

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra matematiky

Diplomová práce

Plošné a objemové B-spline/NURBS parametrizace pro isogeometrickou analýzu

Místo této strany bude
zadání práce.

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 16. května 2018

Vojtěch Ouda

Poděkování

Chtěl bych poděkovat doc. Ing. Bohumíru Bastlovi, Ph.D. za jeho vstřícnost, rady a ochotu při vedení mé diplomové práce.

Abstrakt

Tato diplomová práce se zabývá B-Spline a NURBS parametrizací rovinných i prostorových oblastí pro isogeometrickou analýzu. Obsahuje shrnutí základních definic a vlastností B-spline a NURBS objektů. Stěžejní částí textu je popis hledání parametrizace oblasti, která je určena svojí hranicí. Hledání různých parametrizací využívá grafové struktury, tzv. grafu sousednosti. Hlavním přínosem práce je zobecnění hledání parametrizace i grafů sousednosti do 3D případu. Na závěr jsou na příkladech ilustrovány postupy z teoretické části a porovnány různé parametrizace.

Klíčová slova: B-spline, NURBS, isogeometrická analýza, plošná parametrizace, objemová parametrizace

Abstract

This diploma thesis takes a view of planar and volumetric B-spline/NURBS parameterizations for isogeometric analysis. It contains a summary of basic definitions and attributes of B-Spline and NURBS objects. The main part of the text is description of searching the parameterization of domain, which is determined by its boundary. Exploring of various parameterizations is using graphs, so called patch adjacency graphs. The main acquisition of this thesis is generalization of searching parameterization and patch adjacency graphs into 3D cases. In the end there are illustrations of advances in examples from theoretical part and there are compared different parameterizations as well.

Keywords: B-spline, NURBS, isogeometric analysis, planar parameterization, volumetric parameterization

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 7 |
| 2 | B-spline a NURBS | 9 |
| 2.1 | B-spline křivky | 9 |
| 2.2 | NURBS křivky | 11 |
| 2.3 | B-spline a NURBS plochy – plochy tenzorového součinu . . . | 11 |
| 3 | Parametrizace oblasti | 13 |
| 3.1 | Multi-patch parametrizace | 13 |
| 3.2 | Graf susednosti patchů | 15 |
| 3.2.1 | Generování PAG | 18 |
| 3.3 | Algoritmus multi-patch parametrizace | 26 |
| 3.3.1 | Vytvoření struktury řídicích bodů | 26 |
| 3.3.2 | Lineární optimalizace | 27 |
| 3.3.3 | Nelineární optimalizace | 27 |
| 4 | Parametrizace objemu | 29 |
| 4.1 | 3D multi-patch parametrizace | 29 |
| 4.2 | Graf susednosti | 31 |
| 4.2.1 | Generování grafu susednosti | 33 |
| 4.2.2 | Propojení b-grafu s p-grafem | 35 |
| 4.3 | Algoritmus MPP | 39 |
| 5 | Experimenty | 41 |
| 5.1 | 2D oblast | 41 |
| 5.2 | 3D objem | 46 |
| 6 | Závěr | 52 |
| | Literatura | 53 |

1 Úvod

Diplomová práce se zabývá parametrizací rovinných a prostorových oblastí pro další využití v isogeometrické analýze. Isogeometrická analýza je relativně mladá numerická metoda (formulovaná roku 2005) pro řešení parciálních diferenciálních rovnic. Isogeometrická analýza souvisí s metodou konečných prvků, ovšem využívá metod geometrického modelování. Je založena na NURBS prvcích, které jsou standardem využívaných v CAD systémech. Bázové funkce prostoru řešení a funkce použité pro popis geometrie jsou stejné. Typicky ale v CAD reprezentujeme oblast či objem, na kterých počítáme diferenciální rovnice, pomocí hraničních křivek nebo ploch. Ovšem pro použití isogeometrické analýzy potřebujeme popsat (parametrizovat) i vnitřek oblasti či objemu. Další informace o isogeometrické analýze lze najít v [3].

Práce je rozdělena na čtyři kapitoly. První se věnuje B-spline a NURBS objektům, jsou zde zavedené některé pojmy, uvedené vlastnosti těchto objektů a některé poznatky jsou demonstrovány na příkladech. Více o B-spline a NURBS objektech je např. v [10].

Další kapitola už se věnuje samotné parametrizaci dvojrozměrné oblasti. Konkrétně takové parametrizaci, kdy danou oblast rozdělíme na menší, po dvou disjunktní podoblasti, a každou parametrizujeme jako jeden prvek – hledáme tedy tzv. multi-patch parametrizaci. Výhodou tohoto popisu pomocí více podoblastí je fakt, že složité tvary můžeme popsat větším počtem jednodušších oblastí. Navíc ne každou oblast umíme parametrizovat jedním plátem. Je zřejmé, že takových parametrizací bude existovat více, hlavní otázkou této kapitoly tedy bude, jakým způsobem získat optimální multi-patch parametrizaci. Strukturu každé takové parametrizace můžeme reprezentovat grafem, budeme tedy zkoumat i vlastnosti těchto grafů a možnosti jejich generování, případně algoritmus hledání uvést na příkladu. Literatura, která popisuje hledání parametrizace 2D oblasti, je např. [2], [4], [5], [9].

Další kapitola se věnuje jednomu z hlavních cílů této práce, tedy zobecnění poznatků multi-patch parametrizace z 2D oblasti do trojrozměrného objemu. Literatura, která se problematikou parametrizace objemu zabývá, je např. [1], [7].

Poslední kapitola se věnuje numerickým experimentům, kde jsou na příkladech demonstrovány poznatky z předešlých kapitol. Cílem kapitoly je především ozřejmit vliv některých funkcionalů zavedených v teoretické části na výsledný tvar parametrizace, případně ilustrovat vliv různých grafových struktur, následně různé výsledky porovnat. Veškeré výpočty a implementace probíhaly v softwaru Wolfram Mathematica 10.0.

2 B-spline a NURBS

Tato kapitola shrne základní poznatky problematiky B-spline a NURBS křivek a ploch. Kapitola je do práce zařazena z toho důvodu, že kdykoli budeme mluvit o parametrizaci, bude myšlena B-spline či NURBS parametrizace. Další podrobnosti je možné najít v [6], [10].

2.1 B-spline křivky

B-spline i NURBS křivky jsou aproximačními křivkami řídicích bodů. Jednoznačně B-spline křivku určuje:

- vektor parametrizace (uzlový vektor) $\mathbf{T} = (t_0, t_1, \dots, t_m)$, kde $t_0 \leq t_1 \leq \dots \leq t_m$,
- B-spline báze funkce N_i^k stupně k , $i = 0, \dots, n$:

– pro $k = 0$

$$N_i^0(t) = \begin{cases} 1 & \text{pro } t_i \leq t < t_{i+1} \\ 0 & \text{jinde} \end{cases}, \quad (2.1)$$

– pro $k > 0$

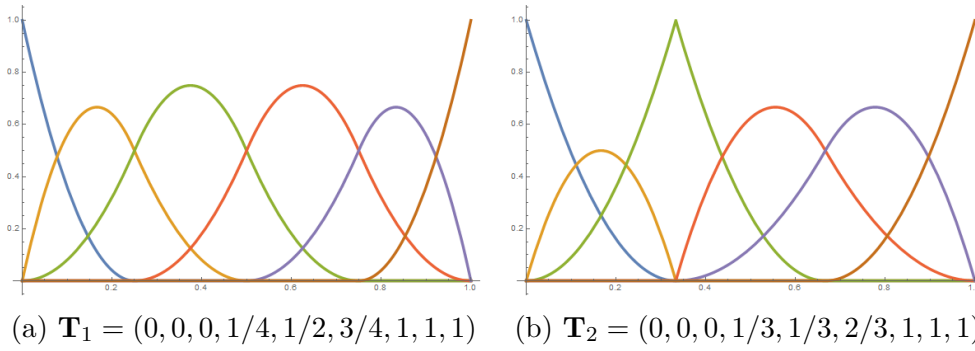
$$N_i^k(t) = \frac{t - t_i}{t_{i+k} - t_i} N_i^{k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1}^{k-1}(t), \quad (2.2)$$

- řídicí polygon (řídicí body) $\mathbf{V}_0, \mathbf{V}_1, \dots, \mathbf{V}_n$.

Ve vztahu (2.2) pro výpočet B-spline báze může vzniknout výraz typu $\frac{a}{0}$, který položíme roven nule. Ze vztahu (2.2) také plyne, že vektor parametrizace musí obsahovat složku složku t_{n+k+1} – tedy stupeň B-spline báze a počet B-spline báze funkcí je provázaný vztahem $m = n + k + 1$.

Vektor parametrizace může být buďto *uniformní* – složky vektoru jsou v parametrickém prostoru rovnoměrně rozloženy, nebo *neuniformní*. Dále rozlišujeme *periodický* a *neperiodický* vektor parametrizace. Neperiodický má prvních $k + 1$ a posledních $k + 1$ složek stejných.

Protože složky vektoru parametrizace tvoří neklesající posloupnost, může být ve vektoru l uzlů stejné hodnoty ($1 < l \leq k$), pak hovoříme o l -násobném uzlu.



Obrázek 2.1: B-spline báze – vliv různých vektorů parametrizace

B-spline křivka $\mathbf{P}(t)$ je určena lineární kombinací B-spline bázových funkcí s řídicími body:

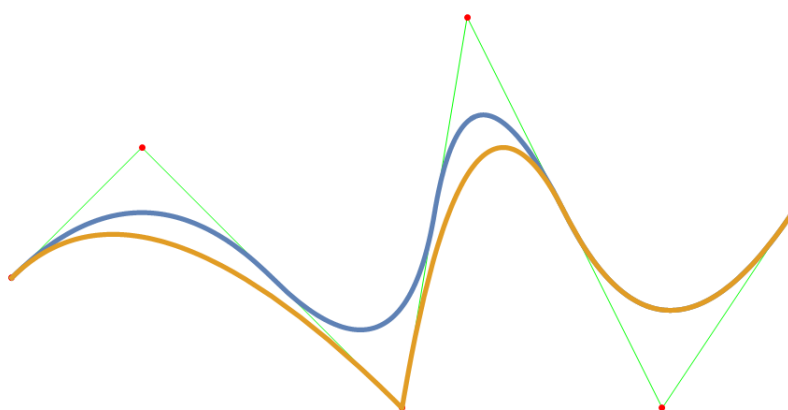
$$\mathbf{P}(t) = \sum_{i=0}^n \mathbf{V}_i N_i^k(t). \quad (2.3)$$

Support bázové funkce $N_i^k(t)$ je lokální – na intervalu $[t_i, t_{i+k+1})$; na intervalu $[t_i, t_{i+1})$ jsou nenulové pouze bázové funkce N_{i-k}^k, \dots, N_i^k . Hlavní výhodou oproti např. Bézierovým křivkám je, že změna souřadnic řídicího bodu ovlivní výslednou B-spline křivku pouze lokálně. Další výhodou je, že s rostoucím počtem bodů řídicího polygonu neroste stupeň křivky.

Za předpokladu, že v uzlovém vektoru nejsou násobné uzly, B-spline křivka je obecně třídy C^{k-1} . Je-li v uzlovém vektoru l -násobný uzel, snižuje se třída křivky na C^{k-l} .

Příklad bázových funkcí pro dva různé vektory parametrizace a vliv násobného uzlu je uveden na obrázku 2.1, kde oba uzlové vektory jsou neperiodické, obrázek 2.1a ukazuje B-spline bázové funkce pro vektor parametrizace $\mathbf{T}_1 = (0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1)$ (uniformní uzlový vektor), obrázek 2.1b pro $\mathbf{T}_2 = (0, 0, 0, 1/3, 1/3, 2/3, 1, 1, 1)$ (neuniformní uzlový vektor). První a poslední tři uzly jsou stejné, tedy stupeň bázových funkcí je $k = 2$, uzlů ve vektoru je 9, tedy $m = 8$. Ze vztahu $m = n + k + 1$ musí být $n = 5$, tedy počet bázových funkcí je 6. Protože ve druhém případě je jeden dvojnásobný uzel, v bodě $t = 1/3$ je bázová funkce třídy C^0 .

Na obrázku 2.2 jsou zobrazeny dvě B-spline křivky pro jeden řídicí polygon a pro různé vektory parametrizace z předešlého příkladu. Modrou křivku určuje vektor parametrizace \mathbf{T}_1 , oranžovou \mathbf{T}_2 .



Obrázek 2.2: B-spline křivky – vliv různých vektorů parametrizace

2.2 NURBS křivky

Zkratka NURBS je z anglického Non Uniform Rational B-Spline, tedy neuniformní racionální B-spline, a je zobecněním B-spline křivek. NURBS křivka je stejně jako B-spline křivka určena vektorem parametrizace, řídicími body a stupněm básových funkcí, navíc ale ke každému řídicímu bodu \mathbf{V}_i přiřadíme i jeho váhu w_i .

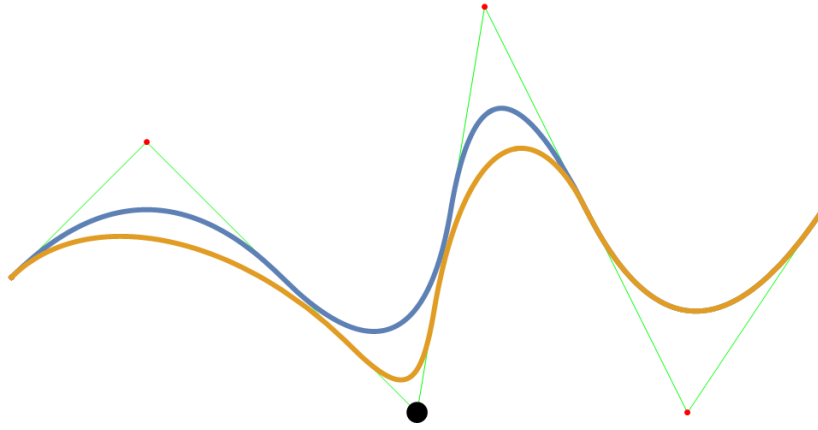
Výsledná křivka je racionální a je definována následujícím vztahem:

$$\mathbf{P}(t) = \frac{\sum_{i=0}^j N_i^k(t) w_i \mathbf{V}_i}{\sum_{i=0}^j N_i^k(t) w_i}. \quad (2.4)$$

Jsou-li váhy všech řídicích bodů stejné, potom je výsledná křivka B-spline křivkou. Výhodou NURBS křivek je, že dokáží popsat i křivky, které nejdou vyjádřit B-spline parametrizací, např. části kružnice. Na obrázku 2.3 je znázorněn význam vah v jednotlivých bodech – B-spline křivka je zobrazena modře, oranžově NURBS křivka, která má ve zvýrazněném černém bodě váhu $w_2 = 3$, v ostatních bodech váhy $w_i = 1$, $i = 0, 1, 3, 4, 5$. Ostatní vstupní data, tedy vektor parametrizace, stupeň i řídicí polygon mají tyto křivky shodné. Na tomto obrázku můžeme vidět i lokální vliv bodu na výslednou křivku.

2.3 B-spline a NURBS plochy – plochy tenzorového součinu

Parametrizace B-spline a NURBS ploch vycházejí z vlastností a způsobu parametrizace křivek, jsou tedy určeny:



Obrázek 2.3: Porovnání B-spline křivky a NURBS křivky

- vektory parametrizace (uzlovými vektory) $\mathbf{U} = (u_0, \dots, u_p)$, $\mathbf{V} = (v_0, \dots, v_q)$ pro parametry u, v plochy,
- řídicí sítě $(m + 1) \times (n + 1)$ tvořenou body $\mathbf{W}_{i,j}$, $i = 0, \dots, m$, $j = 0, \dots, n$,
- B-spline bázovými funkcemi jedné proměnné a jejich stupni pro u, v , tedy polynomy $N_i^k(u), N_j^l(v)$ definované vztahy (2.1), (2.2).
 - Navíc pro NURBS plochy váhami $w_{i,j}$ v bodech $\mathbf{W}_{i,j}$.

B-spline plocha je pak určena vztahem

$$\mathbf{P}(u, v) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{W}_{i,j} N_i^k(u) N_j^l(v), \quad (2.5)$$

NURBS plocha vztahem

$$\mathbf{P}(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n w_{i,j} \mathbf{W}_{i,j} N_i^k(u) N_j^l(v)}{\sum_{i=0}^m \sum_{j=0}^n w_{i,j} N_i^k(u) N_j^l(v)}. \quad (2.6)$$

V další části textu budeme parametrizací rozumět B-spline nebo NURBS parametrizace. Tato parametrizace je jednoznačně určena uzlovým vektorem, stupněm a řídicími body (případně váhami). Budeme předpokládat, že všechny hraniční křivky nebo plochy jedné oblasti budou mít stejný stupeň a stejný vektor parametrizace. Budeme-li hledat parametrizaci, omezíme se pouze na hledání řídicích bodů.

3 Parametrizace oblasti

3.1 Multi-patch parametrizace

Tato část se věnuje parametrizaci oblasti v rovině. Hlavním zdrojem pro metodu parametrizování byl článek [2]. Ten popisuje nalezení parametrizace s pomocí grafů přiřazených k hledané parametrizaci, přičemž jsou porovnány všechny parametrizace, které vzniknou z různých grafů, a vybrána ta s nejlepšími vlastnostmi.

Nejprve zavedeme některé pojmy a značení potřebné k popsání metody. Mějme jednoduše souvislou rovinnou oblast Ω . Její hranice je parametrizována b křivkami

$$c_i : [0, 1] \rightarrow \mathbb{R}^2, \quad i = 1, 2, \dots, b. \quad (3.1)$$

Tyto křivky budeme nazývat *segmenty*. Pro všechny segmenty platí:

$$\begin{aligned} c_i(1) &= c_{i+1}(0) \quad \forall i = 1, \dots, b-1, \\ c_b(1) &= c_1(0), \\ \forall i, j, \quad \forall s, t \in [0, 1] : c_i(s) &= c_j(t) \Rightarrow (i = j \wedge s = t). \end{aligned} \quad (3.2)$$

První dvě vlastnosti říkají, že segmenty tvoří uzavřenou křivku, třetí hovoří o tom, že se segmenty neprotínají jinde než v koncových bodech dvou sousedních segmentů. Segmenty tedy tvoří hranici jednoduše souvislé oblasti a budeme uvažovat parametrizace orientované ve směru hodinových ručiček.

Každé dva sousední segmenty se protnou ve vrcholu – buďto v konvexním, jestliže vnitřní úhel je menší než π , nebo nekonvexním, je-li vnitřní úhel větší nebo roven π .

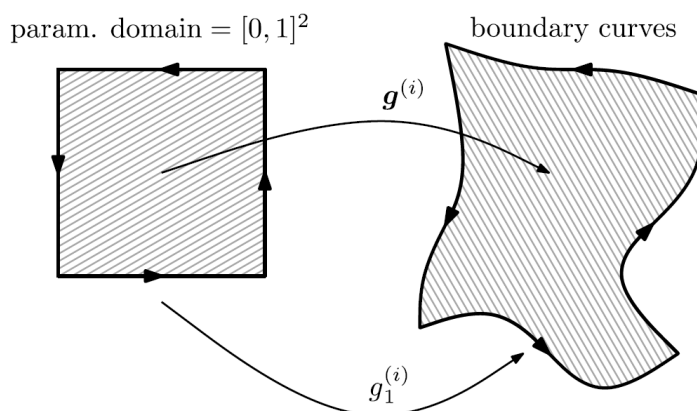
Parametrizaci oblasti Ω , kdy celou oblast rozdělíme na p částí, tzv. *patchů*,

$$\mathbf{g}^{(j)} : [0, 1]^2 \rightarrow \mathbb{R}^2, \quad j = 1, \dots, p, \quad (3.3)$$

budeme říkat *multi-patch paramerizace (MPP)*. Hranice každého patche $\mathbf{g}^{(j)}$ je složena ze 4 křivek $g_i^{(j)}, i = 1, \dots, 4$, které mají orientaci proti směru

hodinových ručiček (obrázek 3.1).

$$\begin{aligned}
 g_1^{(j)} &: [0, 1] \longrightarrow \mathbb{R}^2, & t &\longmapsto \mathbf{g}^{(j)}(t, 0), \\
 g_2^{(j)} &: [0, 1] \longrightarrow \mathbb{R}^2, & t &\longmapsto \mathbf{g}^{(j)}(1, t), \\
 g_3^{(j)} &: [0, 1] \longrightarrow \mathbb{R}^2, & t &\longmapsto \mathbf{g}^{(j)}(1 - t, 1), \\
 g_4^{(j)} &: [0, 1] \longrightarrow \mathbb{R}^2, & t &\longmapsto \mathbf{g}^{(j)}(0, 1 - t).
 \end{aligned} \tag{3.4}$$



Obrázek 3.1: Parametrizace patche (převzato z [2])

Řekneme, že patch je *regulární*, jestliže determinant Jakobiho matice $\nabla \mathbf{g}^{(j)}$ je kladný. Kladný jakobián pak implikuje, že žádný vnitřní úhel dvou hraničních křivek jednoho patche není více než π (více viz [5]).

Pro segmenty jsme uvažovali orientaci parametrizace po směru hodinových ručiček, u hraničních křivek patchů orientaci proti směru. Nyní, když zavedeme množinu S všech těchto křivek:

$$S = \{g_l^{(j)} \mid l = 1, \dots, 4; j = 1, \dots, p\} \cup \{c_i \mid i = 1, \dots, b\}, \tag{3.5}$$

můžeme zavést relaci \sim na této množině takovou, že:

$$s \sim s' \Leftrightarrow s(t) = s'(1 - t), \forall t \in [0, 1], s, s' \in S. \tag{3.6}$$

Tedy v relaci jsou takové křivky, které se překrývají v celé své délce a zároveň mají opačnou orientaci. Tato relace nám pomůže s definováním takové multi-patch parametrizace, která bude mít očekávatelné vlastnosti.

Poznámka. Je zřejmé, že platí $s \sim s' \Leftrightarrow s' \sim s$, tedy tato relace je symetrická.

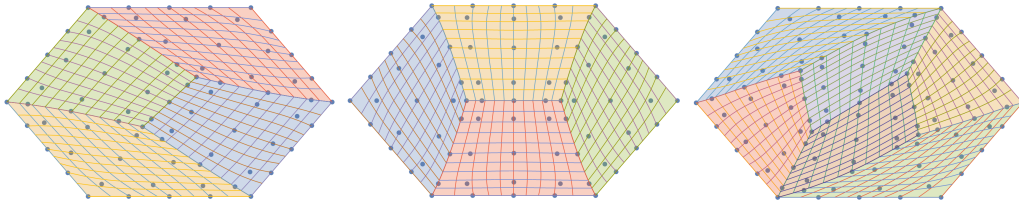
Definice 1 (Validní multi-patch parametrizace). *Množinu patchů $\{\mathbf{g}^{(j)}\}_{j=1}^p$*

nazveme validní multi-patch parametrizací (validní MPP), jestliže platí:

- (i) všechny patche jsou regulární,
- (ii) $\text{int}(\mathbf{g}^{(i)}) \cap \text{int}(\mathbf{g}^{(j)}) = \emptyset, \quad \forall i \neq j$
- (iii) žádné dvě hraniční křivky jednoho patche spolu nejsou v relaci \sim ,
- (iv) $\forall s \in S \exists! s' \in S : s \sim s'$.

Vlastnosti (i), (ii) a (iii) v definici jsou zřejmé – požadujeme, aby se patche nepřekrývaly jinde než na jejich hraničních křivkách, aby vnitřní úhel dvou hraničních křivek jednoho patche byl méně než π . Vlastnost (iv) říká, že každá křivka (segment nebo hraniční křivka patche) je v relaci právě s jednou křivkou. Z toho ale plyne, že v množině S musí být sudý počet křivek, protože všechny křivky utvoří dvojice. Dále počet všech hraničních křivek $g_i^{(j)}$ je také sudý, tedy počet b segmentů c_i musí být také sudý. V dalším textu tedy budeme uvažovat oblasti, jejichž hranice bude tvořena sudým počtem křivek.

Z vlastností v definici 1 pak plyne, že patche vyplňují celou oblast Ω . Tedy validní multi-patch parametrizace je taková parametrizace, kdy je nějaká oblast reprezentována jako sjednocení menších disjunktních oblastí, jejichž všechny vrcholy jsou konvexní. Na obrázku 3.2 jsou zobrazeny různé validní MPP parametrizace jedné oblasti.



Obrázek 3.2: Příklad různých rozdělení oblasti pomocí patchů

3.2 Graf sousednosti patchů

Cílem této kapitoly je definovat strukturu, pomocí níž budeme hledat různé validní multi-patch parametrizace. Poznamenejme, že dále budeme v textu používat slovo *vrchol* pro označení průsečíku dvou křivek. Pro pojem z teorie grafů budeme používat označení *uzel*.

Ke každé multi-patch parametrizaci můžeme jednoznačně přiřadit graf takový, pro který bude platit:

- každá křivka c_i je reprezentována v grafu jedním uzlem, těmto uzlům budeme říkat b-uzly,
- každý patch $\mathbf{g}^{(j)}$ je reprezentován jedním uzlem v grafu – p-uzly.

Hranou jsou v grafu spojeny takové uzly, které reprezentují:

- (i) c_i, c_j , jestliže $c_i(0) = c_j(1)$,
- (ii) $c_i, \mathbf{g}^{(j)}$, jestliže $c_i \sim g_l^{(j)}$ pro nějaké $l = 1, \dots, 4$,
- (iii) $\mathbf{g}^{(i)}, \mathbf{g}^{(j)}$, jestliže $g_k^{(i)} \sim g_l^{(j)}$ pro nějaká $k, l = 1, \dots, 4$.

Jestliže segmenty na sebe navazují, tj. mají jeden společný (koncový) bod, pravidlo (i) říká, že b-uzly, které reprezentují tyto segmenty, jsou v grafu spojeny hranou. Pravidlo (ii) pak hovoří o sousednosti segmentu a patche, pravidlo (iii) o sousednosti dvou patchů.

Z konstrukce grafu a vlastností validní multi-pach parametrizace plynou vlastnosti tohoto grafu:

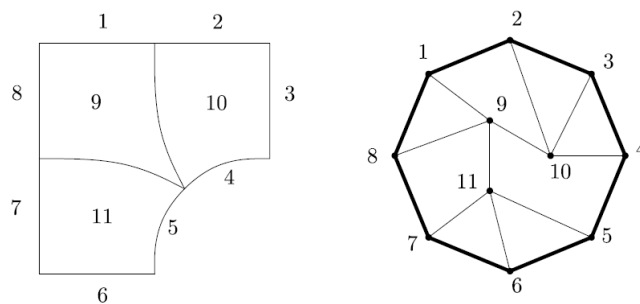
1. Počet uzlů v grafu je $b + p$,
2. každý b-uzel má valenci 3,
3. každý p-uzel má valenci 4,
4. v grafu nejsou smyčky,
5. graf je rovinný,
6. podgraf indukovaný všemi b-uzly je kružnice délky b ,
7. podgraf indukovaný p-uzly je souvislý.

Vlastnost 1 je zřejmá. Vlastnost 2 vyplývá z toho, že každý segment sousedí právě se dvěma dalšími segmenty a právě jedním patchem. Protože každý patch má 4 hraniční křivky a každá křivka je v relaci s právě jednou jinou křivkou, platí vlastnost 3. Protože jsme vyloučili případy, kdy dvě hraniční křivky jednoho patche jsou v relaci \sim , nebo samoprůniky segmentů, nemůže graf obsahovat smyčky. Rovinnost grafu je dána jednoduchou souvislou oblastí a validní MPP parametrizací. Vlastnosti 6, 7 jsou zřejmé.

Definice 2. *Takový graf, který má výše uvedené vlastnosti 1 – 7, nazveme graf susednosti patchů (PAG¹).*

V grafu susednosti očísľujeme b-uzly čísky 1, ..., b; p-uzly čísky b + 1, ..., b + p.

Příklad PAG je uveden na obrázku 3.3, kde vlevo můžeme vidět hranici oblasti určenou jednotlivými segmenty (křivky 1, 2, ..., 8), a patche (9, 10, 11). Obrázek vpravo ukazuje graf susednosti patchů, který z této MPP vznikne.



Obrázek 3.3: Segmenty a patche – vznik grafu (převzato z [2])

Dále definujeme funkce:

$$\begin{aligned} I : \{1, \dots, b\} &\longrightarrow \{b + 1, \dots, b + p\}, \\ E, N, W, S : \{b + 1, \dots, b + p\} &\longrightarrow \{1, \dots, b + p\}. \end{aligned} \quad (3.7)$$

Funkce I b-uzlu přiřadí p-uzel takový, který s ním je spojen hranou; funkce E, N, W, S přiřazuje p-uzlu jeho čtyři sousedy (p-uzly i b-uzly). Označení E, N, W, S je převzato z anglického *East, North, West, South*.

Jako příklad tohoto přiřazení může být v obrázku 3.3 pro b-uzel 7 a p-uzel 11 následující:

$$\begin{aligned} I(7) &= 11, \\ E(11) &= 5, \\ N(11) &= 9, \\ W(11) &= 7, \\ S(11) &= 6. \end{aligned} \quad (3.8)$$

Poznamenejme, že funkce I je pro PAG určena jednoznačně, ostatní funkce nejsou jednoznačně určeny a slouží spíše pro snazší orientaci – E, N, W, S jdou proti směru hodinových ručiček. Tedy další možnost přiřazení

¹Z anglického patch adjacency graph.

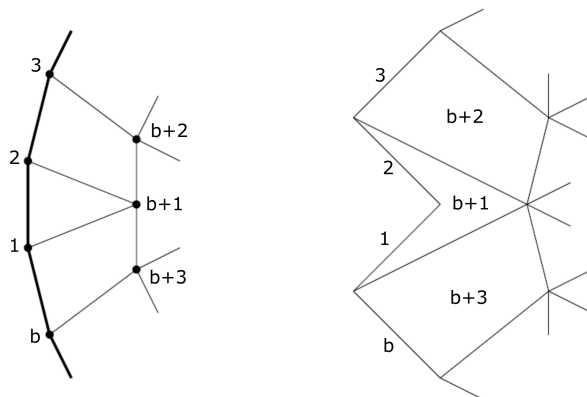
p-uzlu s číslem 11 mohlo být $E(11) = 9, N(11) = 7, W(11) = 6, S(11) = 5$. Nemusí tedy obecně platit například to, že $E(i) = j \wedge W(j) = i$.

Nyní tedy můžeme každé multi-patch parametrizaci přiřadit graf sousednosti a z něj získat nějaké přiřazení jednotlivých uzlů výše uvedenými funkcemi, které vyjadřuje strukturu MPP.

Metoda pro zjištění multi-patch parametrizace, která využívá těchto grafů, bude pracovat tak, že prohledá pro známý počet segmentů a pro uživatelem zadaný počet patchů všechny možnosti PAG. Následně nalezne pro každý graf možné souřadnice řídicích bodů a tedy požadovanou validní multi-patch parametrizaci. Musíme proto najít všechny možné grafy, které by mohly reprezentovat námi hledanou parametrizaci. Protože ale PAG zachycuje pouze strukturu a ne geometrii patchů, musíme ještě upřesnit, jaké grafy sousednosti budeme brát v potaz.

Definice 3. *Validní graf sousednosti patchů (validní PAG) je takový PAG přiřazený k nějaké množině segmentů, kdy pro všechny segmenty c_i, c_{i+1} , které tvoří nekonvexní vrcholy, platí $I(i) \neq I(i+1)$.*

Požadavek na validní PAG v definici 3 jen vylučuje ty případy, kdy bychom vygenerovali nějaký PAG takový, který by byl přiřazen k nevalidní multi-patch parametrizaci. Takový případ je zobrazen na obrázku 3.4, kde vlevo je graf sousednosti, ze kterého by vznikla nevalidní multi-patch parametrizace díky nekonvexnímu vrcholu na hranici. Tedy tento PAG není vzhledem k oblasti, pro kterou byl hledán, validní.

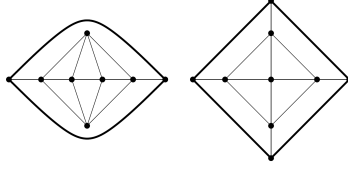


Obrázek 3.4: Nevalidní PAG (převzato z [2])

3.2.1 Generování PAG

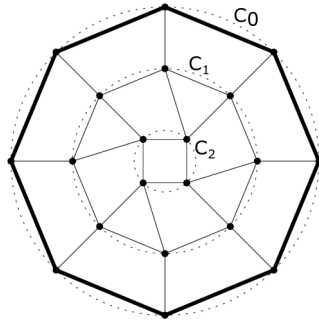
Z kapitoly 3.1 víme, že počet segmentů musí být sudý. Následující lemma říká, že vždy dokážeme najít nějaký validní PAG.

Lemma 1. *Validní graf sousednosti patchů existuje pro zadanou oblast Ω , jestliže počet segmentů je sudý.*



Obrázek 3.5: Pro speciální případ $b = 2$ a $b = 4$ (z [2])

Důkaz. Pro $b = 2, b = 4$ je PAG zobrazen na obrázku 3.5. Pro $b \geq 6$:
V grafu sousednosti b -uzly tvoří kružnici, označme ji \mathcal{C}_0 . Počet p -uzlů zvolme $p = b + \frac{b}{2}$ a tyto uzly budou tvořit dvě kružnice: \mathcal{C}_1 délky b a \mathcal{C}_2 délky $\frac{b}{2}$. Důkaz a konstrukce tohoto PAG je ilustrován na obrázku 3.6. Každý b -uzel z kružnice \mathcal{C}_0 spojíme hranou právě s jedním p -uzlem z \mathcal{C}_1 ; každý p -uzel z \mathcal{C}_2 spojíme právě se dvěma p -uzly z \mathcal{C}_1 , které spolu sousedí. Každý uzel v \mathcal{C}_0 tedy sousedí s právě třemi dalšími uzly, každý uzel v \mathcal{C}_1 a \mathcal{C}_2 sousedí právě se 4 uzly. Navíc $I(i) \neq I(i + 1) \forall i = 1, \dots, b$, tedy získaný graf je validní graf sousednosti patchů nějaké multi-patch parametrizace. \square



Obrázek 3.6: Vytvoření validního PAG

Víme tedy, že pro každou oblast Ω zadanou pomocí sudého počtu segmentů, získáme alespoň jeden validní PAG s určitým počtem p -uzlů. Dále bude popsán algoritmus na získání PAG s jiným počtem p -uzlů.

Algoritmus pro získání PAG

Mějme oblast Ω reprezentovanou pomocí sudého počtu b segmentů c_i . Zvolme číslo p , které vyjadřuje počet p -uzlů.

1. Segmenty jsou reprezentovány b-uzly. Ty budou tvořit kružnici, označme je čísla $1, \dots, b$, číslujeme proti směru hodinových ručiček.
2. Ke každému b-uzlu s čísly $1, \dots, b - 1$ vytvoříme nový uzel, který nazveme b-list a ten spojíme s b-uzlem.
3. Nyní sestrojíme úplný ternární strom², který má kořen v posledním b-uzlu. Počet vnitřních uzlů tohoto stromu bude p , tedy počet listů je roven $2p + 1$. Očíslujeme listy proti směru hodinových ručiček čísly $1, \dots, 2p + 1$.
4. Každý b-list i spojíme hranou s nějakým listem stromu f_i tak, aby $(f_i)_{i=1}^{b-1}$ byla rostoucí posloupnost a aby f_1 a každý rozdíl $f_{i+1} - f_i$ byla lichá čísla.
5. Každý z těch listů ve stromu, který není spojen s žádným b-listem, nyní spojíme hranou s jiným listem stromu funkcí

$$a_i : \{f_i + 1, \dots, f_{i+1} - 1\} \longrightarrow \{f_i + 1, \dots, f_{i+1} - 1\}, \quad (3.9)$$

tak, aby byly splněny podmínky

$$\begin{aligned} a_i(x) = y &\Rightarrow (x \neq y \wedge a_i(y) = x), \\ a_i(x_1) = y_1 \wedge a_i(x_2) = y_2 &\Rightarrow x_1 \leq x_2, y_2 \leq y_1 \vee \\ & y_1 \leq x_2, y_2 \leq x_1 \vee \\ & x_2 \leq x_1, y_1 \leq y_2 \vee \\ & x_2 \leq y_1, x_1 \leq y_2. \end{aligned} \quad (3.10)$$

První podmínka pouze říká, že je-li spojen uzel x s uzlem y , je i y spojen s x , navíc v grafu nejsou smyčky. Druhá podmínka zajišťuje, že vzniklý graf bude rovinný.

6. Nyní všechny cesty délky 3, kde oba vnitřní uzly mají valenci 2, nahradíme jedinou hranou a vnitřní uzly odstraníme.

Uvedeným algoritmem získáme graf, který může obsahovat smyčky. Jestliže vezmeme v úvahu jen ty grafy, které nebudou obsahovat smyčky, výsledný graf má požadované vlastnosti PAG.

²Úplný ternární strom je strom, kdy každý uzel má buďto 3 syny nebo žádné syny, a s výjimkou jednoho uzlu právě jednoho předka. Uzel, který nemá předka, se nazývá kořen. Uzel, který nemá syny, je list. Uzel se syny je vnitřní uzel stromu.

Poznámka. Požadavek v bodě 4, aby rozdíl $f_{i+1} - f_i$ byl lichý, je důležitý proto, abychom mohli spojit zbylé listy v bodě 5 po dvojicích a zároveň mohla být zachována rovinnost.

Příklad Demostrujme uvedený postup na příkladě. Uvažujme $b = 6, p = 6$. Situace po prvních třech krocích algoritmu je zobrazena na obrázku 3.7a, kde jsou zobrazeny b-uzly na kružnici délky 6, navíc ke každému b-uzlu je vytvořena kopie (b-listy) a spojena s příslušným b-uzlem. Vnitřní uzly stromu poté budou p-uzly. Kořen stromu je v posledním – šestém b-uzlu. Počet listů stromu je $2p + 1 = 13$.

Dalším krokem algoritmu je spojení hranou každého b-listu s nějakým listem stromu tak, aby byly splněny podmínky v bodě 4. Tento krok je vyznačen na obrázku 3.7b fialovými hranami.

Dále jsou spojeny hranou zbylé listy stromu tak, aby byly splněny podmínky (3.10) v bodě 5 (neporušení rovinnosti grafu). Toto přiřazení je zobrazeno na obrázku 3.7c modrými hranami.

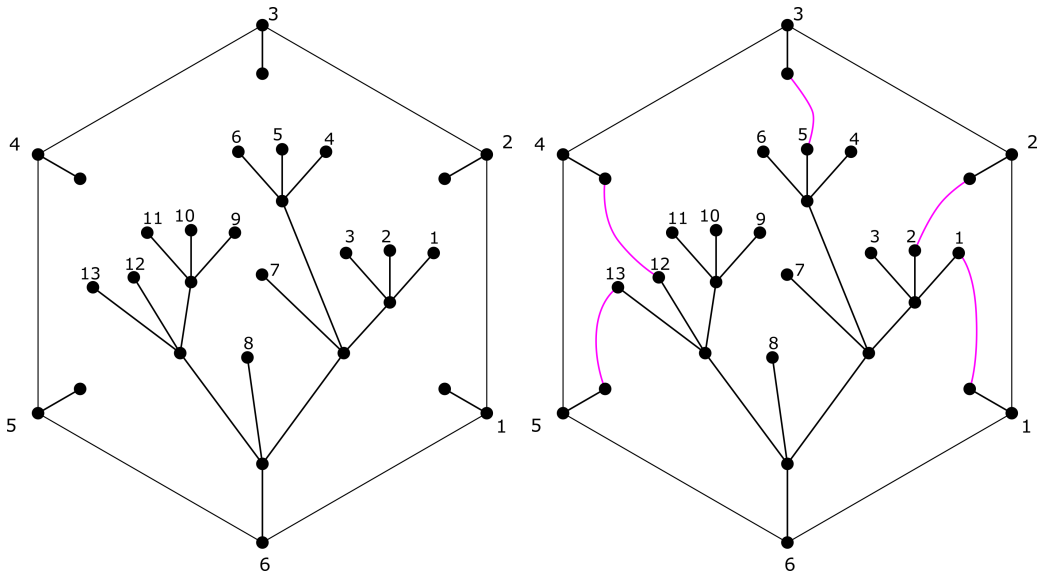
Posledním krokem je nalezení všech cest délky 3, kde oba vnitřní uzly cesty mají valenci 2 (na obrázku 3.7d zobrazeno červenými hranami) a nahrazením každé této cesty jednou hranou. Výsledný graf je pak zobrazen na obrázku 3.8, kde červeně jsou vyznačeny hrany, které vznikly z cest délky 3, ostatní černé hrany jsou původní hrany stromu a kružnice. Ve výsledném PAG vznikly p-uzly z vnitřních uzlů stromu, všechny listy stromu byly odstraněny v posledním kroku.

Poznamenejme, že počet t_p úplných ternárních stromů s p vnitřními uzly vytvářených v bodě 3, je dán následujícím vztahem (o úplných ternárních stromech více v [8]):

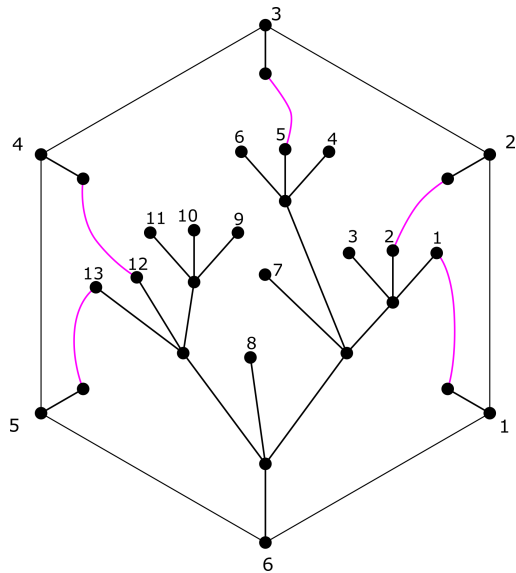
$$t_p = \frac{1}{2p+1} \binom{3p}{p}. \quad (3.11)$$

Počet možných přiřazení b-listů k listům stromu v bodě 4 je omezen číslem $\binom{2p+1}{b-1}$, počet různých spojení hranou zbylých listů ve stromu (bod 5) je omezen číslem $\frac{M!}{2^{M/2}(M/2)!}$, kde $M = 2p - b + 2$. Součinem těchto čísel dostáváme horní odhad pro počet všech možností, které musíme algoritmem prohledat. Ovšem počet grafů sousednosti, které algoritmem vzniknou, je menší. Důvodem je, že různé stromy a různá přiřazení hran v bodech 4 a 5 můžou vést na stejný graf sousednosti.

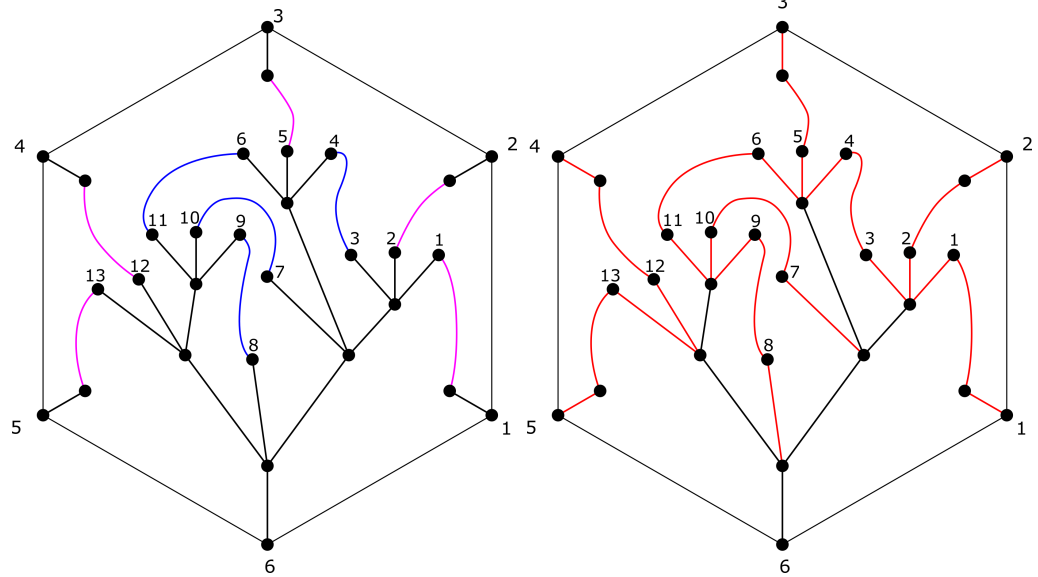
Navíc ne každé přiřazení hran vede k vytvoření grafu sousednosti, důvodem je vznik smyček v posledním kroku algoritmu. Příklad je uveden na obrázku 3.9, kde je zobrazen stejný strom jako na obrázku 3.7a. V tomto případě jsou ale uzly spojeny tak, že odstraněním cest délky 3 (zobrazeny



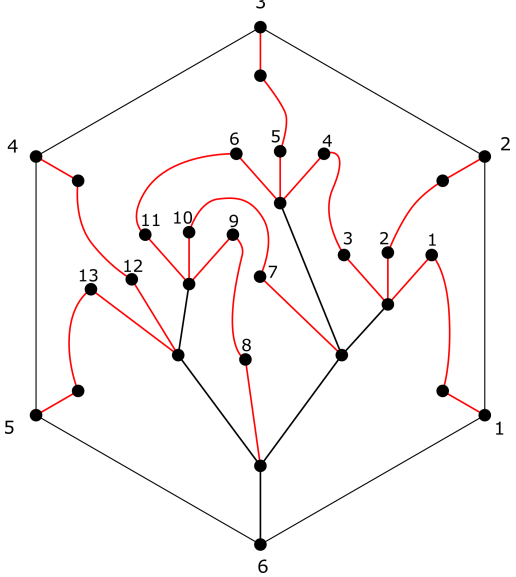
(a) Vytvoření stromu



(b) Spojení b-listů

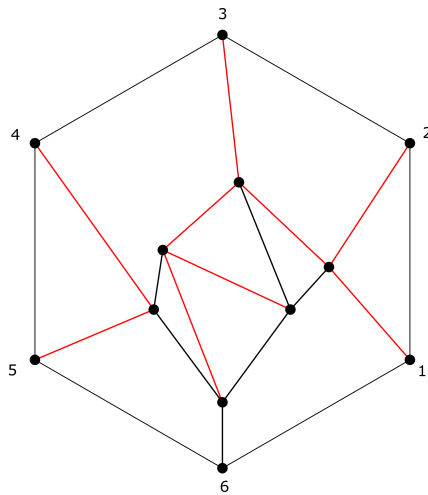


(c) Spojení listů stromu



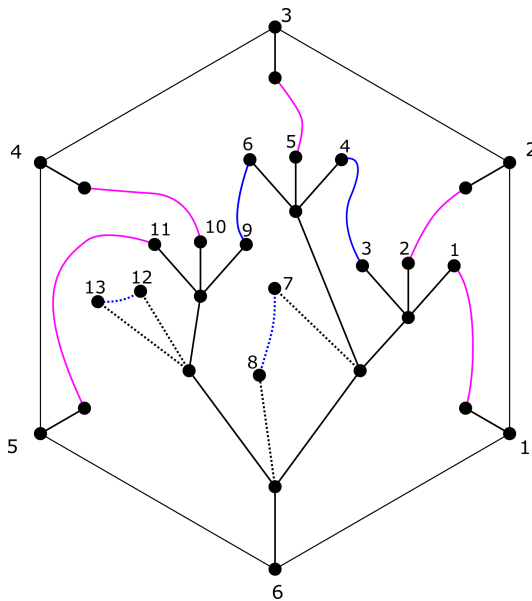
(d) Nalezení cest délky 3

Obrázek 3.7: Generování PAG



Obrázek 3.8: Výsledný PAG

čárkovaně – u listů 7,8 a 12,13) a nahrazením novou cestou v posledním kroku algoritmu, vzniknou smyčky.



Obrázek 3.9: Příklad, kdy nevznikne PAG

Pro příklad $b = 6, p = 6$ je počet všech úplných ternárních stromů, které je třeba prohledat, roven

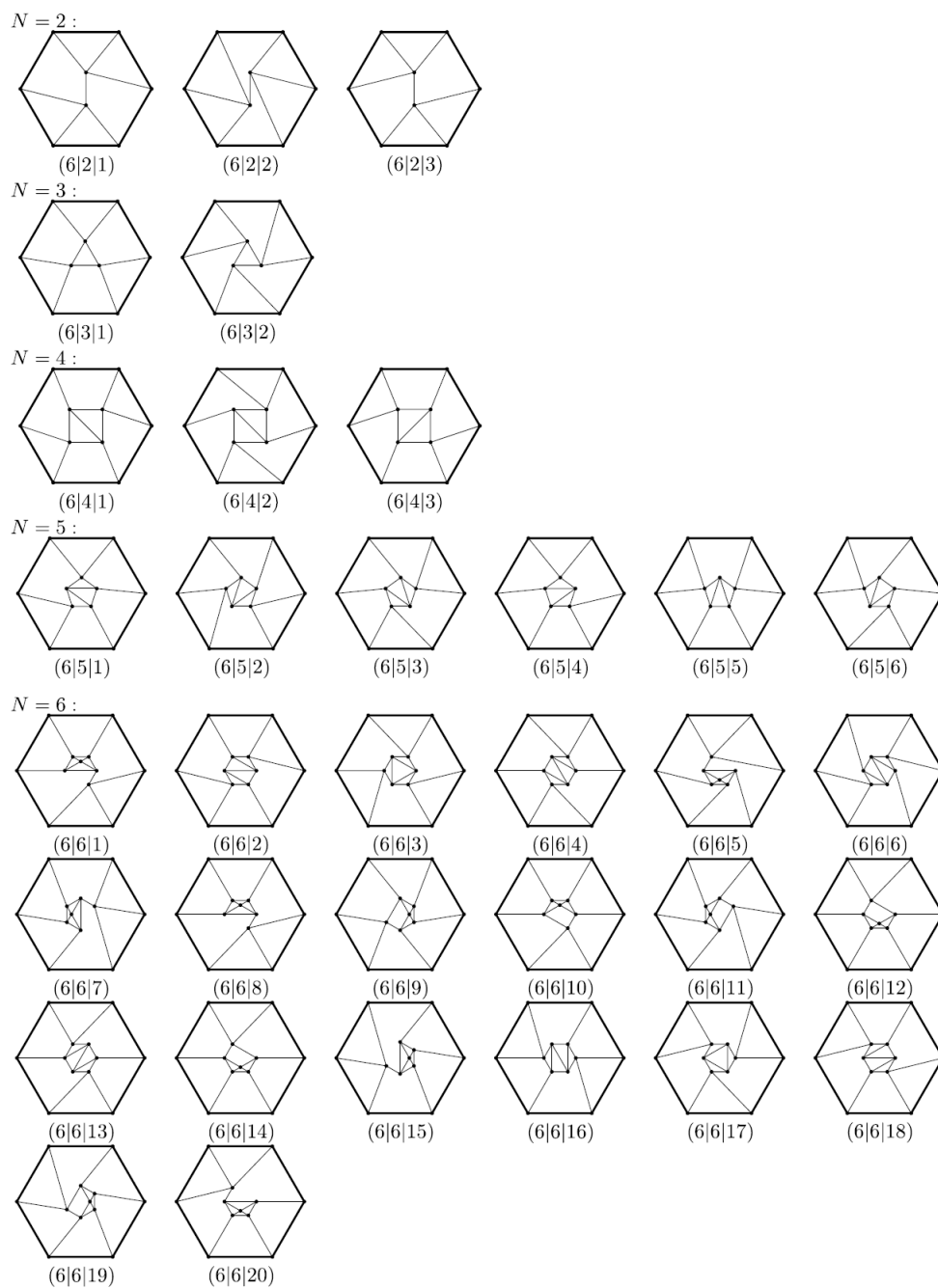
$$t_p = \frac{1}{2p+1} \binom{3p}{p} = \frac{1}{13} \binom{18}{3} = 1428, \quad (3.12)$$

dále maximální možný počet přiřazení uzlů z kroku 4 je $\binom{13}{5} = 1287$, respektive z kroku 5 je $\frac{8!}{2^{8/2}(8/2)!} = 105$. Horní odhad počtu všech možných situací je roven součinu těchto čísel. Ovšem celkový počet grafů sousednosti pro $b = 6, p = 6$ je 20. Tento algoritmus pro nalezení grafů sousednosti je zřejmě vhodný jen pro relativně malá čísla b, p .

Další počty grafů sousednosti v závislosti na číslech počtu segmentů b a počtu patchů p jsou uvedeny v tabulce 3.1. Obrázek 3.10 ukazuje všechny grafy sousednosti pro $b = 6, p = 2, \dots, 6$.

| Počet patchů p | Počet segmentů b | | | | | |
|------------------|--------------------|-----|-------|-------|-------|--------|
| | 2 | 4 | 6 | 8 | 10 | 12 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 3 | 0 | 0 | 0 |
| 3 | 0 | 0 | 2 | 12 | 0 | 0 |
| 4 | 0 | 0 | 3 | 18 | 55 | 0 |
| 5 | 0 | 1 | 6 | 36 | 132 | 273 |
| 6 | 5 | 4 | 20 | 84 | 330 | 910 |
| 7 | 9 | 39 | 66 | 252 | 880 | 2 730 |
| 8 | 17 | 116 | 382 | 825 | 2 698 | 8 191 |
| 9 | 99 | 511 | 1 476 | 3 792 | 8 969 | 26 423 |

Tabulka 3.1: Počet PAG v závislosti na b, p (převzato z [2])



Obrázek 3.10: PAG pro $b = 6, p = 2, \dots, 6$ (převzato z [2])

3.3 Algoritmus multi-patch parametrizace

Vstupem algoritmu bude oblast Ω , jejíž hranice je rozdělena na sudý počet b segmentů c_i . Dále zadáme číslo p_{max} , které určuje maximální počet patchů, na který oblast Ω chceme rozdělit. Budeme prohledávat všechny validní PAG takové, kdy $p \leq p_{max}$. Poté pro každý validní PAG spočítáme řídicí body multi-patch parametrizace řešením optimalizačního problému. Následně vybereme takové řešení, jehož cílová funkce má nejnižší hodnotu a zároveň aby získaná MPP byla regulární.

3.3.1 Vytvoření struktury řídicích bodů

Jak už bylo zmíněno výše, pro jednoduchost předpokládejme, že každý segment je určen stejným vektorem parametrizace a $n+1$ řídicími body. Mějme validní PAG, který reprezentuje MPP, kterou chceme získat. Řídicí body k -tého segmentu označme

$$d_{i,0}^k, \quad i = 0, \dots, n, \quad k = 1, \dots, b. \quad (3.13)$$

Sít každého patche bude tvořena $(n+1) \times (n+1)$ body, které označíme

$$d_{i,j}^k, \quad i, j = 0, \dots, n, \quad k = b+1, \dots, b+p. \quad (3.14)$$

Nyní využijeme graf sousednosti, respektive funkce I, E, N, W, S , které určují strukturu MPP a segmentů. Díky těmto funkcím přiřadíme řídicí body segmentů sousedícím patchům:

$$\begin{aligned} d_{n,n-i}^{I(k)} &= d_{i,0}^k, \text{ jestliže } E(I(k)) = k, \\ d_{i,n}^{I(k)} &= d_{i,0}^k, \text{ jestliže } N(I(k)) = k, \\ d_{0,i}^{I(k)} &= d_{i,0}^k, \text{ jestliže } W(I(k)) = k, \\ d_{n-i,0}^{I(k)} &= d_{i,0}^k, \text{ jestliže } S(I(k)) = k. \end{aligned} \quad (3.15)$$

Vztahy v (3.15) jsou celkem čtyři, neboť patch se segmentem může sousedit pouze jednou ze čtyř stěn. Ale vztahů pro sousednost dvou patchů bude 16, protože je celkem 16 možností, jakými hranami dva patche sousedí:

$$\begin{aligned} d_{n,i}^k &= d_{n-i,1}^{E(k)}, \text{ jestliže } S(E(k)) = k, \\ d_{i,n}^k &= d_{i,1}^{N(k)}, \text{ jestliže } S(N(k)) = k, \\ d_{1,i}^k &= d_{i,1}^{W(k)}, \text{ jestliže } S(W(k)) = k, \\ d_{i,1}^k &= d_{n-i,1}^{S(k)}, \text{ jestliže } S(S(k)) = k. \end{aligned} \quad (3.16)$$

Dalších 12 podmínek je analogických.

Nyní máme svázaný řídicí body hran patchů k segmentům i k ostatním patchům. Dále vložíme souřadnice řídicích bodů $d_{i,j}^k$, kde $i, j = 0, \dots, n$, $k = b + 1, \dots, b + p$ do vektoru \mathbf{d} . Navíc označme \mathbf{d}^k vektor souřadnic řídicích bodů patche k .

Řídicí body $d_{i,j}^k$ nám následně spolu s danými uzlovými vektory určí B-spline parametrizaci patche $\mathbf{g}^{(k)}$:

$$\mathbf{g}^{(k)}(u, v) = \sum_{i=0}^n \sum_{j=0}^n d_{i,j}^k N_i^q(u) N_j^q(v), \quad (3.17)$$

případně NURBS parametrizaci, známe-li váhy bodů:

$$\mathbf{g}^{(k)}(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^n w_{i,j} d_{i,j}^k N_i^q(u) N_j^q(v)}{\sum_{i=0}^n \sum_{j=0}^n w_{i,j} N_i^q(u) N_j^q(v)}. \quad (3.18)$$

3.3.2 Lineární optimalizace

Označme funkcionály

$$\mathcal{Q}_l(\mathbf{d}^k) = \int_{\Omega} \|\mathbf{g}_u^{(k)}\|^2 + \|\mathbf{g}_v^{(k)}\|^2 \, du \, dv, \quad (3.19)$$

$$\mathcal{Q}_u(\mathbf{d}^k) = \int_{\Omega} \|\mathbf{g}_{uu}^{(k)}\|^2 + 2 \|\mathbf{g}_{uv}^{(k)}\|^2 + \|\mathbf{g}_{vv}^{(k)}\|^2 \, du \, dv, \quad (3.20)$$

kde \mathcal{Q}_l zohledňuje délku parametrických křivek hledané parametrizace a \mathcal{Q}_u rovnoměrnost rozložení parametrických křivek. Řešením optimalizačního problému

$$\sum_{k=b+1}^{b+p} \lambda_l \mathcal{Q}_l(\mathbf{d}^k) + \lambda_u \mathcal{Q}_u(\mathbf{d}^k) \rightarrow \min_{\mathbf{d}} \quad (3.21)$$

získáme počáteční řešení souřadnic řídicích bodů patchů, kde váhy λ_l, λ_u jsou nezáporná reálná čísla. Jejich vliv na parametrizaci bude ukázán v kapitole 5, která se věnuje experimentům. Optimalizační problém vede na řešení lineární soustavy rovnic pro neznámé řídicí body. Při výpočtu příkladů v tomto textu byla použita funkce `Minimize` v programu Wolfram Mathematica 10.0.

3.3.3 Nelineární optimalizace

Dále označme

$$\mathcal{Q}_o(\mathbf{d}^k) = \int_{\Omega} \left(\mathbf{g}_u^{(k)} \cdot \mathbf{g}_v^{(k)} \right)^2 \, du \, dv, \quad (3.22)$$

$$\mathcal{Q}_s(\mathbf{d}^k) = \int_{\Omega} \left(\frac{(\mathbf{g}_u^{(k)} \cdot \mathbf{g}_v^{(k)})^2}{(\mathbf{g}_u^{(k)} \cdot \mathbf{g}_u^{(k)}) (\mathbf{g}_v^{(k)} \cdot \mathbf{g}_v^{(k)})} \right)^2 du dv, \quad (3.23)$$

$$\mathcal{Q}_a(\mathbf{d}^k) = \int_{\Omega} \det(\mathbf{g}_u^{(k)}, \mathbf{g}_v^{(k)})^2 du dv, \quad (3.24)$$

kde \mathcal{Q}_o je funkcionál, který zohledňuje ortogonalitu parametrických křivek. \mathcal{Q}_s měří kvadrát kosinu úhlu mezi tečnými vektory parametrických křivek. Rozdíl mezi funkcionály \mathcal{Q}_o a \mathcal{Q}_s je, že \mathcal{Q}_s je na rozdíl od funkcionálu ortogonalitativní vůči změně měřítka. Funkcionál \mathcal{Q}_a měří druhou mocninu determinantu Jacobiho matice. Omezuje překlápění oblasti a podporuje jednotnou velikost prvků.

Potom pro zlepšení a upřesnění polohy řídicích bodů můžeme použít cílovou funkci

$$\begin{aligned} & \sum_{k=b+1}^{b+p} \lambda_l \mathcal{Q}_l(\mathbf{d}^k) + \lambda_u \mathcal{Q}_u(\mathbf{d}^k) + \lambda_o \mathcal{Q}_o(\mathbf{d}^k) + \lambda_s \mathcal{Q}_s(\mathbf{d}^k) \\ & + \lambda_a \mathcal{Q}_a(\mathbf{d}^k) \rightarrow \min_{\mathbf{d}}, \end{aligned} \quad (3.25)$$

kde váhy $\lambda_o, \lambda_s, \lambda_a$ jsou opět nezáporná reálná čísla. Tento optimalizační problém můžeme řešit např. iterační Gaussovo-Newtonovou metodou (viz [2] nebo [4]). V příkladech uvedených v této diplomové práci byla použita funkce `FindMinimum` ve Wolfram Mathematica 10.0.

Vliv vah, respektive funkcionálů na tvar výsledné multi-patch parametrizace je ukázán v kapitole 5. Další výsledky je možné najít například v [2], [4], [7].

Poznámka. V literatuře je možno najít další funkcionály, např. v [4] se namísto \mathcal{Q}_a používá funkcionál excentricity \mathcal{Q}_e

$$\mathcal{Q}_e(\mathbf{d}^k) = \int_{\Omega} \left(\frac{\mathbf{g}_u^{(k)} \mathbf{g}_{uu}^{(k)}}{\mathbf{g}_u^{(k)} \mathbf{g}_u^{(k)}} \right)^2 + \left(\frac{\mathbf{g}_v^{(k)} \mathbf{g}_{vv}^{(k)}}{\mathbf{g}_v^{(k)} \mathbf{g}_v^{(k)}} \right)^2 du dv.$$

4 Parametrizace objemu

Článek [2], ze kterého vycházela předchozí kapitola, se věnuje pouze multi-patch parametrizaci 2D oblasti zadané pomocí hraničních B-spline nebo NURBS křivek. Cílem této kapitoly bude zobecnění předchozí situace, tedy analýza multi-patch parametrizace pro objem. Některé poznatky budou přirozeně zobecněné, ovšem např. vyjádření struktury pomocí grafu sousednosti bude odlišné.

4.1 3D multi-patch parametrizace

Mějme jednoduše souvislý objem Ω . Jeho hranice je parametrizována pomocí b B-spline nebo NURBS ploch

$$c_i : [0, 1]^2 \rightarrow \mathbb{R}^3, \quad i = 1, 2, \dots, b. \quad (4.1)$$

Opět budeme hraniční plochy nazývat segmenty a budou pro ně platit podobné vlastnosti jako v případě 2D oblasti – každá hraniční křivka jednoho segmentu se bude překrývat právě s jednou hraniční křivkou jiného segmentu, tj. $\forall i \exists! j \neq i$ takové, že platí jedna z následujících osmi rovností.

$$\begin{aligned} c_i(0, t) &= c_j(0, t) && \text{nebo} \\ c_i(0, t) &= c_j(0, 1 - t) && \text{nebo} \\ c_i(0, t) &= c_j(1, t) && \text{nebo} \\ c_i(0, t) &= c_j(1, 1 - t) && \text{nebo} \\ c_i(0, t) &= c_j(t, 0) && \text{nebo} \\ c_i(0, t) &= c_j(1 - t, 0) && \text{nebo} \\ c_i(0, t) &= c_j(t, 1) && \text{nebo} \\ c_i(0, t) &= c_j(1 - t, 1). && \end{aligned} \quad (4.2)$$

Protože každý segment c_i má čtyři hraniční křivky, existují další tři různé segmenty c_k, c_l, c_m , pro něž platí analogické vztahy ke vztahům (4.2) (pro další hraniční křivky segmentu c_i). Zároveň se dva segmenty nemůžou překrývat jinde, než v těchto hraničních křivkách, platí tedy

$$\begin{aligned} \forall i, j, \quad \forall u_1, v_1, u_2, v_2 \in (0, 1) : c_i(u_1, v_1) = c_j(u_2, v_2) \Rightarrow \\ (i = j \wedge u_1 = u_2 \wedge v_1 = v_2) \end{aligned} \quad (4.3)$$

Pokud dva segmenty, které mají společnou hranu, svírají v každém bodě

hrany vnitřní úhel menší než π , budeme hovořit o konvexní hraně, v opačném případě o nekonvexní.

Objem Ω rozdělíme na menší disjunktní objemy – patche:

$$\mathbf{g}^{(j)} : [0, 1]^3 \rightarrow \mathbb{R}^3, \quad j = 1, \dots, p. \quad (4.4)$$

Každý patch má 6 stěn $g_i^{(j)}, i = 1, \dots, 6$:

$$\begin{aligned} g_1^{(j)} &: [0, 1]^2 \rightarrow \mathbb{R}^3, & (u, v) &\mapsto \mathbf{g}^{(j)}(u, v, 0), \\ g_2^{(j)} &: [0, 1]^2 \rightarrow \mathbb{R}^3, & (u, v) &\mapsto \mathbf{g}^{(j)}(1, u, v), \\ g_3^{(j)} &: [0, 1]^2 \rightarrow \mathbb{R}^3, & (u, v) &\mapsto \mathbf{g}^{(j)}(u, 1, v), \\ g_4^{(j)} &: [0, 1]^2 \rightarrow \mathbb{R}^3, & (u, v) &\mapsto \mathbf{g}^{(j)}(0, u, v), \\ g_5^{(j)} &: [0, 1]^2 \rightarrow \mathbb{R}^3, & (u, v) &\mapsto \mathbf{g}^{(j)}(u, 0, v), \\ g_6^{(j)} &: [0, 1]^2 \rightarrow \mathbb{R}^3, & (u, v) &\mapsto \mathbf{g}^{(j)}(u, v, 1). \end{aligned} \quad (4.5)$$

Uvažujeme opět takové patche, které jsou regulární, tedy všechny hrany patche jsou konvexní. Parametrizaci, kdy je celý objem parametrizován pomocí patchů, budeme nazývat multi-patch parametrizací. Obdobně zavedeme i množinu všech ploch

$$S = \{g_l^{(j)} \mid l = 1, \dots, 6, j = 1, \dots, p\} \cup \{c_i \mid i = 1, \dots, b\}. \quad (4.6)$$

Všimněme si, že tentokrát jsme neuvažovali orientaci jako ve 2D případě, tedy relace \sim na množině S , která by jednoduše popsala plochy $s, s' \in S$, které se dotýkají ve všech bodech, bude definovaná následovně:

$$\begin{aligned} s \sim s' \Leftrightarrow & \quad s(u, v) = s'(u, v) && \text{nebo} \\ & \quad s(u, v) = s'(1 - u, v) && \text{nebo} \\ & \quad s(u, v) = s'(1 - u, 1 - v) && \text{nebo} \\ & \quad s(u, v) = s'(u, 1 - v) && \text{nebo} \\ & \quad s(u, v) = s'(v, u) && \text{nebo} \\ & \quad s(u, v) = s'(1 - v, u) && \text{nebo} \\ & \quad s(u, v) = s'(1 - v, 1 - u) && \text{nebo} \\ & \quad s(u, v) = s'(v, 1 - u). \end{aligned} \quad (4.7)$$

Relace je stejně jako ve 2D případě symetrická, ale navíc je i reflexivní.

V relaci \sim na množině S jsou dvojice ploch, a protože je počet hraničních ploch každého patche sudý, musí být i počet segmentů sudý.

Protože jsme pro objem zavedli pojmy stejně jako pro 2D oblast, i definice validní multi-patch parametrizace pro objem je analogická.

Definice 4 (Validní multi-patch parametrizace objemu). *Množinu patchů $\{\mathbf{g}^{(j)}\}_{j=1}^p$ nazveme validní multi-patch parametrizací (validní MPP), jestliže platí:*

- (i) všechny patche jsou regulární,
- (ii) $\text{int}(\mathbf{g}^{(i)}) \cap \text{int}(\mathbf{g}^{(j)}) = \emptyset, \quad \forall i \neq j$
- (iii) žádné dvě hraniční plochy jednoho patche spolu nejsou v relaci \sim ,
- (iv) $\forall s \in S \exists! s' \in S : s \sim s'$.

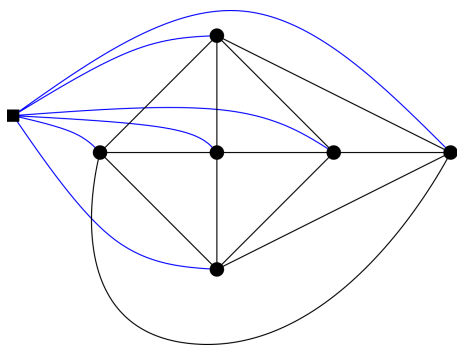
4.2 Graf susednosti

Stejným způsobem jako pro rovinnou oblast, můžeme vytvořit ke každé objemové multi-patch parametrizaci graf susednosti. Patche budou reprezentovat p-uzly, segmenty b-uzly a budeme je spojovat hranami stejným způsobem; pouze s tím rozdílem, že tentokrát kvůli reflexivitě relace \sim musíme zavést podmínku, že nebudeme hranou spojovat uzel sám se sebou, tj. nebudeme vytvářet smyčky. Můžeme odvodit tyto vlastnosti grafu susednosti:

1. Počet uzlů v grafu je $b + p$,
2. každý b-uzel má valenci 5,
3. každý p-uzel má valenci 6,
4. v grafu nejsou smyčky,
5. podgraf indukovaný všemi b-uzly je souvislý,
6. podgraf indukovaný všemi p-uzly je souvislý.

Vlastnost 1 je zřejmá a plyne z konstrukce grafu susednosti. Každý segment susedí právě se čtyřmi dalšími segmenty a právě jedním patchem, tedy b-uzel má valenci 5, každý patch má 6 hraničních ploch, proto každý p-uzel má 6 susedů. Poslední dvě vlastnosti plynou ze souvislosti objemu.

Nyní už mezi vlastnostmi není uvedena rovinnost grafu, která byla důležitá v algoritmu pro nalezení všech grafů susednosti. Abychom dokázali, že graf není obecně rovinný, postačí nám příklad pro takovou multi-patch parametrizaci, kde $b = 6, p = 1$. Příkladem takového objemu může být např. krychle. Graf susednosti je zobrazen na obrázku 4.1, b-uzly jsou vyznačeny symbolem \bullet , p-uzel symbolem \blacksquare .



Obrázek 4.1: PAG pro $b = 6, p = 1$

Nutná podmínka rovinného grafu říká, že jestliže $G = (V, E)$ je rovinný graf, pak $|E| \leq 3|V| - 6$, kde V jsou uzly a E jsou hrany. Tedy pro příklad na obrázku 4.1 platí $|V| = 7, |E| = 18$. Po dosazení dostáváme $18 \leq 15$, tj. nutná podmínka je porušena.

Uvedli jsme jednoduchý příklad, kdy graf sousednosti není rovinný. Zda je každý graf sousednosti ve 3D případě nerovinný, nelze určit pomocí této nutné podmínky, a tedy pro důkaz budeme potřebovat jiné tvrzení.

Věta 1 (Wagnerova). *Konečný graf je rovinný právě tehdy, když neobsahuje K_5 nebo $K_{3,3}$ jako minor.*

Důkaz. viz např. [11]. □

Definice 5 (Minor grafu). *Minor H grafu G je takový graf, který může vzniknout z nějakého podgrafu G kontrakcemi některých hran.*

Kontrakcí hrany $e = \{u, v\}$ v grafu rozumíme odebrání hrany e a identifikaci uzlů u, v .

Věta 2. *Žádný graf sousednosti objemové multi-patch parametrizace není rovinný.*

Důkaz. Protože Wagnerova věta je ve formě ekvivalence, musí graf na obrázku 4.1 obsahovat K_5 nebo $K_{3,3}$ jako minor. Stačí tedy ukázat, že každý graf sousednosti objemové multi-patch parametrizace obsahuje jako minor uvedený graf.

Označme graf sousednosti G . Vyberme p -uzel reprezentující nějaký patch a označme ho u_0 . Tento uzel sousedí právě s šesti dalšími uzly, označme je u_1, \dots, u_6 . Jestliže 2 sousedi vybraného patche (buďto segment nebo patch) mají v objemu Ω společnou hranu, pro jejich reprezentanty u_i, u_j v grafu G najdeme cestu, která neprochází uzlem u_0 . Taková cesta existuje, protože graf $G \setminus u_0$ je stále souvislý. Takovou cestu najdeme pro každou dvojici uzlů u_i, u_j ($i, j = 1, \dots, 6, i \neq j$).

Máme tedy 12 cest (pro každou hranu patche jednu). Z grafu G vytvoříme podgraf G_1 takový, který bude obsahovat uzly u_0, \dots, u_6 , hrany incidentní s u_0 a všechny uzly a hrany na nalezených cestách. Z tohoto grafu G_1 vytvoříme minor G_2 tak, že provedeme kontrakci hran na všechny hrany na cestě z u_i do u_j ($i, j = 1, \dots, 6, i \neq j$) (kontrakci můžeme provést opakovaně, dokud mezi touto dvojicí uzlů existuje jiný uzel). Tímto postupem dostaneme graf z obrázku 4.1. Tento graf ale obsahuje K_5 nebo $K_{3,3}$ a tudíž původní graf G nemůže být rovinný. \square

Největším problémem při zobecňování do 3D je skutečnost, že z grafu sousednosti nemáme jasnou představu o vzhledu objemu. Z rovinného grafu sousednosti pro 2D případ jsme sice nedokázali mít konkrétní představu o geometrii oblasti a patchů, ale struktura byla zřejmá. Dokázali jsme z grafu sousednosti určit jednoznačně rozpoložení patchů (dokážeme přiřadit funkce I, E, N, W, S), i když přesnou geometrii neznáme. Například určíme-li funkci E , ostatní už dokážeme ze struktury dourčit.

Oproti tomu z grafu sousednosti pro objem není zřejmá struktura, tj. kdybychom určovali opět funkce, které přiřazují každému prvku MPP prvek jiný (dohromady 7 funkcí, např. I, E, N, W, S, F, B), nedokážeme pouze z grafu určit, jakým sousedům kterou funkci přiřadit. Tento problém může být zřejmý už z jednoduchého příkladu grafu sousednosti na obrázku 4.1. Intuitivně bychom dokázali říct, že se jedná např. o „zdeformovanou krychli“, ovšem z grafu není zřejmé, který segment je od patche „nahore“, „dole“, „vpředu“, „vpravo“ apod.

I přes tyto problémy se pokusíme sestavit algoritmus pro generování grafů sousednosti a následně navrhnout řešení problému nejednoznačné grafové struktury.

4.2.1 Generování grafu sousednosti

Mějme dán jednoduše souvislý objem Ω reprezentovaný pomocí sudého počtu b segmentů c_i a zvolme číslo p vyjadřující počet patchů, na který chceme objem Ω rozdělit. Cílem této kapitoly je nalezení všech grafů sousednosti, které budou reprezentovat různé struktury hledané multi-patch parametrizace.

Nabízí se přímé zobecnění algoritmu v kapitole 3.2.1 pomocí úplných stromů. Protože každý p -uzel má valenci 6, využili bychom analogii k úplným ternárním stromům, tedy úplné 5-ární stromy. Ale protože žádný graf sousednosti multi-patch parametrizace objemu není rovinný, ztratili jsme oproti 2D případu důležitou vlastnost, které se využívalo v bodě 5 algoritmu – konkrétně podmínky (3.10) by ve 3D případě neměly smysl. To ale

znamená, že bez nějakých podobných podmínek bychom v tomto kroku algoritmu kombinovali všechny možnosti grafu na p vrcholech a využití stromu by tak postrádalo smysl.

Alternativou může být využití grafové posloupnosti¹ a databáze grafů (pro potřeby této práce byla využita funkce `GraphData` v softwaru Wolfram Mathematica 10.0).

Postup generování PAG

Vstupem algoritmu je množina segmentů, které tvoří hranici objemu Ω a číslo p , které určuje počet patchů. Podgraf indukovaný b-uzly (graf sousednosti hranice – označme ho jako b-graf) výsledného PAG už tedy máme – struktura hraničních ploch se nemění. Budeme hledat podgraf grafu sousednosti, který je indukován p-uzly (označme ho p-graf). Pro tento podgraf musí platit:

- je souvislý,
- neobsahuje smyčky
- uzlový stupeň je nejvýše 6, nejméně 1,
- neobsahuje kliku na 5 uzlech.

První tři vlastnosti přímo vyplývají z vlastností grafu sousednosti. Poslední pak plyne z geometrie MPP – není možné vytvořit validní MPP takovou, aby existovalo 5 patchů, z nichž každý sousedí se všemi ostatními. Kliku na 4 uzlech existovat může – 4 patche, kde každý sousedí s každým, musí mít společný vrchol.

Dále víme, že uzly p-grafu budeme chtít spojit hranami s b-uzly. Jestliže označíme (d_1, d_2, \dots, d_p) grafovou posloupnost stupňů uzlů p-grafu, potom součet stupňů hledaného grafu reprezentující vnitřek bude $\sum_{i=1}^p d_i = 6p - b$. Zároveň víme-li, že stupeň každého uzlu je nejvýše 6, hledáme tuto nerostoucí posloupnost čísel (d_1, d_2, \dots, d_p) pomocí následujících podmínek:

$$\sum_{i=1}^p (6 - d_i) = b, \quad \text{kde } 6 \geq d_1 \geq d_2 \geq \dots \geq d_p \geq 1. \quad (4.8)$$

Tato posloupnost ještě nemusí být grafová. Použitím následující věty ale můžeme ze všech nalezených posloupností vybrat takové, které grafové jsou.

Věta 3 (Erdősova–Gallaiova). *Posloupnost celých nezáporných čísel $d_1 \geq d_2 \geq \dots \geq d_n$ reprezentuje grafovou posloupnost konečného grafu bez smyček*

¹Grafová posloupnost je posloupnost stupňů uzlů grafu.

a násobných hran na n uzlech právě tehdy, když $d_1 + \dots + d_n$ je sudé a zároveň platí

$$\sum_{i=1}^k d_i \leq k(k-1) + \sum_{i=k+1}^n \min(d_i, k) \quad \forall k = 1, \dots, n. \quad (4.9)$$

Důkaz. Např. v [12] □

První podmínka ve větě 3 je automaticky splněna, neboť posloupnost jsme získávali ze vztahu $\sum_{i=1}^p d_i = 6p - b$. Víme, že b je sudé, tedy součet stupňů je také sudý.

Ze všech posloupností získaných ze vztahu (4.8) můžeme jednoduše pomocí vztahu (4.9) vybrat grafové posloupnosti. Sestrojením všech takových grafů, které budou mít získané grafové posloupnosti, dostaneme podgraf grafu sousednosti indukovaný p -uzly.

Možný postup pro získání tohoto podgrafu je využití funkce `GraphData["Connected", p]` v softwaru Wolfram Mathematica. Tato funkce najde všechny souvislé grafy na p uzlech. Následně jednoduchou podmínkou můžeme vybrat pouze takové grafy, které mají požadovanou grafovou posloupnost.

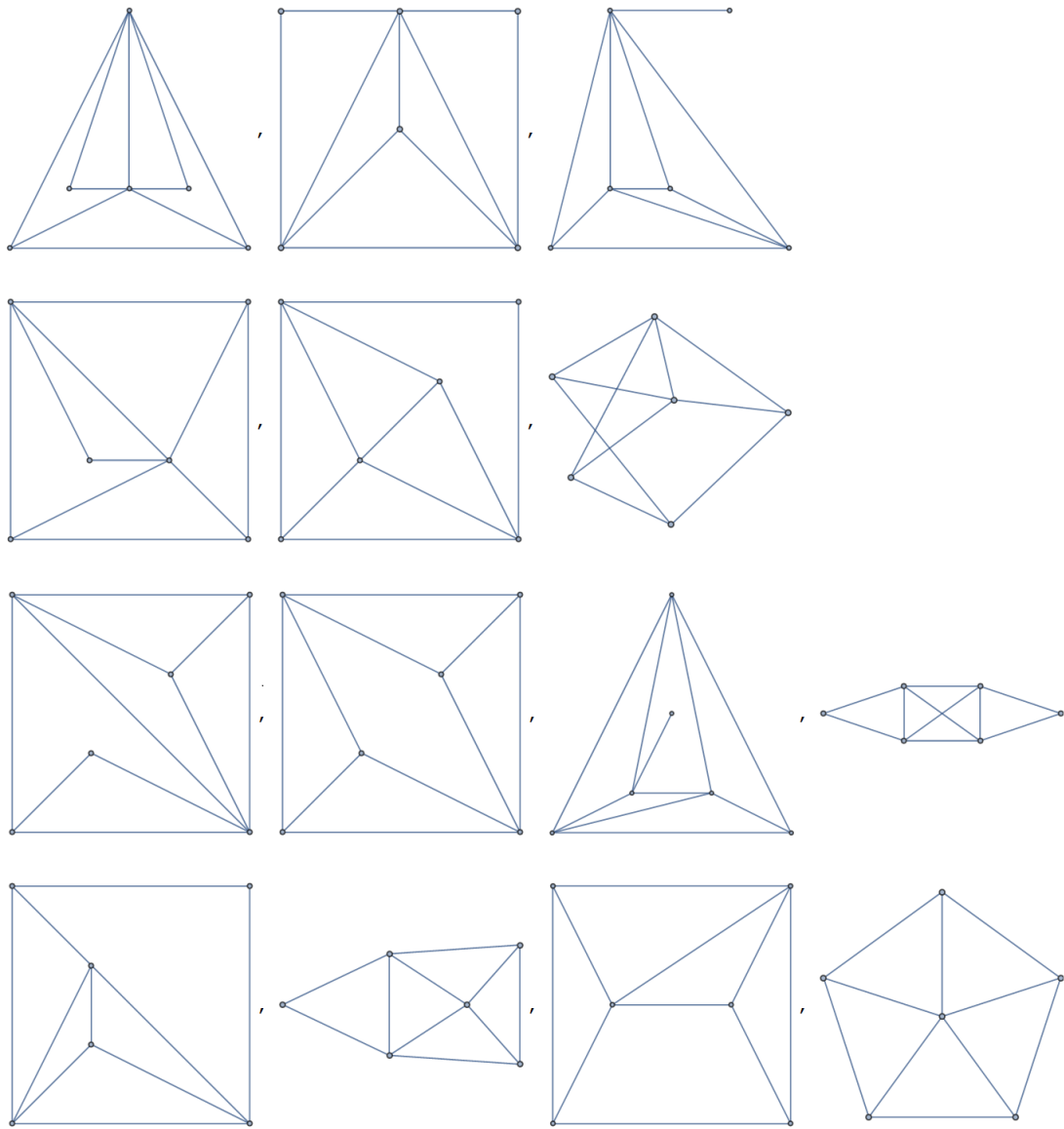
Příklady vnitřních grafů sousednosti pro $b = 16, p = 6$ jsou uvedeny na obrázku 4.2.

Nevýhodou tohoto postupu je, že jsme vnitřní graf generovali nezávisle na struktuře hraničních ploch, uvažovali jsme pouze jejich počet. Musíme tedy ještě vyřešit, jakým způsobem spojit daný graf sousednosti segmentů s vygenerovanými grafy tak, aby výsledný graf sousednosti dával geometricky smysl.

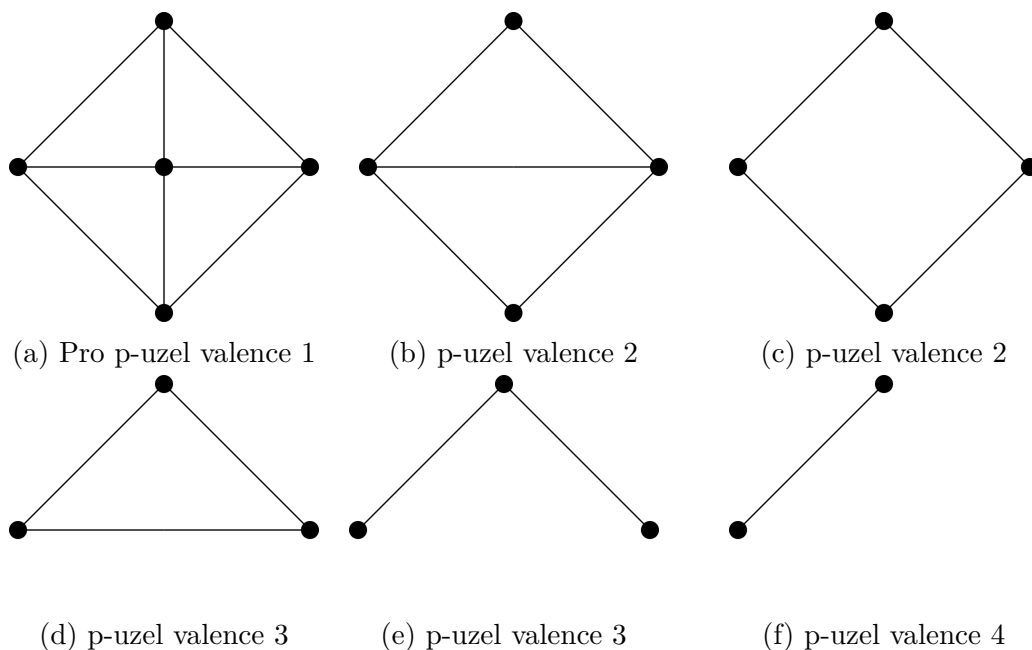
4.2.2 Propojení b-grafu s p-grafem

Tato kapitola navrhne postup pro spojení vygenerovaného p -grafu s b -grafem sousednosti segmentů. V b -grafu budeme hledat podgrafy a spojovat je s uzly z p -grafu.

Bude-li v p -grafu existovat uzel valence 1, tj. existuje takový patch, který sousedí právě s jedním patchem a pěti segmenty, musíme tento uzel spojit hranami s pěti uzly b -grafu. Těchto pět uzlů musí tvořit podgraf uvedený na obrázku 4.3a, aby výsledný PAG mohl dávat geometricky smysl. Jestliže takový podgraf v b -grafu neexistuje, pak s uvedeným p -grafem nemůže tvořit graf sousednosti a graf sousednosti sestrojíme pomocí jiného vygenerovaného p -grafu.



Obrázek 4.2: Vnitřní grafy sousednosti pro $b = 16, p = 6$

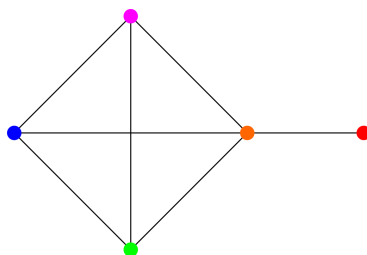


Obrázek 4.3: Podgrafy b-grafu pro spojení s p-uzlem

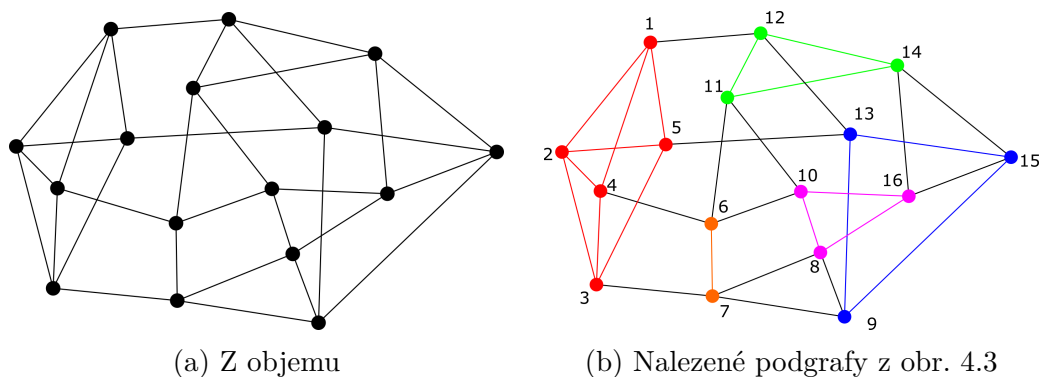
Analogicky můžeme pokračovat dál – jestliže v p-grafu existuje uzel valence 2, spojíme tento uzel hranami s podgrafem b-grafu na obrázku 4.3b nebo 4.3c, p-uzel stupně 3 tohoto grafu spojíme hranami s podgrafem zobrazeným na obrázku 4.3d nebo 4.3e. Dále p-uzel stupně 4 spojíme hranami s podgrafem b-grafu, který tvoří 2 uzly spojené hranou (obrázek 4.3f) a konečně p-uzel stupně 5 s nějakým dosud ještě nespojeným uzlem. Kdybychom v nějakém kroku již nemohli najít požadovaný podgraf v b-grafu, vrátíme se na předchozí krok, najdeme jiné podgrafy a postup opakujeme. Tímto postupem jsme všechny uzly b-grafu spojili s p-grafem.

Příklad Ukažme tento postup na příkladu. Mějme nějaký objem, který je reprezentován šestnácti hraničními B-spline plochami, jeho b-graf je zobrazen na obrázku 4.5a. Zvolme $p = 5$ a uvažujme konkrétní vygenerovaný p-graf zobrazený na obrázku 4.4. V tomto grafu je jeden uzel stupně 1, tedy víme, že tento uzel buďto spojíme s podgrafem b-grafu na obrázku 4.3a, a nebo tento p-graf nebudeme moci použít. Protože takový podgraf existuje (na obrázku 4.5b vyznačen červeně), spojíme tento uzel p-grafu se všemi červenými uzly b-grafu a pokračujeme dále. Žádný uzel stupně 2 v p-grafu není, grafy z obrázků 4.3b a 4.3c v b-grafu nehledáme. Poté najdeme 3 trojúhelníky nebo cesty délky 3 pro uzly p-grafu stupně 3, opět vyznačeny v b-grafu příslušnými barvami. Uzel stupně 4 musíme spojit hranami s podgrafem na obrázku 4.3f. Všechny takto nalezené podgrafy jsou zobrazeny v

obrázku 4.5b. Konkrétní objem, který je reprezentován tímto b-grafem, je v kapitole 5.2, která se věnuje experimentům.



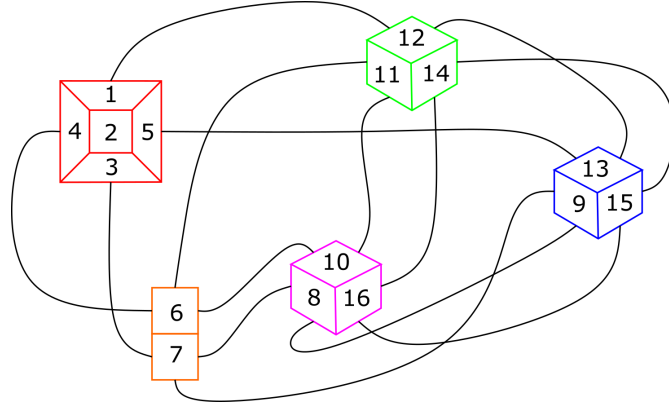
Obrázek 4.4: p-graf



Obrázek 4.5: B-graf

Poznamenejme, že stejně jako při generování grafu sousednosti ve 2D případě, ne vždy vede uvedený postup ke grafu sousednosti nějaké multi-patch parametrizace. Navíc není na první pohled zřejmé, jestli má vygenerovaný graf vzhledem k segmentům geometricky smysl. Efektivnější hledání grafů sousednosti může být předmětem další práce.

Návrhem pro zlepšení orientace v grafu sousednosti by mohla být kombinace b-grafu se segmenty. Jak už jsme uvedli, z b-grafu, respektive z grafu sousednosti není zřejmá struktura hraničních ploch (ani patchů uvnitř), ale použijeme-li b-graf s nalezenými podgrafy a každý uzel budeme reprezentovat čtyřúhelníkem, můžeme zachytit i strukturu a napojení segmentů. Příkladem může být uvedený graf na obrázku 4.5b a následující graf (obrázek 4.6).



Obrázek 4.6: B-graf s informací o způsobu propojení segmentů

4.3 Algoritmus MPP

Podobným postupem jako v kapitole 3.3.1 svážeme řídicí body $d_{i,j,k}^l$ (kde $i, j, k = 0, \dots, n$, $l = b + 1, \dots, b + p$) dvou sousedních patchů (nebo patche a segmentu) a tyto souřadnice vložíme do vektoru \mathbf{d} . Označme \mathbf{d}^l vektor souřadnic řídicích bodů patche l . Obdobně jako ve 2D parametrizujeme patch $\mathbf{g}^{(l)}$ jako B-spline objekt

$$\mathbf{g}^{(l)}(u, v, w) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n d_{i,j,k}^l N_i^q(u) N_j^q(v) N_k^q(w), \quad (4.10)$$

nebo NURBS objekt s váhami $\omega_{i,j,k}$ v bodech $d_{i,j,k}$:

$$\mathbf{g}^{(l)}(u, v, w) = \frac{\sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n \omega_{i,j,k} d_{i,j,k}^l N_i^q(u) N_j^q(v) N_k^q(w)}{\sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n \omega_{i,j,k} N_i^q(u) N_j^q(v) N_k^q(w)}. \quad (4.11)$$

Poznámka. Kvůli složitější struktuře a chybějící orientaci není přiřazení řídicích bodů sousedních patchů detailně algoritmicky řešené.

Dále můžeme zobecnit funkcionály \mathcal{Q}_l , \mathcal{Q}_u z kapitoly 3.3.2 a dostáváme:

$$\mathcal{Q}_l(\mathbf{d}^k) = \int_{\Omega} \|\mathbf{g}_u^{(k)}\|^2 + \|\mathbf{g}_v^{(k)}\|^2 + \|\mathbf{g}_w^{(k)}\|^2 du dv dw, \quad (4.12)$$

$$\begin{aligned} \mathcal{Q}_u(\mathbf{d}^k) = \int_{\Omega} & \|\mathbf{g}_{uu}^{(k)}\|^2 + \|\mathbf{g}_{ww}^{(k)}\|^2 + \|\mathbf{g}_{vv}^{(k)}\|^2 + 2\|\mathbf{g}_{uv}^{(k)}\|^2 \\ & + 2\|\mathbf{g}_{uw}^{(k)}\|^2 + 2\|\mathbf{g}_{vw}^{(k)}\|^2 du dv dw, \end{aligned} \quad (4.13)$$

a řešíme optimalizační problém

$$\sum_{k=b+1}^{b+p} \lambda_l \mathcal{Q}_l(\mathbf{d}^k) + \lambda_u \mathcal{Q}_u(\mathbf{d}^k) \rightarrow \min_{\mathbf{d}}. \quad (4.14)$$

Nalezené souřadnice řídicích bodů ještě upřesníme řešením nelineárního problému, ovšem funkcionály \mathcal{Q}_s , \mathcal{Q}_a (vztah (3.23) a (3.24)) nemá smysl zobecňovat do 3D případu (další informace o funkcionálech v [7]). Řešíme tedy problém

$$\sum_{k=b+1}^{b+p} \lambda_l \mathcal{Q}_l(\mathbf{d}^k) + \lambda_u \mathcal{Q}_u(\mathbf{d}^k) + \lambda_o \mathcal{Q}_o(\mathbf{d}^k) \rightarrow \min_{\mathbf{d}}, \quad (4.15)$$

kde

$$\begin{aligned} \mathcal{Q}_o(\mathbf{d}^k) = \int_{\Omega} & \left(\mathbf{g}_u^{(k)} \cdot \mathbf{g}_v^{(k)} \right)^2 + \left(\mathbf{g}_u^{(k)} \cdot \mathbf{g}_w^{(k)} \right)^2 \\ & + \left(\mathbf{g}_v^{(k)} \cdot \mathbf{g}_w^{(k)} \right)^2 du dv dw, \end{aligned} \quad (4.16)$$

5 Experimenty

Numerické experimenty pro tuto práci byly prováděny v softwaru Wolfram Mathematica 10.0. Problematika generování grafů sousednosti pro 2D oblast nebyla algoritmicky řešena, byl využit seznam grafů sousednosti z obrázku 3.10.

Pro řešení integrálů bylo využito Gaussovo kvadrurní pravidlo, tj. aproximace integrálu sumou:

$$\int_a^b f(x) dx \approx \sum_{i=1}^n w_i f(x_i), \quad (5.1)$$

kde w_i jsou váhy v bodech $x_i \in [a, b]$. Jestliže $f(x)$ je polynom řádu $2n - 1$ a méně, nastává ve vztahu (5.1) rovnost (více např. viz [13]). Pro získání těchto bodů a vah v programu Wolfram Mathematica můžeme použít funkci `GaussianQuadratureWeights` v balíčce `NumericalDifferentialEquationAnalysis`.

Například integrál Q_l pro rovinnou oblast Ω aproximujeme pomocí následujícího vztahu (a ostatní integrály vyjádříme analogicky):

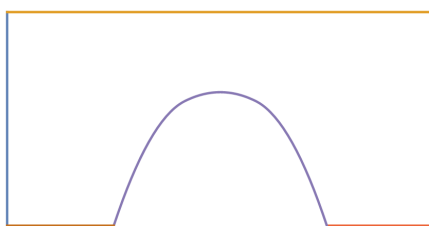
$$\int_{\Omega} \|\mathbf{g}_u^{(k)}(u, v)\|^2 + \|\mathbf{g}_v^{(k)}(u, v)\|^2 du dv \approx \sum_{i=1}^n \sum_{j=1}^n w_i w_j \left(\|\mathbf{g}_u^{(k)}(u_i, v_j)\|^2 + \|\mathbf{g}_v^{(k)}(u_i, v_j)\|^2 \right). \quad (5.2)$$

Dále poznamenejme, že pro ověřování validity patchů pomocí jakobiánu počítáme jakobián v konečně mnoha bodech patche.

5.1 2D oblast

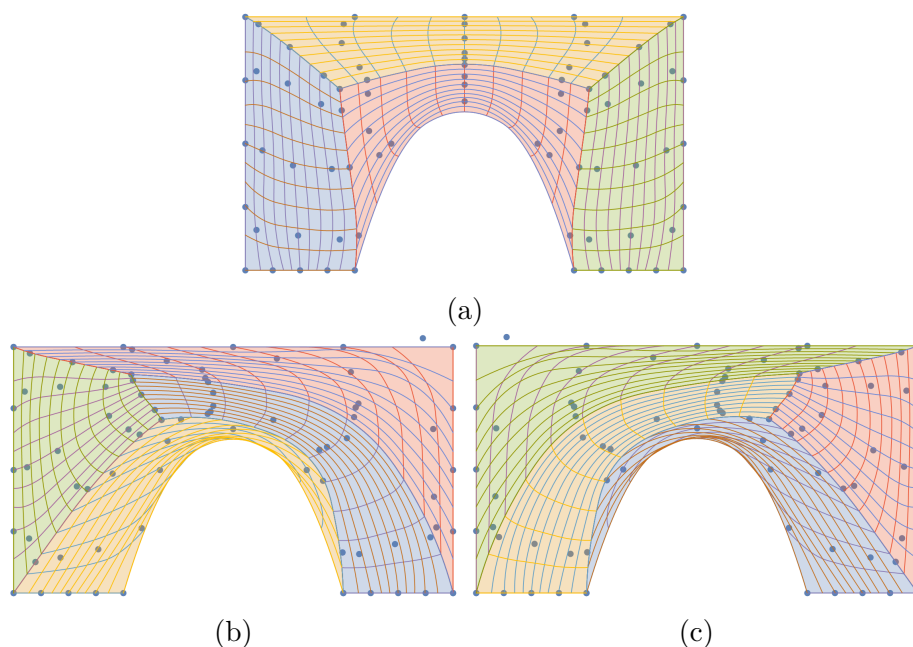
Uvažujme oblast na obrázku 5.1 reprezentovanou šesti B-spline segmenty, tedy $b = 6$. Každá B-spline křivka je učena pěti řídicími body a neperiodickým uniformním vektorem parametrizace $\mathbf{T} = (0, 0, 0, 1/3, 2/3, 1, 1, 1)$ (tj. stupeň křivky je 2).

Zvolme $p = 4$, tedy rozdělíme tuto oblast na 4 disjunktní podoblasti. Protože každá hraniční křivka oblasti je určena pěti body, každý patch bude určen 25 řídicími body. Možné grafy sousednosti, které mohou reprezentovat strukturu patchů jsou zobrazeny na obrázku 3.10 pod označením (6|4|1), (6|4|2) a (6|4|3). Obrázek 5.2 ilustruje výsledné multi-patch parametrizace



Obrázek 5.1: Oblast s šesti segmenty

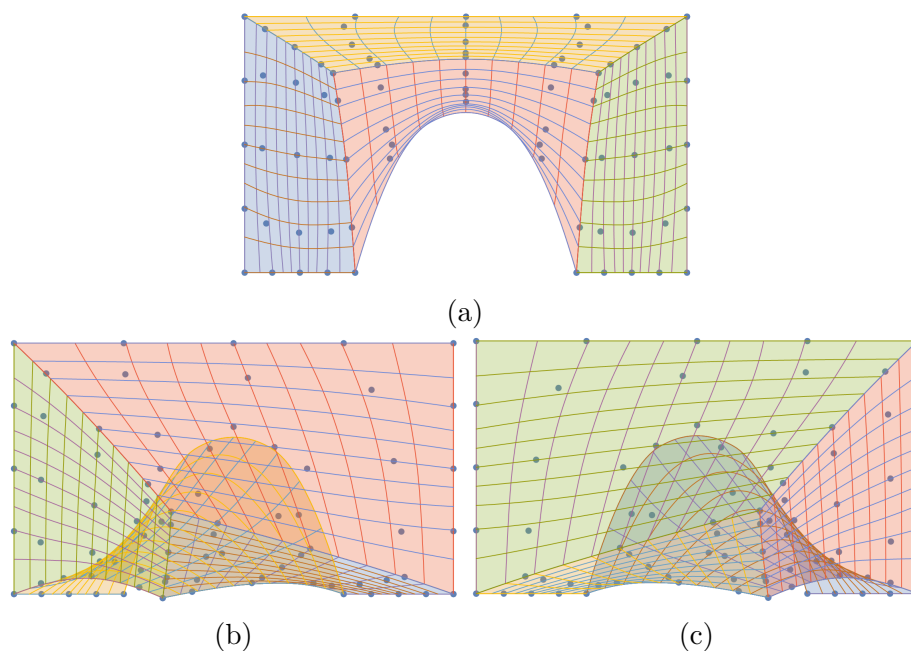
pro různou volbu grafů sousednosti po nelineární optimalizaci. V obrázku jsou vyznačeny různými barvami jednotlivé patche, řídicí body, a parametrické křivky (tj. křivky pro pevně zvolenou proměnnou u , resp. v).



Obrázek 5.2: Multi-patch parametrizace po nelineární optimalizaci

Tyto parametrizace byly spočteny pro volbu vah $\lambda_l = 1, \lambda_u = 1, \lambda_o = 3, \lambda_s = 100, \lambda_a = 100$. Na obrázku 5.3 je znázorněna multi-patch parametrizace pouze po lineární optimalizaci, je zde tedy ukázána důležitost funkcionalů použitých v nelineární optimalizaci, resp. důležitost vah λ_s, λ_a . Ačkoli tedy multi-patch parametrizace po lineární optimalizaci evidentně nebyla validní, je šance, že po nelineární optimalizaci validní bude.

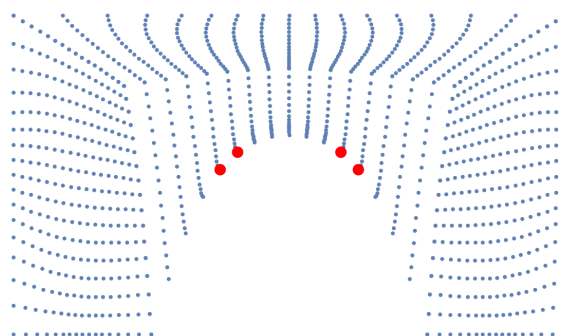
Na tomto jednoduchém příkladě je zřejmé, že problém v parametrizaci po lineární optimalizaci nastal v důsledku velkého zúžení oblasti ve svislém směru. Je-li tedy oblast dostatečně jednoduchá (konvexní), přijatelné výsledky dává už lineární optimalizace.



Obrázek 5.3: Po lineární optimalizaci

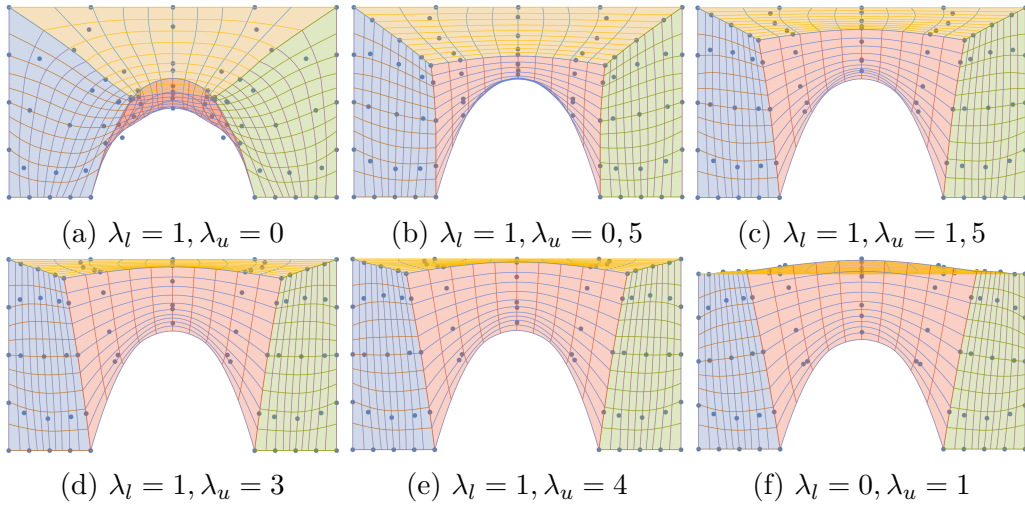
Hodnota cílové funkce po lineární optimalizaci pro multi-patch parametrizaci z obrázku 5.3a byla 104, pro 5.3b a 5.3c 115 (hodnoty jsou stejné, neboť na symetrickou oblast použijeme dva osově souměrné grafy sousednosti, výsledná MPP je tedy také osově souměrná). Ale ani jedna parametrizace není validní – pro obrázky 5.3b a 5.3c je to zřejmé.

Pro 5.3a problém nastává u oblouku, kde můžeme vidět, jak se parametrické křivky sbíhají k sobě a protínají se. Na obrázku 5.4 můžeme vidět body, ve kterých byl jakobián spočítán (pro případ na obrázku 5.3a). Červeně jsou zobrazeny body, kde je jakobián záporný nebo nulový, modrými body je zobrazen kladný jakobián.



Obrázek 5.4: Znázornění znaménka jakobiánu

Hodnota cílové funkce po nelineární optimalizaci pro MPP z obrázku



Obrázek 5.5: Porovnání vah λ_l, λ_u po lineární optimalizaci

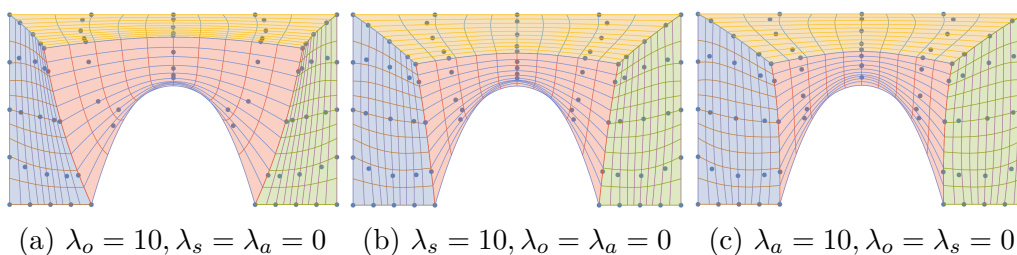
5.2a je rovna 1116, pro 5.2b a 5.2c je cílová funkce rovna 1668. Ale validní je jen pro první případ, tj. pro MPP z obrázku 5.2a (parametrické křivky se v tomto případě od sebe vzdálily, jsou rovnoměrněji rozmístěny než po lineární optimalizaci), tedy toto je hledaná multi-patch parametrizace.

Vliv vah λ_l, λ_u na kvalitu nalezené multi-patch parametrizace po lineární optimalizaci je zobrazen na obrázku 5.5. Z těchto šesti multi-patch parametrizací je validní pouze ta na obrázku 5.5c (pro $\lambda_l = 1, \lambda_u = 1,5$) a na obrázku 5.5d (pro $\lambda_l = 1, \lambda_u = 3$).

Ačkoli jsme získali validní multi-patch parametrizaci bez použití kroku s nelineární optimalizací, porovnáním těchto obrázků s multi-patch parametrizací získanou nelineární optimalizací (na obrázku 5.2a) můžeme vidět, že nelineární optimalizace zajistila to, že nejsou tak velké rozdíly ve vzdálenostech parametrických křivek v místě zúžení oblasti.

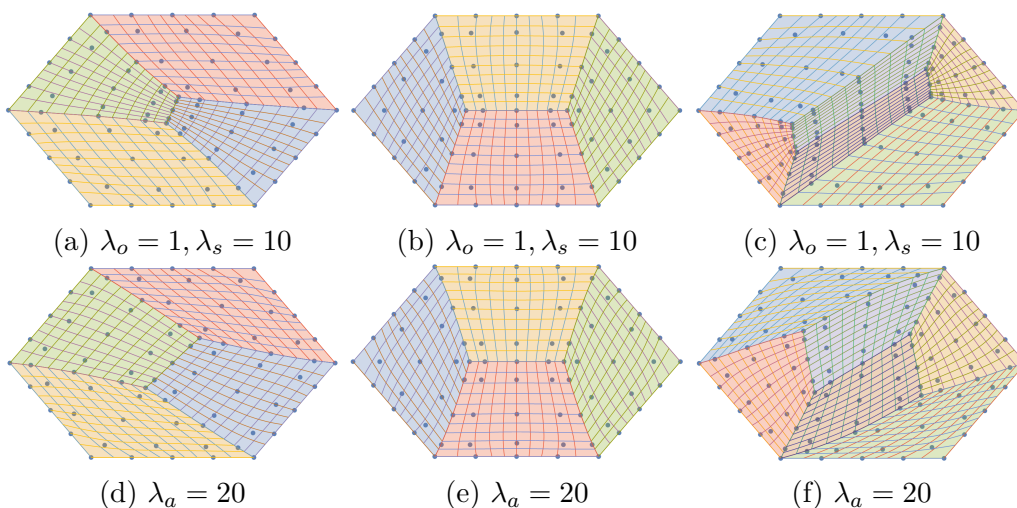
Na obrázku 5.6 je pak zobrazen vliv jednotlivých funkcionalů v nelineární optimalizaci. Ve všech případech jsou váhy $\lambda_l = \lambda_u = 1$ (viz obrázek 5.3a). Na obrázku 5.6a je zobrazena multi-patch parametrizace pro $\lambda_o = 10, \lambda_s = \lambda_a = 0$, na obrázku 5.6b pro $\lambda_s = 10, \lambda_o = \lambda_a = 0$ a na obrázku 5.6c pro $\lambda_a = 10, \lambda_o = \lambda_s = 0$. Validní MPP je ale pouze na posledním z těchto obrázků, u ostatních opět nastává problém u oblouku.

Obrázek 5.7 pak ukazuje různé validní multi-patch parametrizace pro šestiúhelníkovou oblast. První řádek zobrazuje multi-patch parametrizace, které vzniknou z různých grafů sousednosti s volbou vah $\lambda_l = \lambda_u = \lambda_o = 1, \lambda_s = 10, \lambda_a = 0$. Hodnoty cílových funkcí jsou: pro 5.7a 51, pro 5.7b 30, pro 5.7c 63. Druhý řádek zobrazuje multi-patch parametrizace, které vznikly ze stejných grafů sousednosti jako na prvním řádku. Hodnoty vah



Obrázek 5.6: Vliv dalších vah

jsou $\lambda_l = \lambda_u = 1, \lambda_o = \lambda_s = 0, \lambda_a = 20$. Hodnoty cílových funkcí jsou: pro 5.7d 219, pro 5.7e 215, pro 5.7f 168. Tedy nejmenší hodnota cílové funkce pro hodnoty vah z prvního řádku má druhá multi-patch parametrizace, pro hodnoty vah z druhého řádku poslední parametrizace.



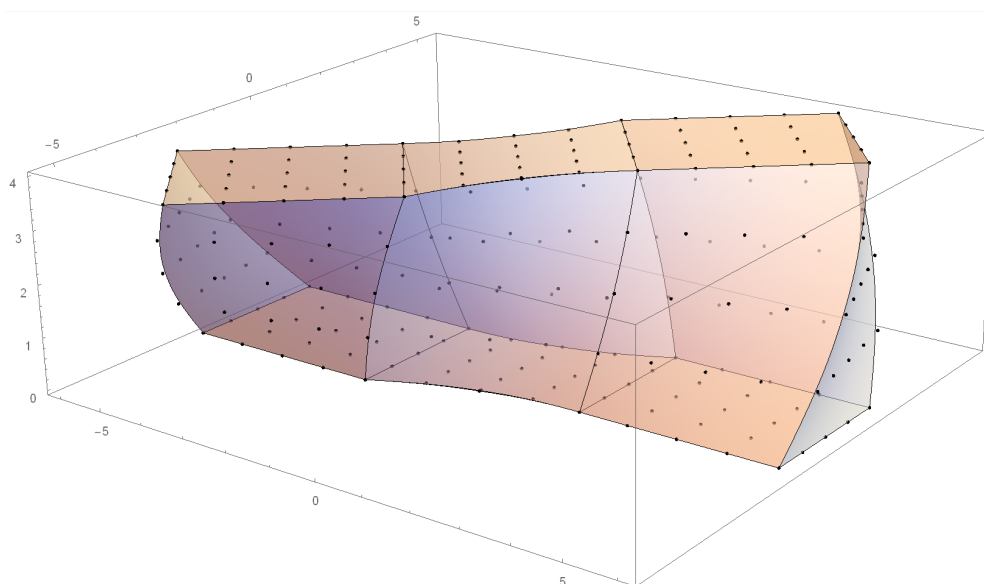
Obrázek 5.7: Multi-patch parametrizace šestiúhelníku

Z příkladů je zřejmé, že váhy mají vliv nejen na vzhled výsledné multi-patch parametrizace, ale i na hodnoty cílové funkce pro různé grafy sousednosti – pro jiné hodnoty vah může být optimální řešení reprezentováno jiným grafem sousednosti.

Z příkladů demonstrováných na první oblasti vidíme, že při relativně velké váze λ_a se parametrické křivky od sebe vzdalují tak, aby se neprotínaly. Pro složitější nekonvexní oblasti bude tedy vhodné používat váhu λ_a relativně velkou. Naopak u jednoduchých konvexních oblastí, kdy validní multi-patch parametrizace budou vznikat už po lineární optimalizaci, můžeme nechat tuto váhu nulovou. Další příklady a různé kombinace vah je možné dohledat v [2], [4], [7].

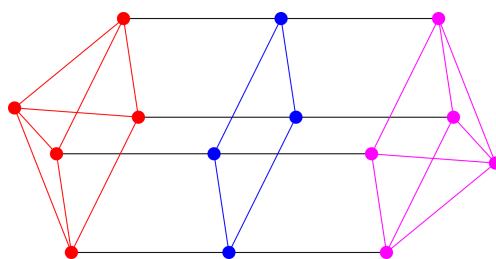
5.2 3D objem

Mějme objem na obrázku 5.8, který je určen 14 hraničními B-spline plochami. Každá B-spline plocha je učena sítí 5×5 řídicích bodů a neperiodickými uniformními vektory parametrizace $\mathbf{U} = \mathbf{V} = (0, 0, 0, 1/3, 2/3, 1, 1, 1)$ (tj. stupeň plochy je 2).



Obrázek 5.8: Objem zadaný 14 hraničními plochami

Zvolíme-li $p = 3$, tj. chceme objem parametrizovat pomocí 3 patchů. Obrázek 5.9 zobrazuje b-graf objemu s již nalezenými barevně odlišenými podgrafy (podle kapitoly 4.2.2).

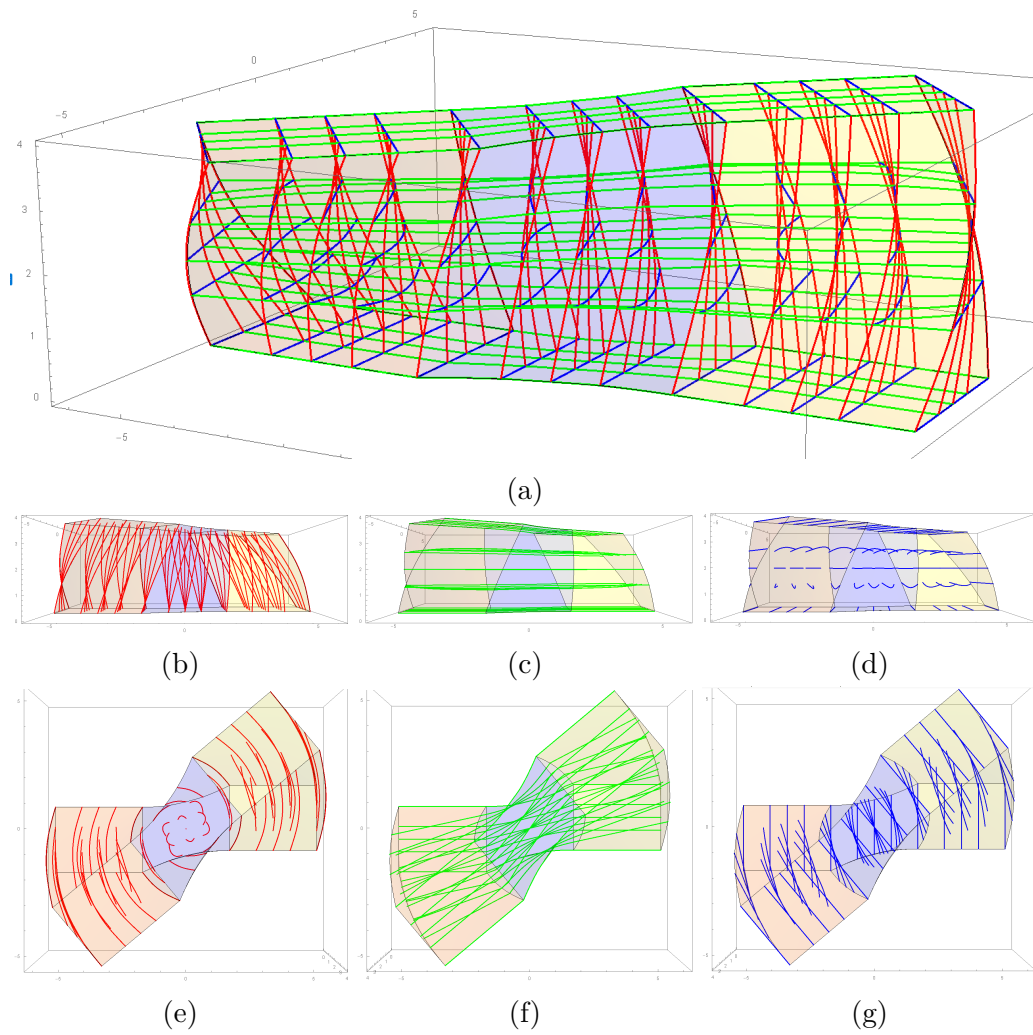


Obrázek 5.9: Graf susednosti segmentů (b-graf)

Protože každý segment je parametrizován pomocí 25 řídicích bodů, každý ze tří patchů bude určen 125 řídicími body.

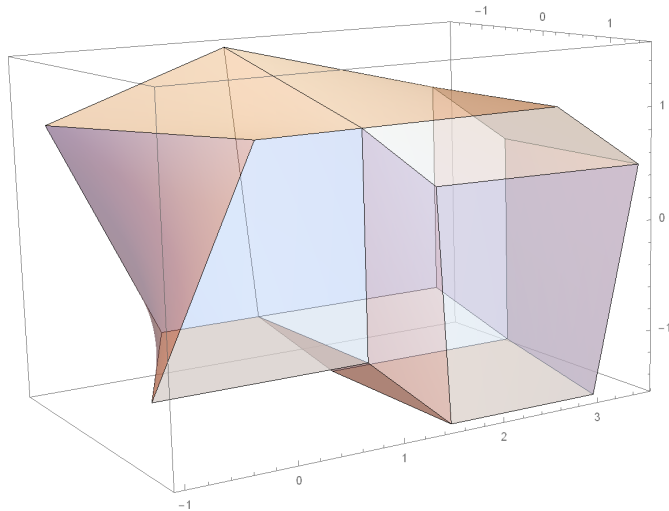
Na obrázku 5.10a je zobrazena validní multi-patch parametrizace po nelineární optimalizaci pro volbu vah $\lambda_l = \lambda_u = \lambda_o = 1$. Pro větší přehlednost na obrázku nejsou zobrazeny řídicí body. Jednotlivé patche jsou barevně od-

lišeny. Křivky zobrazené uvnitř objemu jsou parametrické křivky, pro lepší orientaci jsou zobrazeny z různých pohledů na obrázcích 5.10b až 5.10g.



Obrázek 5.10: Multi-patch parametrizace objemu

V kapitole 4.2.2 jsme uváděli b-graf (obrázek 4.5a) a na něm ilustrovali napojení na p-graf. Tento b-graf reprezentuje následující objem (obrázek 5.11).



Obrázek 5.11: Objem zadaný 16 hraničními plochami

Tento objem je reprezentovaný šestnácti hraničními B-spline plochami, každá plocha je určena vektory parametrizace $\mathbf{U} = \mathbf{V} = (0, 0, 0, 1/2, 1, 1, 1)$ a šestnácti řídicími body.

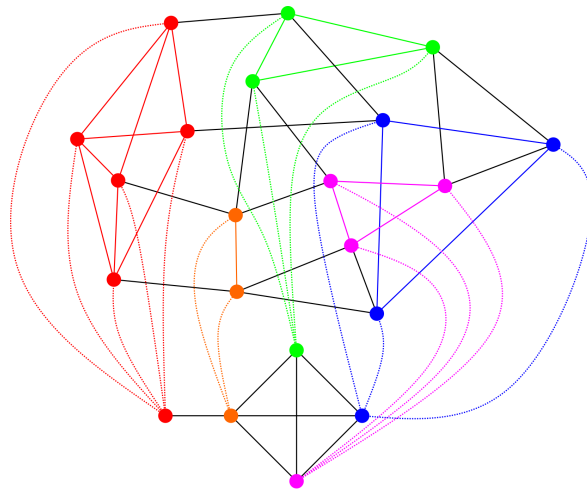
Uvažujme $p = 5$ a p-graf z obrázku 4.4, který jsme získali algoritmem pro generování p-grafů. Graf susednosti segmentů s již nalezenými podgrafy pro spojení s p-grafem je zobrazen na 4.5b. Postupem z kapitoly 4.2.2, tedy spojením příslušných uzlů obou grafů, získáme graf susednosti, který reprezentuje strukturu multi-patch parametrizace. Výsledný graf susednosti je zobrazen na obrázku 5.12.

Validní multi-patch parametrizace, jejíž strukturu zachycuje tento graf susednosti, je zobrazena na obrázku 5.13. Opět je získána nelineární optimalizací s volbou vah $\lambda_l = \lambda_u = \lambda_o = 1$, křivky uvnitř objemu jsou parametrické křivky multi-patch parametrizace.

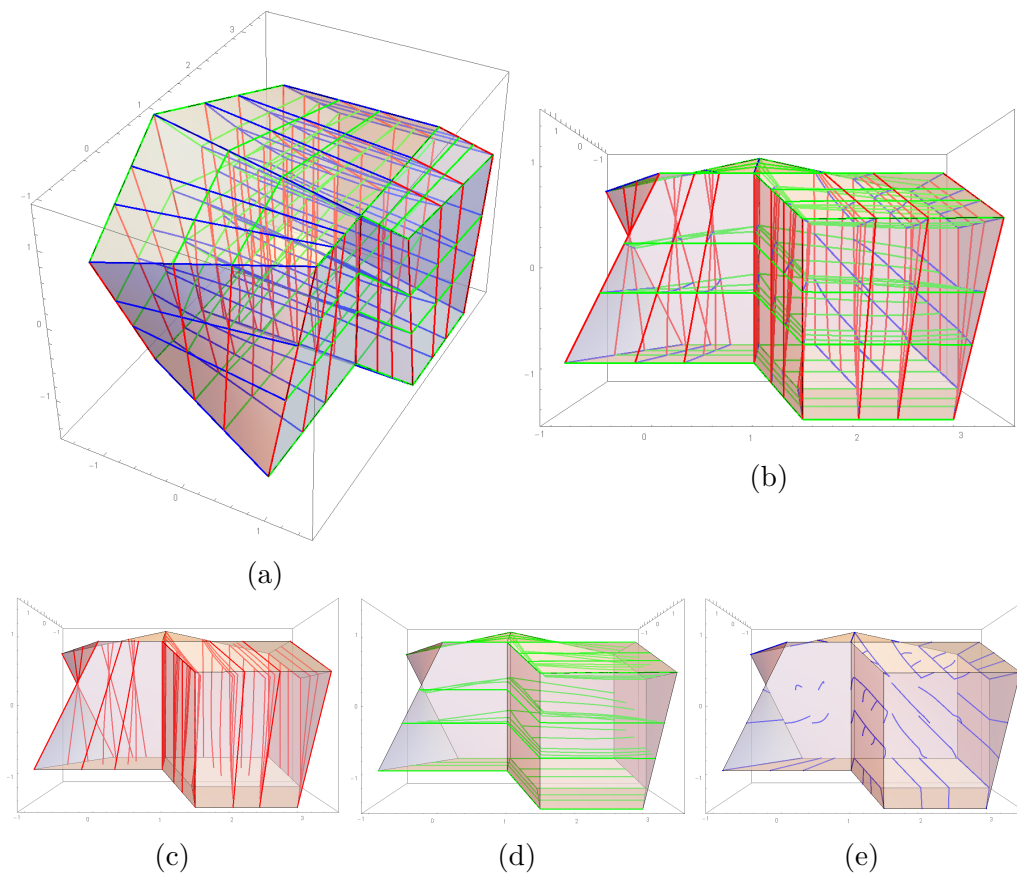
Normování cílové funkce

Kdybychom chtěli porovnávat kvalitu různých multi-patch parametrizací, které jsou určeny jedním grafem susednosti, ale rozdílnými váhami, musíme porovnávat cílové funkce, které jsou normované. Možným postupem normování je následující algoritmus.

1. Zvolíme pevně váhy a provedeme optimalizaci vzhledem k cílové funkci,



Obrázek 5.12: Spojení b-grafu s p-grafem



Obrázek 5.13: Multi-patch parametrizace objemu

kteřá vznikne dosazením těchto vah. Dostaneme tak hodnotu cílové funkce a parametrizaci oblasti.

2. Pro takto nalezenou geometrii určíme hodnoty jednotlivých funkcionalů.
3. Kroky 1 a 2 opakujeme pro všechny volby vah, které chceme zkoumat.
4. Provedeme přeškálování:
 - (a) Najdeme minimum a maximum ze všech hodnot funkcionalů Q_l (resp. Q_u , Q_o atd.), označme je $Q_{l,max}$, $Q_{l,min}$ (resp. $Q_{u,max}$, $Q_{u,min}$, atd.).

- (b) Pro danou volbu vah přeškálujeme příslušné hodnoty funkcionalů podle vztahu

$$Q_{l,scale} = \frac{Q_l - Q_{l,min}}{Q_{l,max} - Q_{l,min}},$$

a analogicky pro ostatní funkcionaly.

- (c) Pro každou konkrétní volbu vah tak dostaneme přeškálované hodnoty funkcionalů, které jsou v intervalu $[0, 1]$.
- (d) Pro tuto konkrétní volbu vah pak dostaneme přeškálovanou celkovou cílovou funkci jako součet $Q_{l,scale} + Q_{u,scale} + Q_{o,scale} + Q_{s,scale} + Q_{a,scale}$.

V tabulce 5.1 jsou uvedeny různé hodnoty vah $\lambda_l, \lambda_u, \lambda_o$ pro hledání multi-patch parametrizace šestnáctistěny z obrázku 5.11. Dále jsou v tabulce uvedeny hodnoty cílové funkce pro jednotlivé parametrizace a normované cílové funkce. Optimální multi-patch parametrizace je taková regulární parametrizace, která má nejnižší hodnotu normované cílové funkce. V tomto případě je to parametrizace, která je dána váhami $\lambda_l = 0, \lambda_u = 1, \lambda_o = 10$, jejíž cílová funkce po nelineární optimalizaci je rovna 158 a normovaná cílová funkce je rovna 0,908 (ostatní parametrizace s nižší hodnotou normované cílové funkce nebyly regulární).

Čas potřebný k výpočtu parametrizace

Uvedme ještě čas, který byl potřebný k výpočtu jedné multi-patch parametrizace. Veškeré výpočty byly prováděny na notebooku s dvoujádrovým procesorem Intel Core i3-7100U 2.4GHz, s 4GB operační pamětí v softwaru Wolfram Mathematica. Tabulka 5.2 ukazuje orientační časy potřebné k provedení jednotlivých operací a celkový čas. Sloupec „cíl. fce“ vyjadřuje čas

| λ_l | λ_u | λ_o | cílová funkce | normovaná cílová funkce |
|-------------|-------------|-------------|---------------|-------------------------|
| 1 | 1 | 1 | 108 | 1,130 |
| 0 | 1 | 1 | 55 | 1,199 |
| 0 | 1 | 10 | 158 | 0,908 |
| 0 | 1 | 100 | 918 | 0,494 |
| 0 | 10 | 1 | 434 | 1,307 |
| 0 | 100 | 1 | 4216 | 1,323 |
| 1 | 0 | 1 | 56 | 0,492 |
| 1 | 0 | 10 | 116 | 0,932 |
| 1 | 0 | 100 | 581 | 2,000 |
| 10 | 0 | 1 | 474 | 0,830 |
| 100 | 0 | 1 | 4630 | 1,045 |
| 1 | 1 | 0 | 95 | 1,220 |
| 1 | 10 | 0 | 474 | 1,310 |
| 1 | 100 | 0 | 4257 | 1,324 |
| 10 | 1 | 0 | 544 | 1,082 |
| 100 | 1 | 0 | 4764 | 1,011 |

Tabulka 5.1: Normovaná cílová funkce pro šestnáctistěn

potřebný k vyjádření cílové funkce, sloupec „zjednoduš.“ uvádí čas potřebný ke zjednodušení cílové funkce příkazem `Simplify`. Zjednodušením se sníží čas potřebný k minimalizaci (sloupec „minim.“).

Zatímco multi-patch parametrizace 2D oblasti byla spočtena v řádu desítek sekund, 3D oblast až v řádu několika minut (v případě šestistěnu na obrázku 5.13 cca 6 minut a 40 sekund).

| Oblast | Lineární optimalizace | | | Nelin. optimalizace | | celkem |
|---------------------------|-----------------------|------------|--------|---------------------|--------|--------|
| | cíl. fce | zjednoduš. | minim. | cíl. fce | minim. | |
| Tunel (obr. 5.2a) | 3 | 2 | 2 | 10 | 3 | 20 |
| Šestiúhelník (obr. 5.7f) | 4,5 | 6 | 5 | 17 | 1,5 | 34 |
| Šestnáctistěn (obr. 5.13) | 95 | 25 | 10 | 180 | 90 | 400 |

Tabulka 5.2: Doba (v sekundách) potřebná k výpočtu MPP

6 Závěr

Cílem práce byl teoretický popis hledání B-spline a NURBS parametrizace 2D oblasti i 3D objemu (které jsou zadané pomocí B-spline nebo NURBS hranice) pro další využití v isogeometrické analýze a následné předvedení uvedených postupů na příkladech. Práce obsahuje podrobný popis vzniku multi-patch parametrizace, tedy takové parametrizace, kdy oblast rozdělíme na menší disjunktní podoblasti. Pro tento přístup bylo nutné zavést pojem graf sousednosti a zkoumat vlastnosti těchto struktur. Mezi hlavní přínosy práce tedy patří podrobný popis generování grafů sousednosti pomocí ternárních stromů ve 2D případě a ilustrování algoritmu na konkrétním příkladu.

Dalším přínosem je popsání problémů, které nastávají při zobecňování do 3D a následný návrh řešení těchto problémů. Využitím poznatků z teorie grafů jsme zjistili, že vlastnosti grafu sousednosti ve 3D případě jsou značně odlišné od 2D případu (díky větě 2), byl tedy popsán odlišný algoritmus pro hledání těchto grafů. Z důvodu složitější struktury ale není detailně algoritmicky řešeno přiřazování jednotlivých patchů v závislosti na grafu sousednosti, proto by případné rozšíření práce mohlo řešit tento otevřený problém – tedy svázání grafové struktury s multi-patch parametrizací. I přes problémy spojené se složitostí struktury se podařilo navrhnout částečné řešení.

Praktická část práce se věnuje experimentům a ilustruje poznatky z teoretické části na konkrétních příkladech pro plošnou i objemovou parametrizaci, v příkladech je diskutován vliv funkcionalů zavedených v teoretické části na kvalitu výsledné multi-patch parametrizace.

Literatura

- [1] AIGNER, M. et al. Swept Volume Parameterization for Isogeometric Analysis. In HANCOCK, E. R. – MARTIN, R. R. – SABIN, M. A. (Ed.) *Mathematics of Surfaces XIII*, s. 19–44, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-03596-8.
- [2] BUCHEGGER, F. – JÜTTLER, B. Planar multi-patch domain parameterization via patch adjacency graphs. *Computer-Aided Design*. 2017, 82, s. 2 – 12. ISSN 0010-4485. doi: <https://doi.org/10.1016/j.cad.2016.05.019>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0010448516300483>. Isogeometric Design and Analysis.
- [3] COTTRELL, J. A. – HUGHES, T. J. R. – BAZILEVS, Y. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley Publishing, 1st edition, 2009. ISBN 0470748737, 9780470748732.
- [4] FALINI, A. – ŠPEH, J. – JÜTTLER, B. Planar Domain Parameterization with THB-splines. *Comput. Aided Geom. Des.* May 2015, 35, C, s. 95–108. ISSN 0167-8396. doi: 10.1016/j.cagd.2015.03.014. Dostupné z: <http://dx.doi.org/10.1016/j.cagd.2015.03.014>.
- [5] GRAVESEN, J. et al. *Planar Parametrization in Isogeometric Analysis*. Springer, 2014. doi: 10.1007/978-3-642-54382-1_11. ISBN 978-3-642-54381-4.
- [6] JEŽEK, F. Geometrické a počítačové modelování, 2010.
- [7] JÜTTLER, B. et al. Report and prototype software for geometry-aware block structuring and volume parameterization. *Multi-Objective design Optimization of fluid eneRgy machines*. 9. 1. 2017.
- [8] KUBA, M. A note on naturally embedded ternary trees. 03 2009, 18. Dostupné z: <https://dmg.tuwien.ac.at/kuba/preprintdeeb2.pdf>.
- [9] NGUYEN, T. – JÜTTLER, B. Parameterization of Contractible Domains Using Sequences of Harmonic Maps. In BOISSONNAT, J.-D. et al. (Ed.) *Curves and Surfaces*, s. 501–514, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-27413-8.
- [10] PIEGL, L. – TILLER, W. *The NURBS Book (2Nd Ed.)*. Springer-Verlag New York, Inc., 1997. ISBN 3-540-61545-8.

- [11] TAMAR-MATTIS, S. Planar Graphs and Wagner's and Kuratowski's Theorems. 2016. Dostupné z: <http://math.uchicago.edu/~may/REU2016/REUPapers/Tamar-Mattis.pdf>.
- [12] TRIPATHI, A. – VENUGOPALAN, S. – WEST, D. B. A short constructive proof of the Erdős–Gallai characterization of graphic lists. *Discrete Mathematics*. 2010, 310, 4, s. 843 – 844. ISSN 0012-365X. doi: <https://doi.org/10.1016/j.disc.2009.09.023>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0012365X09004683>.
- [13] WIKIPEDIA CONTRIBUTORS. Gaussian quadrature — Wikipedia, The Free Encyclopedia, 2018. Dostupné z: https://en.wikipedia.org/w/index.php?title=Gaussian_quadrature&oldid=837460315. [Online; accessed 14-May-2018].