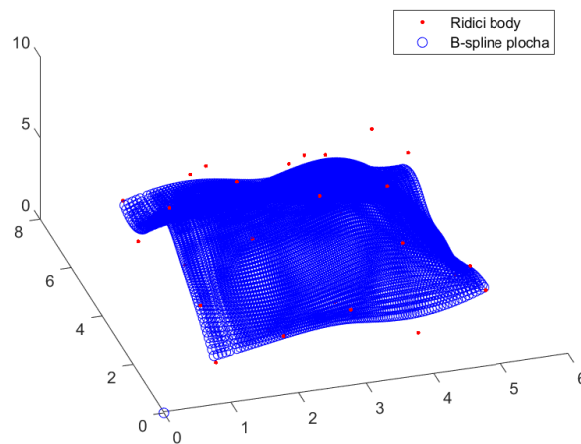
% vypocet poctu iteraci pro smer v
for v=0.01:krok:0.99
    [s1 , h1 , l1 ]= vratindex1 (uzlovyVektor2 , v , stupen );

    for w1=1:pocetbodu2
        for r1=2:h1+1
            for i=l1 -stupen+r1 -1:l1 -s1
                a(i , r1)=(v-uzlovyVektor2(i))
                /(uzlovyVektor2(i+stupen-r1+2)-uzlovyVektor2(i));
                P(i , 1 , r1 , w1)= (1-a(i , r1))*P(i-1 , 1 , r1-1 , w1)+
                    a(i , r1)*P(i , 1 , r1-1 , w1);
                P(i , 2 , r1 , w1)= (1-a(i , r1))*P(i-1 , 2 , r1-1 , w1)+
                    a(i , r1)*P(i , 2 , r1-1 , w1);
                P(i , 3 , r1 , w1)= (1-a(i , r1))*P(i-1 , 3 , r1-1 , w1)+
                    a(i , r1)*P(i , 3 , r1-1 , w1);
            end
        end
    end
    [Q]=PrevedVypocitaneBodyNaRidici(P , l1 , s1 , stupen , pocetbodu2 );
    hold on

%vypocet splajnu pro smer u
for u=0.01:krok:0.99
    [s2 , h2 , l2 ]= vratindex2 (uzlovyVektor1 , u , stupen );
    for r2=2:h2+1
        for i=l2 -stupen+r2 -1:l2 -s2
            a(i , r2)=(u-uzlovyVektor1(i))
            /(uzlovyVektor1(i+stupen-r2+2)-uzlovyVektor1(i));
            Q(i , 1 , r2)= (1-a(i , r2))*Q(i-1 , 1 , r2-1)+
                a(i , r2)*Q(i , 1 , r2-1);
            Q(i , 2 , r2)= (1-a(i , r2))*Q(i-1 , 2 , r2-1)+
                a(i , r2)*Q(i , 2 , r2-1);
            Q(i , 3 , r2)= (1-a(i , r2))*Q(i-1 , 3 , r2-1)+
                a(i , r2)*Q(i , 3 , r2-1);
        end
    end
    vykresli(Q , u , l2 , s2 , pocetbodu2 , stupen)
end
end
end
end

```

---



Obrázek 17: B-splajn plocha

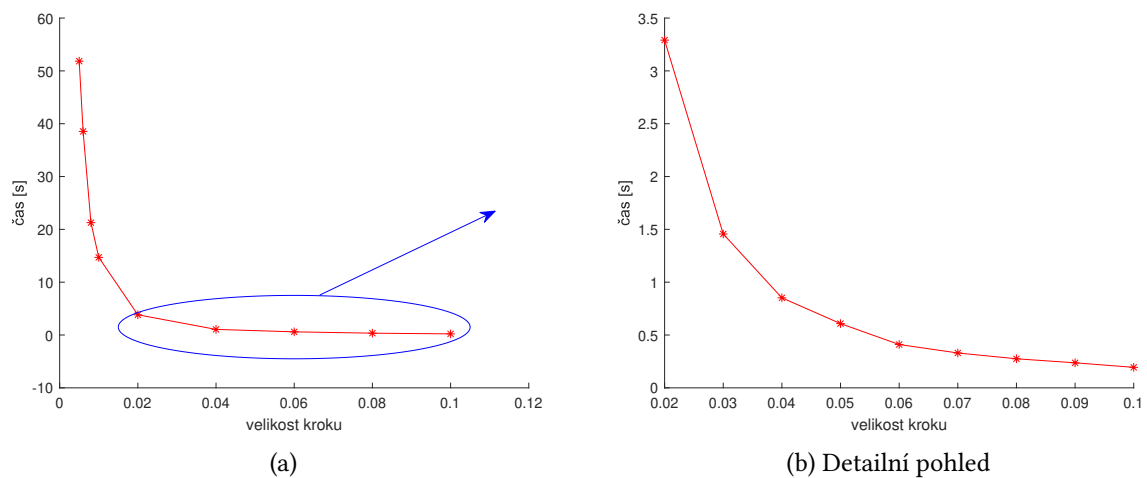
### 2.3 Rychlost a přesnost algoritmu pro výpočet B-splajn ploch

Tabulka 1 popisuje rychlost běhu algoritmu v závislosti na velikosti kroku, tedy na počtu bodů, který musí program vypočítat. Pro menší krok už počítač nebyl schopen algoritmus dokončit v přijatelném čase a tak byl výpočet ukončen.

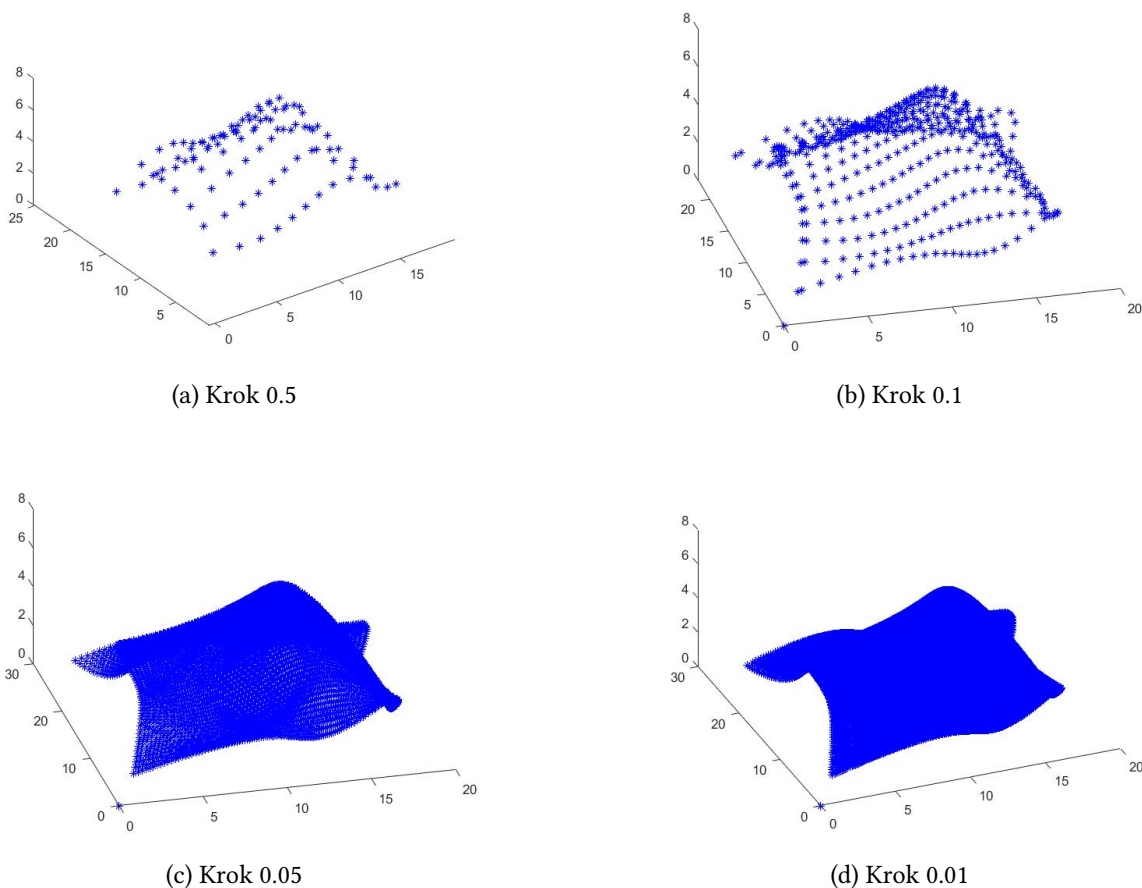
velikost kroku	0.1	0.08	0.06	0.04	0.02	0.01	0.008	0.006	0.005
čas [s]	0.21	0.35	0.60	1.06	3.83	14.69	21.27	38.54	51.84

Tabulka 1: Závislost času na velikosti kroku

Graf 18 znázorňuje závislost času potřebného pro výpočet na velikosti kroku. Průběh připomíná funkci  $e^{1/x}$ , tedy čas klesá exponenciálně. Obrázky 19 ukazují výsledky algoritmu v závislosti na jemnosti kroku.



Obrázek 18: Rychlost programu v závislosti na velikosti kroku



Obrázek 19: Výsledky algoritmu v závislosti na velikosti kroku

## 2.4 Algoritmus pro výpočet T-splajn plochy

Nyní se budeme věnovat algoritmizaci výpočtu T-splajn ploch. Pro každou kombinaci parametrů  $u$  a  $v$  tedy chceme získat bod plochy. Vstupy budou tři matice: Matice bodů  $P$ , matice  $U$  uzlových vektorů ve směru  $u$  a matice  $V$  uzlových vektorů ve směru  $v$ . Jejich pozice si musí vzájemně odpovídat, tedy prvnímu bodu matice  $P$  odpovídá první vektor matice  $U$  a první vektor matice  $V$ .

Funkce `vratIndexy` zjistí, kde v uzlovém vektoru se parametr nachází, tedy pro  $u \in \langle u_{u_1}, u_{u_1+1} \rangle$  vrátí hodnotu  $u_1$ , stejným postupem pro  $v$  vrátí  $v_1$ . Pokud se parametr v uzlovém vektoru vůbec nenachází, funkce vrátí nulu. Výpočet provedeme pro všechny uzlové vektory matic  $U$  a  $V$ ,  $u_1$  a  $v_1$  budou tedy vektory.

Dále přejdeme k výpočtu bázových funkcí. Využijeme vzorec (1.17) a pro každý index  $z$   $u_1$  a  $v_1$  vrátíme hodnotu bázové funkce, pro každý směr samostatně.

Nyní již určíme body T-splajn plochy dány vztahem (1.16) a vše vykreslíme.



## Vstupy

- Matice P řídicích bodů T-mesh
- Matice U uzlových vektorů ve směru u
- Matice V uzlových vektorů ve směru v

## Výstupy

- Vykreslení plochy (obr. 20)

---

### Zdrojový kód 3: Algoritmus pro výpočet T-splajn plochy

---

```
function []=TsplajnPlocha(U_path , V_path , P_path , krok)
    clc , clf
    [U, V, P]= getData(U_path , V_path , P_path );

    % definice promennych
    delkaradku=length(0:krok:max(max(U)));
    delkasloupce=length(0:krok:max(max(V)));
    Q=zeros(delkaradku , delkasloupce ,3);
    pocitadloR =1;
    pocitadloS =1;
    [sizeU ,~]= size (U);
    [sizeV ,~]= size (V);

    % vypocet ve smeru u
    for u= 0:krok:max(max(U))

        % vypocet ve smeru v
        for v=0:krok:max(max(V))

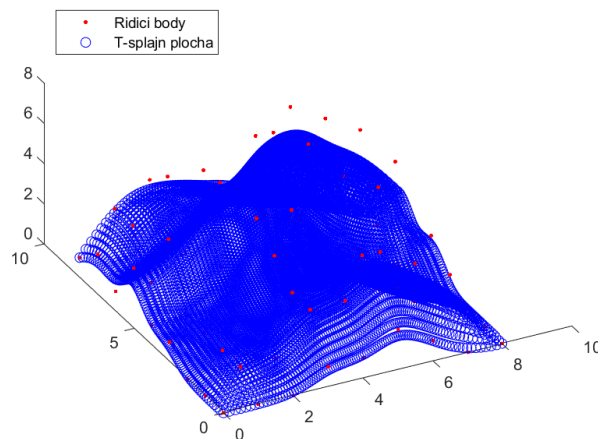
            [u1 ,v1]= vratIndexy (u ,v ,U ,V , sizeU , sizeV );
            [Nu ,Nv]= BazoveFunkce (u ,v ,u1 ,v1 ,U ,V , sizeU , sizeV );

            b = normalize (Nu .* Nv , ' norm ' ,1);

            Q(pocitadloR , pocitadloS ,1)= sum ((b ') .* (P (: ,1)));
            Q(pocitadloR , pocitadloS ,2)= sum ((b ') .* (P (: ,2)));
            Q(pocitadloR , pocitadloS ,3)= sum ((b ') .* (P (: ,3)));

            pocitadloS=pocitadloS +1;
        end
        pocitadloR=pocitadloR +1;
    end
    % vykresleni
    hold on
    plot3 (P (: ,1) ,P (: ,2) ,P (: ,3) , ' r . ')
    plot3 (Q (: ,: ,1) ,Q (: ,: ,2) , Q (: ,: ,3) , ' b * ')
end
```

---



Obrázek 20: T-splajn plocha

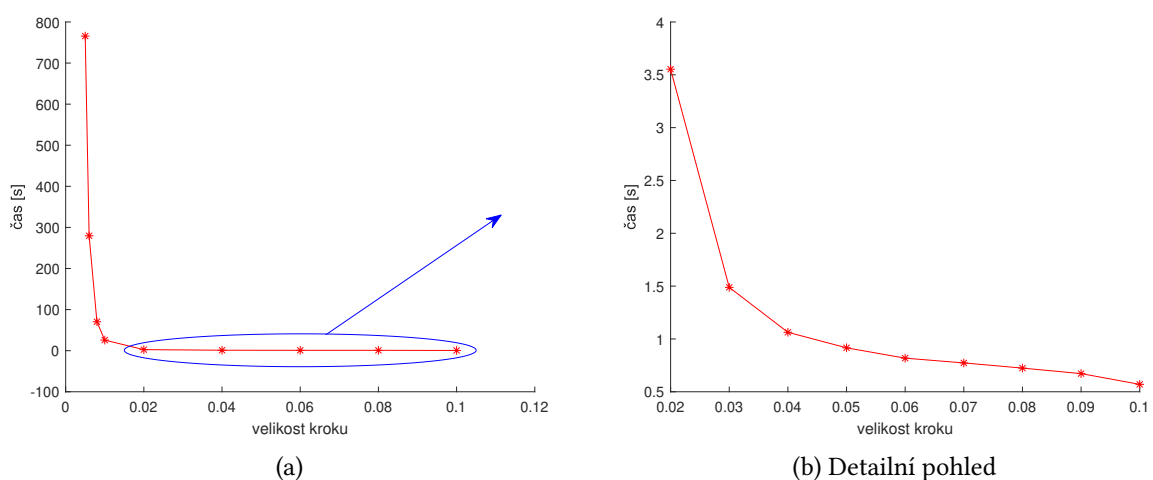
## 2.5 Rychlost a přesnost algoritmu pro výpočet T-splajn ploch

Tabulka 2 popisuje rychlost běhu algoritmu v závislosti na velikosti kroku, tedy na počtu bodů, který musí program vypočítat. Pro menší krok už počítač nebyl schopen algoritmus dokončit v přijatelném čase a tak byl výpočet ukončen.

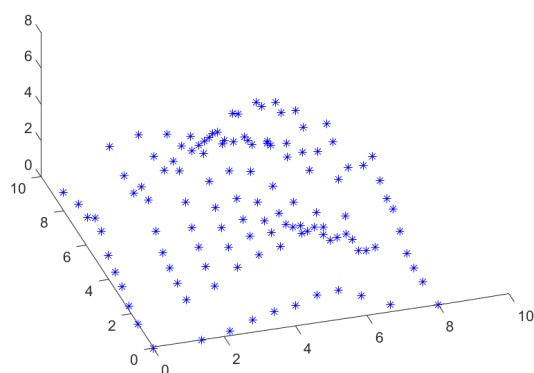
velikost kroku	0.1	0.08	0.06	0.04	0.02	0.01	0.008	0.006	0.005
čas [s]	0.57	0.84	0.90	1.11	2.50	25.74	70.38	279.65	765.61

Tabulka 2: Závislost času na velikosti kroku

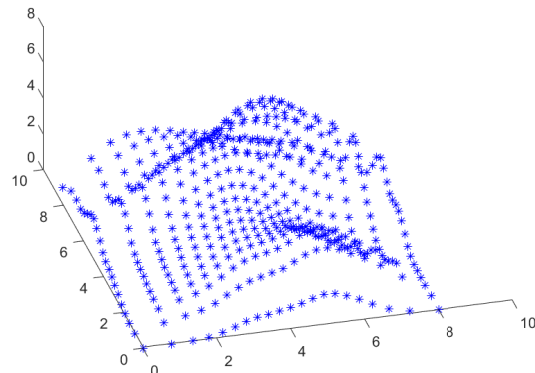
Na grafu 21 je vykreslena závislost z tabulky 2. Můžeme si všimnout, že přibližně od jemnosti kroku 0.01 začíná algoritmus značně zpomalovat a doba výpočtu stejně jako u B-splajn ploch roste exponenciálně. Obrázek 22 znázorňuje grafický výsledek algoritmu v závislosti na volbě kroku.



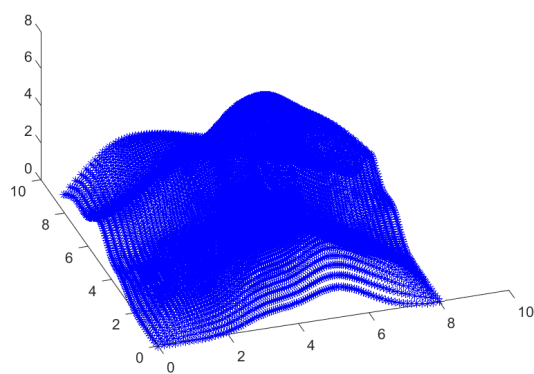
Obrázek 21: Rychlost programu v závislosti na velikosti kroku



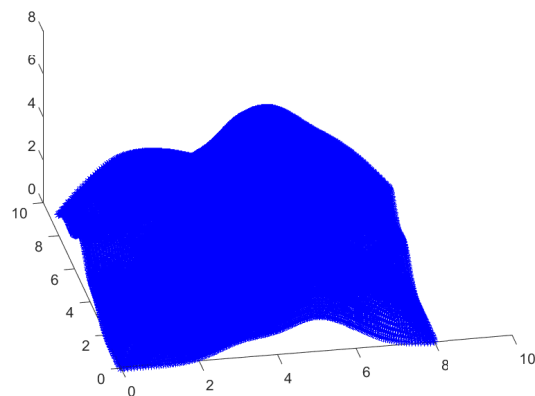
(a) Krok 0.1



(b) Krok 0.05



(c) Krok 0.01



(d) Krok 0.005

Obrázek 22: Výsledky algoritmu v závislosti na velikosti kroku

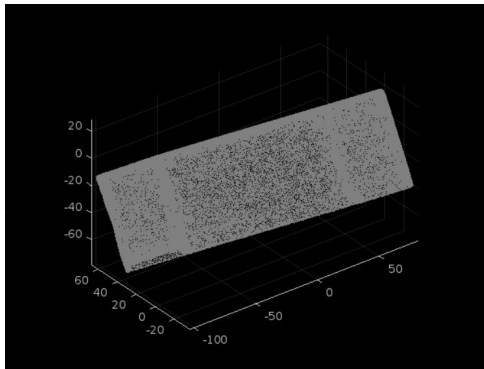
### 3 Aplikace na reálná data

Tato kapitola se bude zabývat prokládáním dat získaných ze 3D skenování objektu. Stejně jako v předchozích případech půjde o aplikaci B-splajn a T-splajn ploch.

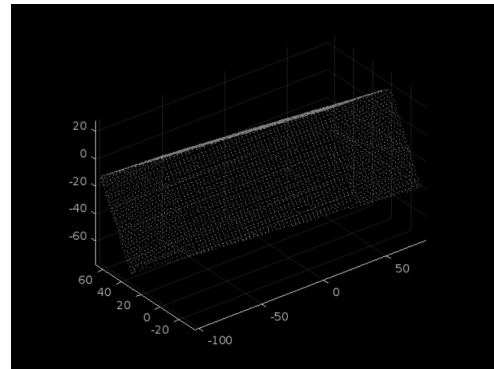
#### 3.1 B-splajny

##### 3.1.1 Získání a úprava dat

Objekt na obrázku 23a byl snímán na 3D skeneru ATOS CompactScan 2M od německého výrobce GOM mbH, Braunschweig. Je vybaven ATOS Professional scanning softwarem verze V.7.5 SR2, díky kterému lze data exportovat do standardních formátů pro mračna bodů. V našem případě se jedná o soubor formátu *.ply*, tedy matice bodů. Soubor obsahuje také další informace, například počet bodů a maximální rozsahy bodů na jednotlivých osách. Vzhledem k velkému množství bodů a jejich malým vzdálenostem bylo nutné je redukovat, aby zde bylo názorněji viditelné prokládání B-splajn plochou (obrázek 23b). Načtení souboru a jeho úpravy jsou popsány v algoritmu 4. Pro lepší přehlednost předvedeme úpravy a algoritmus pouze na části mračna a nakonec ho rozšíříme na celou oblast.



(a) Naskenovaná data



(b) Redukovaná data

Obrázek 23: Data ze skeneru

#### Zdrojový kód 4: Načtení mračna

---

```

ptCloud = pcread( ' Krabicka . ply ' );
e=1;b=1;c=1;d=1; pocetbodu2=100; gridStep = 3;

ptCloud = pcdownsampling(ptCloud, ' gridAverage ', gridStep );

roi = [ -105 -100 -35.8889 68 -77.2554 28.5259 ]; %vyber oblasti
indices = findPointsInROI( ptCloud , roi );
ptCloudB = select( ptCloud , indices );

pcshow( ptCloud . Location , [ 0.5 0.5 0.5 ] )      %vykresleni mracna
hold on
pcshow( ptCloudB . Location , ' r ' );

x=ptCloudB . Location ( : , 1 );
y=ptCloudB . Location ( : , 2 );
z=ptCloudB . Location ( : , 3 );
A=[x,y,z];          %ulozeni do matice

A=sortrows( A , 2 );
while ( e+pocetbodu2 - 1 ) <= ( ptCloudB . Count )
    B( e : ( e + ( pocetbodu2 - 1 ) ) , :) = sortrows( A( e : ( e + ( pocetbodu2 - 1 ) ) , :) , 1 );
    e=e+pocetbodu2 ;
end

```

---

### 3.1.2 Uspořádání bodů do mřížky

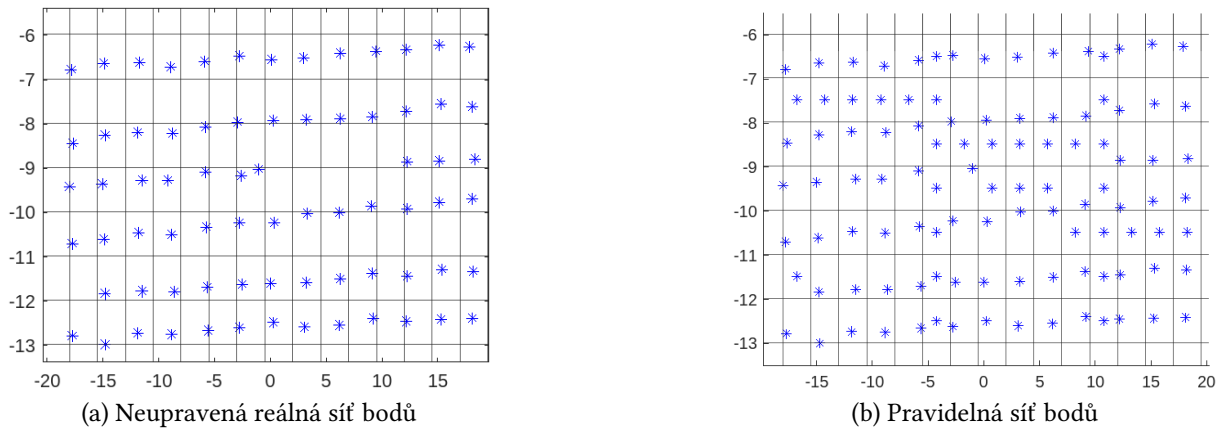
Aby mohl být algoritmus na prokládání bodů B-splajn plochou použit, je nutné body uspořádat do pravidelné mřížky. Nejdříve rozdělíme mřížku na jednotlivé pravidelné oblasti (viz obrázek 24a) a následně doplníme chybějící body do středů oblastí, které zůstaly prázdné (obrázek 25b).

Nyní je nutné doplnit třetí souřadnici nově vzniklých bodů, k čemuž využijeme IDW algorit-

mus (*Inverse Distance Weighting*). Algoritmus určí souřadnici  $z$  bodu, pokud zná souřadnice  $z$  pro  $n$  okolních bodů a jejich vzdálenost  $w$ . Výpočet je potom prováděn podle vztahu:

$$z(\mathbf{x}) = \frac{\sum_{i=1}^n w_i z_i}{\sum_{i=1}^n w_i} \quad (3.1)$$

Více podrobností o tomto algoritmu naleznete např. v [10].

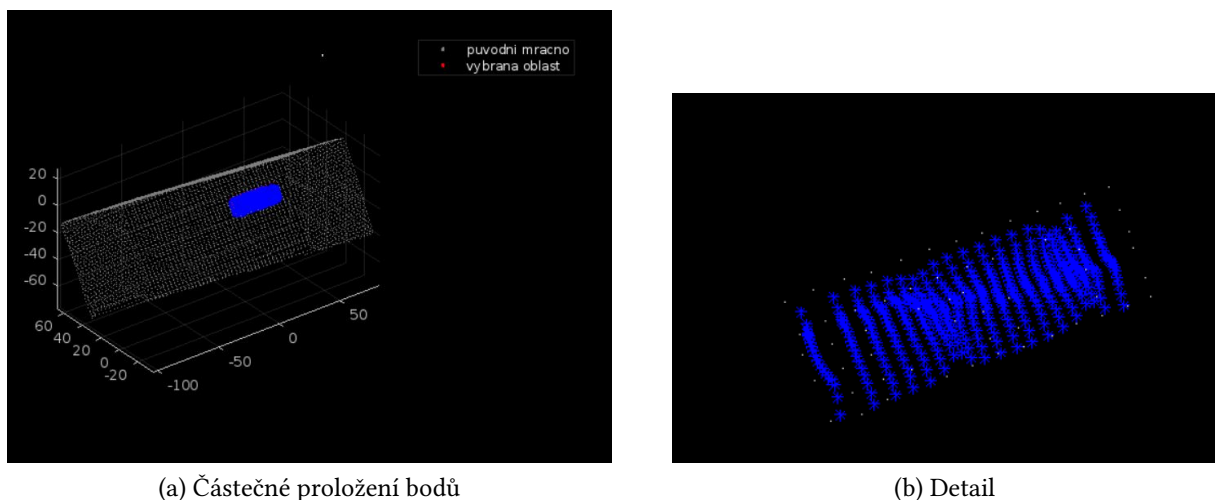


Obrázek 24: Úprava reálné sítě bodů na pravidelnou

Dále je potřeba upravit formát bodů, tedy přidat každému prvku informaci o tom, ve kterém řádku a sloupci se nachází.

### 3.1.3 Aplikace deBoorova algoritmu

Když jsou body naskládány do mřížky, lze na ně aplikovat deBoorův algoritmus pro B-splajn plochu 2. Aplikace bude probíhat po částech s ohledem na přesnost a náročnost výpočtu. Na obrázku 25 vidíme část mračna proloženého B-splajn plochou.



Obrázek 25: Aplikace B-splajn plochy na reálná data

## 3.2 T-splajny

V této kapitole byla použita stejná data jako v 3.1.1, ale nyní na ně byl aplikován algoritmus pro tvorbu T-splajn ploch. Pro načtení dat ze souboru byl použit zdrojový kód 4. Pouze uspořádání bodů proběhlo odlišně, jelikož je nutné jim dodat informaci o tom, ve kterém řádku a sloupci se nachází. To bude poté využito při generování uzlových vektorů ve skriptu 5. Použili jsme data doplněná na pravidelnou mřížku (obrázek 24), což je T-mesh bez T-spojů. Je nutné dodat, že zpracovávána byla jen část bodů celkového skenu a navíc ve 2D. Pro celý objekt by byl skript významně náročnější a nad rámec této práce.

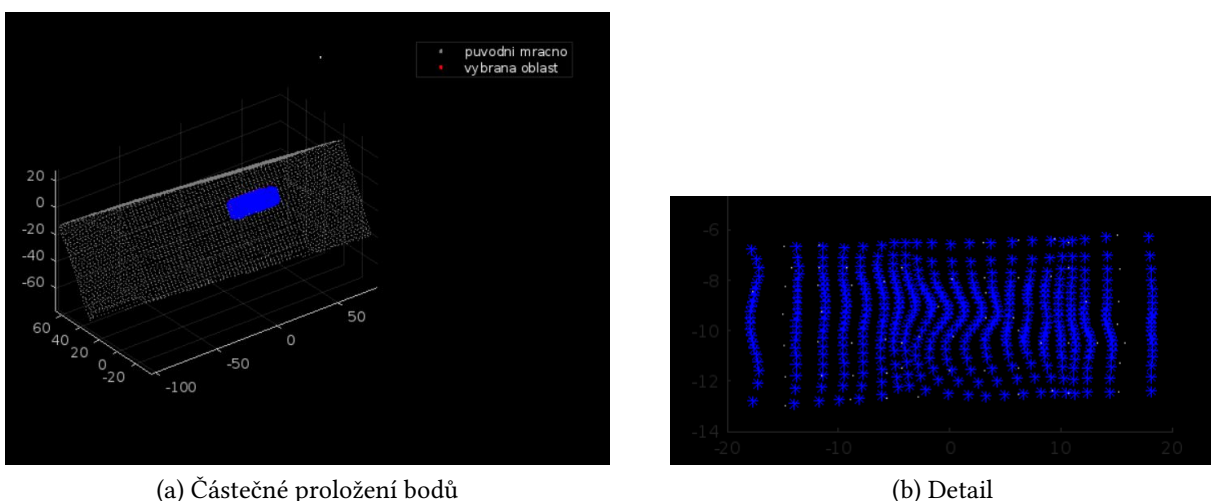
Vzhledem k tomu, že T-splajn plocha má možnost mít mřížku bodů nepravidelnou, prázdné políčka v obrázku nebudou působit problémy.

Skript 5 popisuje přiřazování uzlových vektorů jednotlivým bodům. Jedná se o směr  $u$ , tedy vodorovný. U směru  $v$  je postup podobný, pouze je třeba body seřadit podle souřadnice  $x$  a použít permutační matici, abychom měli na konci vektory ve správném pořadí pro vstup do skriptu.

Zdrojový kód 5: Generování uzlového vektoru  $u$

```
sizep=size(p); %vektor p znaci pocet bodu v kazdem radku
m=1;
for i=1:(sizep-1)
    vektorU=uzlvek(p(i),3);
    k=1;
    for j=m:m+p(i)-1
        U(j,1:5)=vektorU(k:(k+4));
        k=k+1;
    end
    m=m+p(i);
end
```

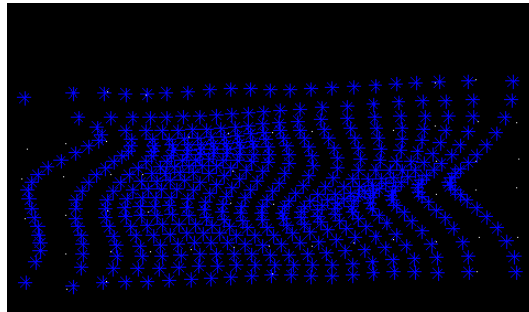
Nyní je již pouze aplikována funkce pro tvorbu T-spline plochy (strana 23). Výstupy z programu vidíme na obrázku 26.



Obrázek 26: Aplikace T-splajn plochy na reálná data

Kdybychom chtěli aplikovat funkci pro výpočet na neupravená data, struktura plochy se rozpadne (obrázek 27). Nebudou totiž dodržena pravidla pro tvorbu T-mesh. Tato skutečnost

značně komplikuje aplikaci T-splajnů na reálné body ze skeneru, jak se zmíníme v závěru této práce.



Obrázek 27: T-splajn na reálných datech ze skeneru

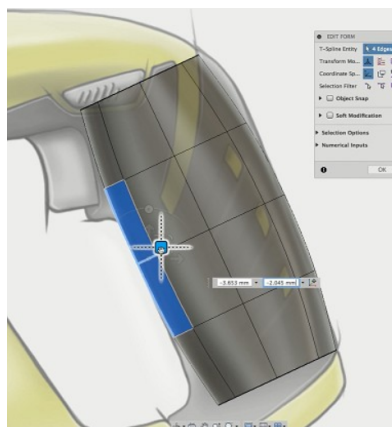
### 3.3 Využití B-splajn a T-splajn ploch a jejich srovnání

V této sekci zdůrazníme využití B-splajn a T-splajn ploch v nejrůznějších odvětvích. Dále se pokusíme se o jejich srovnání jak z hlediska časové, tak výpočetní náročnosti a budeme diskutovat, která je v určité situaci vhodnější.

#### Využití

Přestože byly B-splajny představeny před více než 50 lety, jejich využití exponenciálně vzrostlo v devadesátých letech, kdy jejich rozvoji pomohly hlavně výkonné počítače. Nyní jsou využívány například k odstranění šumu v obraze, tedy k vyhlazení křivek, o čemž pojednává článek [11]. Dalším odvětvím, kde se s plochami setkáme, je atomová a molekulární fyzika. Zde jsou využívány například i díky relativně malému počtu básových funkcí, které jsou potřeba, aby bylo dosaženo požadované přesnosti. Tím je šetřena paměť počítače a výpočty jsou rychlejší. Více v [12]. Nakonec uvedeme ještě využití ve výpočtech fluidní dynamiky, které je zmíněno v [13].

T-splajny jsou zahrnuty do spousty modelovacích softwarů jako je Autodesk Inventor, Autodesk Fusion 360 nebo Rhinoceros. Například ve zmiňovaném Fusion 360 se nachází předdefinované modely složené z několika T-splajn ploch (obrázek 28), které lze modifikovat (rotovat, posouvat, ...). Přitom je zachována jejich návaznost. Více v [14]



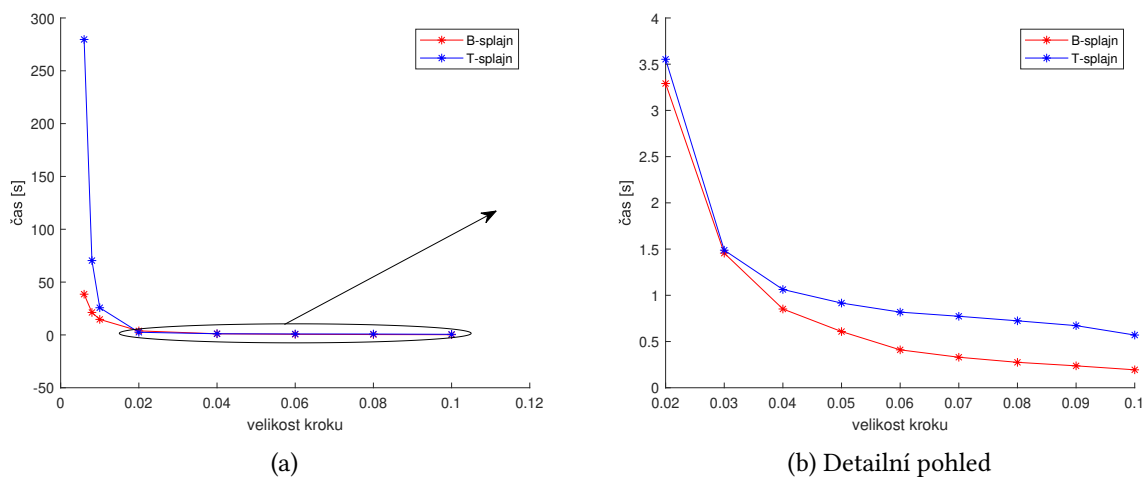
Obrázek 28: Ukázka z Fusion 360 [14]

## Srovnání výpočetní náročnosti B-splajn a T-splajn ploch

Nejdříve srovnáme obě plochy z hlediska trvání jejich výpočtu. Na obrázku 29a vidíme, že pro jemnější kroky, tedy 0.02 a menší, je trvání programu u T-splajnů výrazně pomalejší. Pro větší kroky (obrázek 29b) už jsou časy srovnatelné, konkrétně pro B-splajny téměř o půl sekundy kratší.

Nevýhodou T-splajnů, která také výrazně ovlivní dobu výpočtu, je generování uzlových vektorů. Pro každý bod potřebujeme dva uzlové vektory. Narozdíl od B-splajnů, kde potřebujeme dva uzlové vektory ve směru  $u$  a  $v$  pro celou mřížku bodů dohromady. Nutno zmínit také problematiku uspořádání reálných dat do T-mesh, aby byla dodržena obě pravidla pro její tvorbu (kapitola 1.4.2). Nelze totiž použít neuspořádané body ze skeneru jako řídicí mřížku. Proto jsou zřejmě v modelovacích softwarech plochy předdefinované a vztahy mezi body pevně nastavené.

Naopak výhodou T-splajnů je určitě jejich menší náročnost na počet řídicích bodů, které můžeme právě díky vlastnostem T-mesh značně redukovat. Také přidávání bodů bude méně náročné, protože nemusíme přidáním jednoho bodu doplnit i celý nový řádek nebo sloupec.



Obrázek 29: Srovnání časové náročnosti výpočtu B-splajn a T-splajn ploch.



## 4 Závěr

V této bakalářské práci jsme se zaměřili převážně na nastudování problematiky týkající se B-splajn křivek a ploch a jejich rozšíření na T-splajn plochy konstruované na nepravidelné síti bodů. Dále bylo cílem vytvořit program na proložení bodů těmito plochami a otestovat funkčnost a využití na reálných datech získaných 3D skenováním.

V první kapitole jsme položili teoretický základ týkající se matematických vlastností výše zmiňovaných ploch. Seznámili jsme se s uzlovými vektory a bázovými funkcemi, které jsme ověřili konkrétním příkladem. Definovali jsme si pravidla pro konstrukci T-mesh, tedy zmiňované nepravidelné sítě bodů a popsali tvorbu uzlových vektorů pro body v T-splajn mřížce.

Druhá kapitola se zabývala implementací algoritmů pro tvorbu ploch v softwaru Matlab. Srovnali jsme časovou náročnost výpočtu pro různou jemnost kroku a zjistili, že výpočet B-splajn ploch můžeme obecně prohlásit za rychlejší (podrobnosti v sekci 3.3).

Třetí kapitola se zabývá využitím těchto poznatků na data získaná na 3D skeneru. Nejdůležitější bylo data upravit tak, abychom na ně mohli algoritmus použít. Samotný výpočet potom proběhl úspěšně. Závěrem této kapitoly zmiňujeme především využití těchto ploch v různých odvětvích. U B-splajnů jde například o fluidní dynamiku, u T-splajnů o různé modelovací softwary. Dále se zde snažíme shrnout výhody a nevýhody jejich použití.

Do budoucna ještě určitě stojí za zmínku například úprava reálných bodů získaných skenerem do T-mesh. Museli bychom rozlišit, které chybějící body bude třeba doplnit, což by značně komplikovalo celý algoritmus. To ale bylo nad rámec této práce a ráda bych se tomu do budoucna věnovala podrobněji.

## Reference

- [1] MÍCHAL, Luboš. *Křivky v počítačové grafice* [online]. Brno, 2008 [cit. 2023-01-22]. Dostupné z: <http://www.sciencedirect.com/science/book/9780080964546>
- [2] LINKEOVÁ, Ivana. *Základy počítačového modelování křivek a ploch* [online]. Praha, 2020 [cit. 2023-01-22]. Dostupné z: [http://www.linkeova.cz/vyuka/pg\\_pgs/public/pgr\\_skripta.pdf](http://www.linkeova.cz/vyuka/pg_pgs/public/pgr_skripta.pdf)
- [3] KVASNIČKA, Ondřej. *Křivky v počítačové grafice* [online]. [cit. 2023-01-22]. Dostupné z: [http://home.zcu.cz/~smolik/zpg/cviceni/cv\\_08.html](http://home.zcu.cz/~smolik/zpg/cviceni/cv_08.html)
- [4] SHENE, Ching-Kuang. *CS3621 Introduction to Computing with Geometry Notes* [online]. Michigan: Michigan Technological University, c1997-2014, Last update: May 4, 2011 [cit. 2023-01-22]. Dostupné z: <https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/>
- [5] PIEGL, Les A. a Wayne TILLER. *The NURBS Book*. 2nd ed. New York: Springer, c1997. Monographs in visual communication. ISBN 35-406-1545-8.
- [6] JELÍNEK, Daniel. *Interpoláční a aproximační spline křivky* [online]. 38 s. Brno, 2020 [cit. 2023-01-22]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/125267>
- [7] SEDERBERG, Thomas W., Jianmin ZHENG, Almaz BAKENOV a Ahmand NASRI. *T-spline and T-NURCCs*. Journal ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2003, Vol. 22, No. 3, pp. 477-484, 2003.
- [8] CARDON, David L., G. Thomas FINNIGAN, Nicholas S. NORTH, Thomas W. SEDERBERG, Jianmin ZHENG. *T-splines Simplification and Local Refinement* [online]. Brigham Young University, 2004 [cit. 2023-02-13]. Dostupné z: <https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=2032&context=facpub>
- [9] PEŠKOVÁ, Irena. *T-spline objekty a jejich aplikace* [online]. Plzeň, 2015 [cit. 2023-02-13]. Dostupné z: [https://dspace5.zcu.cz/bitstream/11025/17980/1/DP\\_irenapeskova.pdf](https://dspace5.zcu.cz/bitstream/11025/17980/1/DP_irenapeskova.pdf)
- [10] HARTMANN, K., J. KROIS a B. WASKE. *E-Learning Project SOGA: Statistics and Geospatial Data Analysis* [online]. Department of Earth Sciences, Freie Universitaet Berlin, 2018 [cit. 2023-23-01]. Dostupné z: <https://www.geo.fu-berlin.de/en/v/soga/Geodata-analysis/geostatistics/Inverse-Distance-Weighting/index.html>
- [11] UNSER, M., A. ALDROUBI a M. EDEN. *B-spline signal processing*. IEEE Transactions on Signal Processing, vol. 41, no.2, pp. 821-833, Feb. 1993, doi: 10.1109/78.190220.
- [12] BACHAU, H., E. Comier, P. Decleva, J. E. Hansen a F. Martín. *Applications of B-splines in atomic and molecular physics*. Reports on Progress in Physics, vol. 64, no. 12, Nov. 2001, doi: 10.1088/0034-4885/64/12/205.
- [13] BOTELLA, Oliver a K. Shariff. *B-spline Methods in Fluid Dynamics*. International Journal of Computational Fluid Dynamics, 17:2, 133-149, Sep. 2010, doi: 10.1080/1061856031000104879.
- [14] SCHERER, Jamie. *Basics of T-splines* [online]. [cit. 2023-05-18]. Dostupné z: <https://help.autodesk.com/view/fusion360/ENU/courses/AP-BASICS-OF-T-SPLINES>