VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Převod trojúhelníkových polygonálních 3D sítí na 3D spline plochy

Ročníkový projekt

2006 **Miroslav Šenk**

Převod trojúhelníkových polygonálních 3D sítí na 3D spline plochy

Odevzdáno na Fakultě informačních technologií Vysokého učení technického v Brně dne 2. května 2006

Prohlášení

Prohlašuji že jsem tuto práci vypracoval samostatně pod vedením Ing. Přemysla Krška, Ph.D. uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……………………
Miroslav Šenk
2. května 2006

Abstrakt
Tato práce se zabývá převodem trojúhelníkových polygonálních 3D sítí na 3D spline plochy. Zpracovávanou síť rozdělíme pomocí vlastních vektorů Laplaceova operátoru na čtyřúhelníkové oblasti. Ty budou tvořit jednotlivé spline plochy. Předvedeme některé zajímavé výsledky získané pomocí této metody a provedeme zhodnocení jejích kladů a záporů.

Klíčová slova
spline, vlastní vektory Laplaceova operátoru, Morsův-Smalův komplex

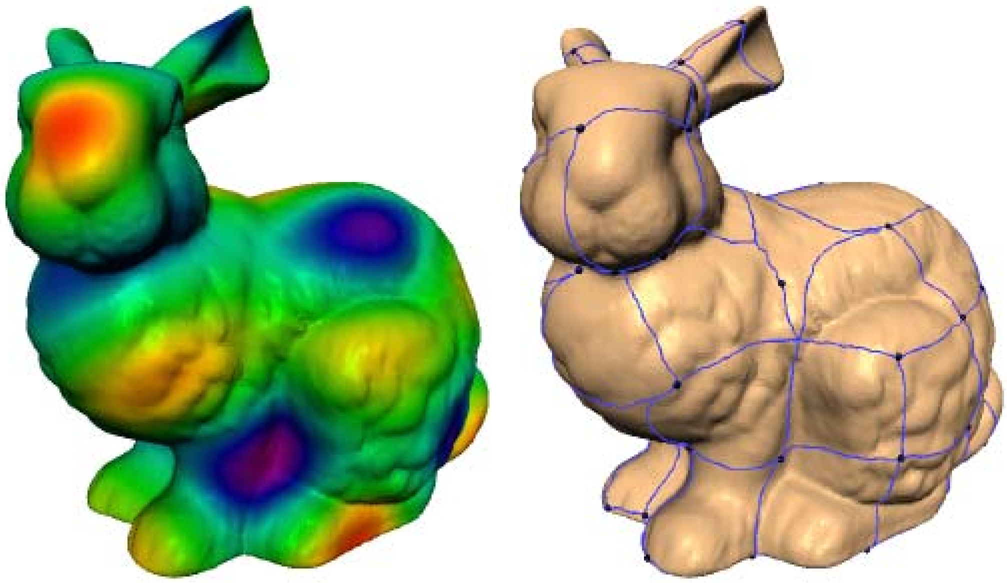
Abstract
In this work we deal with conversion of 3D triagonal polygonal meshes to the 3D spline patches. The converted mesh is divided into quadrilaterals using eigenvectors of Laplacian operator. These quadrilaterals will be converted into spline patches. We will present some interesting results of this method. The assets and imperfections of this method will be briefly discussed.

Keywords
spline, eigenvectors Laplacian, Morse-Smale complex

Obsah

1 Úvod

V současné době, jsou v mnoha oborech získávána trojrozměrná data, které je třeba vizualizovat. Z prvotních získaných dat, lze například metodou marching cubes, získat trojúhelníkovou polygonální 3D sít. Polygonální sítě, zvláště ty, které jsou získány skenováním, často vykazují nedostatky. Mají nedostatečné rozlišení nebo obsahují nevhodně tvarované elementy. Proto tyto sítě je třeba dále zpracovat, především model vyhladit a přitom zachovat jeho podstatné vlastnosti, jako zachování tvarových detailů a objemu. Cílem tohoto ročníkového projektu bylo vyzkoušet další zpracování takového modelu převodem na model tvořený spline plochami. K tomuto účelu je vhodné nejprve převést původní trojúhelníkovou síť na síť tvořenou čtyřúhelníkovými oblastmi, které by se měly blížit čtvercům (tj. přibližně stejná délka stran a vnitřní úhly přibližně 90°). K tomuto účelu je vhodné použít novou metodu [2] využívající vlastností vlastních vektorů Laplaceova operátoru k definování skalárního pole na povrchu sítě, viz obr. 1. Toto pole je analyzováno pomocí Morsovy teorie. Pro výslednou kvalitu spline ploch je obzvláště důležité, aby rozdělení čtyřúhelníků bylo velice kvalitní. Z těchto čtyřúhelníkových oblastí získáme dalšími metodami výsledné spline plochy.



Obrázek 1: Vlastní pole Laplaceova operátoru a z něho získaný Morsův-Smalův komplex

2 Popis metody

2.1 Spektrální analýza sítě

Je dobře známo, že diskrétní Laplaceův operátor po částech lineární funkce přes trojúhelníkovou síť je dán vztahem:

$$\Delta f_i = \sum_{j \in N_i} w_{ij} \left(f_j - f_i \right)$$

kde N_i je množina vrcholů přiléhajících k vrcholu i a w_{ij} je skalární hodnota přidělená hraně (i, j). Hodnota w_{ij} je dána vztahem:

$$w_{ij} = \frac{1}{2} \left(\cot \alpha_{ij} + \cot \beta_{ij} \right)$$

Hodnoty α_{ij} a β_{ij} jsou úhly naproti hraně (i, j). Jestliže reprezentujeme funkci *f* sloupcem vektorů jeho hodnot na všech jeho vrcholech $f = \left[f_1 f_2 ... f_n \right]^T$, můžeme přeformulovat Laplaceův operátor jako matici

$$\Delta f = -Lf$$

kde je tato matice L je definována jako

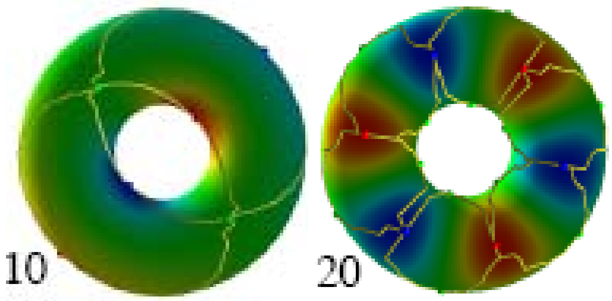
$$L_{ij} = \begin{cases} \sum_k w_{ik} & \text{jestliže i = j,} \\ -w_{ik} & \text{jestliže (i, j) je hrana sítě M,} \\ 0 & \text{jinak.} \end{cases}$$

Vlastní hodnoty $\lambda_1 = 0 \leq \lambda_2 \leq \lambda_3 \leq ... \leq \lambda_n$ L vytváří spektrum sítě a odpovídající vlastní vektory e_1, e_2, e_3 , …, e_n L definují po částech lineární funkci přes síť M.

2.2 Vytvoření čtyřúhelníkového základu

Při rozdělování sítě na čtyřúhelníkové oblasti požadujeme, aby tato oblasti byly opravdu čtyřúhelníkové a také chceme aby byly co nejlépe tvarovány. To samozřejmě závisí na volbě funkce $f: V \rightarrow R$, jenž má přiřadit každému vrcholu sítě skalární hodnotu. Tuto funkci získáme z vlastních vektorů Laplaceovy matice *L*.

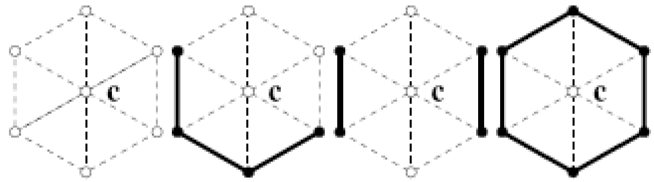
Každý vlastní vektor e matice *L* implicitně určuje funkci $f: V \rightarrow R$ přes síť M. Hodnota této funkce f na vrcholu i jednoduše odpovídá hodnotě e_i na řádku i vlastního vektoru. Tuto funkci f, definovanou vlastním vektorem matice *L*, nazýváme vlastní pole. Pro náš účel mají vlastní funkce několik velice důležitých vlastností. Tou nejdůležitější je z našeho hlediska fakt, že extrémy funkce jsou rovnoměrně rozmístěny, bez této vlastnosti by vzniklé čtyřúhelníkové oblasti byly velice nekvalitní. Další důležitou vlastností vlastních funkcí je také to, že se vyskytují v řádu s rostoucím kmitočtem a tudíž s rostoucím počtem kritických bodů. Z toho plyne, že počet uzlových domén vlastní funkce s vlastní hodnotou λ_k je nejvíce *k*. Díky této vlastnosti je velice jednoduché určit, který vlastní vektor zvolit pro určení funkce *f*. Proto nám také stačí spočítat prvních 40 – 80 vlastních vektorů (obr. 2).



Obrázek 2: Ukazuje prstenec a jeho vlastní pole,
pro jeho 10. a 20. vlastní vektor.

2.3 Morsova teorie

Morsova teorie se zabývá analýzou sítě a pomůže nám klasifikovat vrcholy sítě na minima, maxima a sedla, pomocí předem přidělených jedinečných skalárních hodnot vrcholům, které jsou dány po částech lineární funkcí. Klasifikace vrcholu probíhá na základě jeho hodnoty a hodnoty jeho okolí. Okolí tvoří vrcholy, které jsou s daným vrcholem spojeny hranou. Vrchol je klasifikován jako minimum, je-li jeho hodnota menší než hodnota okolních vrcholů. Má-li vrchol hodnotu větší než jeho okolí, jedná se o maximum. Dojde-li ke čtyřem změnám mezi větší a menší hodnotou vrcholu a jeho okolí, jde o sedlo. V ostatních případech je vrchol klasifikován jako regulární (obr. 3).



Obrázek 3: Klasifikace vrcholu *c*. zleva: minimum, regulární bod, sedlo a maximum. Plný bod značí vyšší hodnotu než *c*, prázdný bod hodnotu nižší.

2.4 Získání Morsova-Smalova komplexu

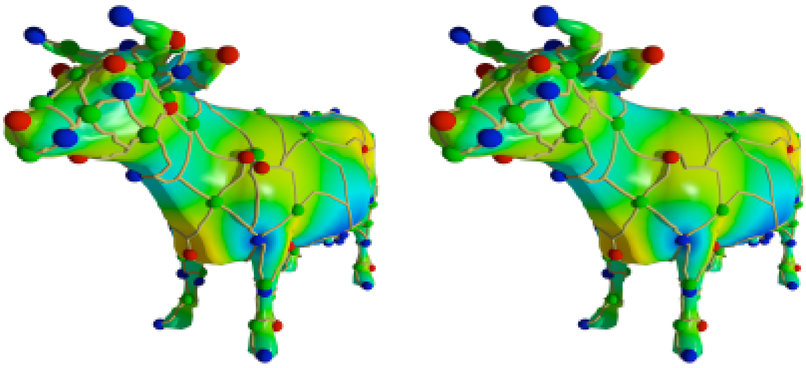
Po dopočítání hodnot vrcholů a jejich klasifikaci, můžeme vytvářet tzv. Morsův-Smalův komplex. Ten nám model rozdělí do oblastí. Morsův-Smalův komplex vytváříme tak, že pro každé sedlo dopočítáme čtyři cesty, které dané sedlo spojují s okolními extrémy. Dvě cesty jsou stoupající a spojují sedlo s maximy a dvě jsou klesající a spojují sedlo s minimy. Po dopočítání všech cest by měla být síť rozdělena na buněk. Každá buňka by měla obsahovat dvě sedla.

2.5 Optimalizování komplexu

Po získání Morsova-Smalova komplexu výše popsaným postupem zjistíme, že komplex není zcela vhodný a že ho je třeba optimalizovat. Toto je způsobeno numerickým výpočtem vlastních vektorů. Ten zanese do vlastního pole nějaký šum, který způsobí, že vzniknou nežádoucí kritické body. Také cesty spojující extrémy mohou být neuspokojivé. Tyto chyby odstraňujeme použitím dvou optimalizačních procesů. Nejprve optimalizujeme topologii komplexu odstraněním nepatřičných kritických bodů, posléze optimalizujeme geometrii buněk.

2.5.1 Topologická optimalizace

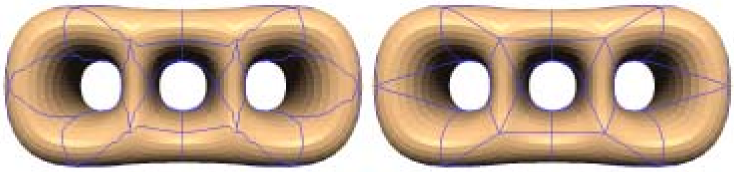
K odstranění nepatřičných kritických bodů používáme *rušení*. Při každém rušení je odstraněna propojená dvojce sedlo a minimum nebo maximum. Rušení provádíme na základě *trvalosti*. Trvalost je určena jako absolutní hodnota rozdílu hodnot dvojce sedlo a minimum nebo maximum. Dvojce s nejmenší trvalostí jsou zrušeny nejdříve (obr. 4).



Obrázek 4: Na modelu krávy (vlevo) lze vidět nežádoucí kritický bod na ramenu.
Model vpravo je po topologické optimalizaci.

2.5.2 Geometrická optimalizace

Tato optimalizace má za úkol odstranit nevhodné tvary oblastí. To se provádí ve dvou krocích. V prvním kroku odstraníme extrémy umístěné příliš blízko sebe. Vždy dva extrémy, dvě sedla, dvě minima nebo dvě maxima, zredukujeme na jeden odstraněním extrému, který má menší hodnotu. Dalším krokem geometrické optimalizace narovnávání. Narovnávání napřimuje hranice mezi buňkami, které tvoří cesty mezi jednotlivými extrémy po hranách sítě (obr. 5).



Obrázek 5: Optimalizace cest

3 Implementace

Danou metodu jsem implementoval strukturálně v C++. Zpracování dat jsem prováděl pomocí dvou aplikací. První aplikace, počítá vlastní čísla a vlastní vektory a druhá zobrazuje získaná data. Zpracovávaná data jsem získal z modelů, které mi poskytl dr. Kršek. Ty jsou uloženy ve formátu definovaném knihovnou VectEntity2, jejíž autorem je dr. Kršek.

3.1 Získání vlastních vektorů Laplaceova operátoru

Jelikož knihovna VectEntity2 uchovává pouze vrcholy a trojúhelníky, jež jsou uloženy v lineárních seznamech, uchovávat hrany není nutné, prvním mým krokem bylo vytvoření hran (funkce VytvoreniHran). Ty jsou vhodné pro dočasné přechování hodnoty w_{ij} (funkce OhodnoceniHran), která je použita pro následné vytvoření matice *L*. Tu je nejvhodnější vytvořit dvojitým průchodem sítě po vrcholech (funkce VytvorMatici).

Jedním z nejtěžších kroků je získání samotných vlastních čísel a vlastních vektorů této matice. Za tímto účelem jsem implementoval dvě metody. První metoda počítá vlastní čísla a vlastní metody pomocí Jacobiho rotací. Tato metoda je sice spolehlivá, ale je velice pomalá a pro rozsáhlé matice je takřka nepoužitelná. Druhá metoda, kterou jsem implementoval, je QL algoritmus s implicitními posuny [3], který počítá vlastní čísla a vlastní vektory tridiagonální matice (funkce tred2). Tu získám z matice *L* Householderovou redukcí [3] (funkce tqli). Všechny tyto výpočty jsou prováděny v aplikaci *vypocet*.

3.2 Vytvoření Morsova-Smalova komplexu a jeho optimalizace

Klasifikaci vrcholů provádím ve funkci KlasifikaceVrcholu. Při klasifikaci srovnávám hodnotu každého vrcholu s hodnotami vrcholů v jeho okolí. Tyto vrcholy mám seřazeny v protisměru hodinových ručiček, jak jsou okolo aktuálního vrcholu.

Morsův-Smalův komplex je dopočítáván tak, že se ke každému sedlu najdou dvě nejbližší maxima a dvě nejbližší minima. Vzdálenost určuji přímou vzdáleností sedla k danému extrému. Posléze hledám cestu po hranách modelu. Tuto cestu počítán iterativním způsobem, vždy použiji tu hranu, která má nejmenší odchylku od přímého směru k danému vrcholu.

Optimalizace jsem řešil podle dříve uvedeného postupu. Ve funkci TopOp provádím topologickou optimalizaci. Nejdříve každé dvojici sedlo-minimum nebo sedlo-maximum přiřadím absolutní hodnotu rozdílu jejich funkčních hodnot a posléze tyto hodnoty normalizuji na hodnotu 0 až 100. Nakonec odstraním ty dvojce, které mají menší hodnotu než je hodnota zadaná uživatelem. Další optimalizaci provádím průchodem kritických bodů a jsou-li dva kritické body stejného typu u sebe, je ten s menší hodnotou odstraněn. U minima je odstraněno to s větší hodnotou (funkce Redukce).

3.3 Ovládání programů

Program **vypocet**:
Parametry

- -v výpočet vlastních čísel a vlastních vektorů a jejich výpis do souboru, za tímto parametrem následuje jako další parametr název souboru s modelem ve formátu VectEntity2.
- -n načte vlastní čísla a vektory ze souboru, který následuje jako druhý parametr a vypíše *k*-tý vlastní vektor, kde *k* je třetí parametr.

Program **Prohlizec**:

Parametry

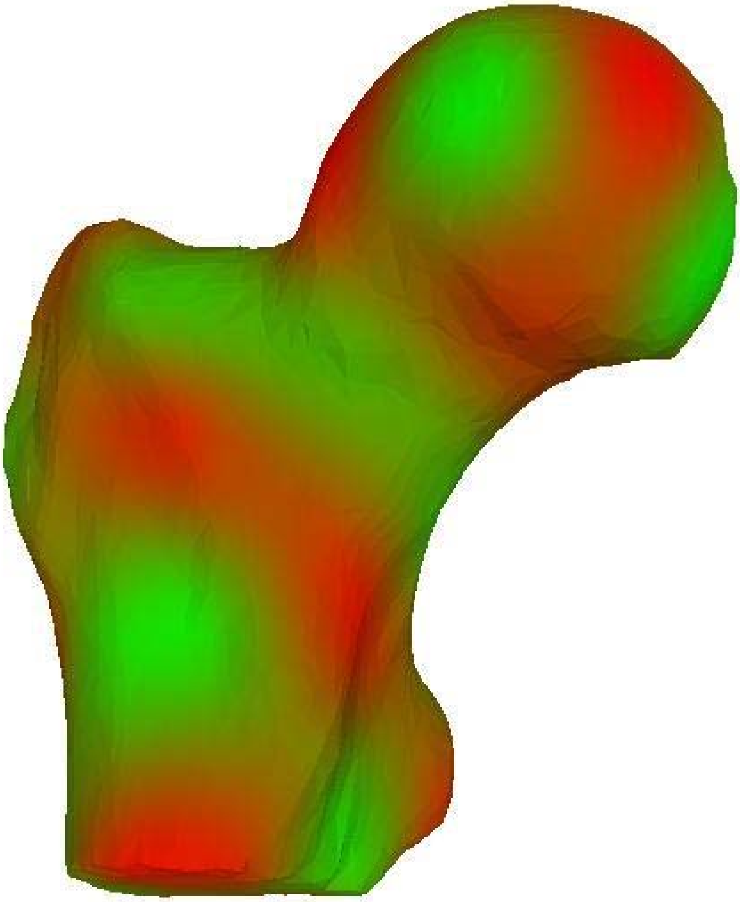
-n načte model ze souboru, který následuje jako druhý parametr a zpracuje ho podle vlastního vektoru, který je zadán jako třetí parametr, čtvrtý parametr udává trvalost pro topologickou optimalizaci.

Program se ovládá pomocí těchto kláves:

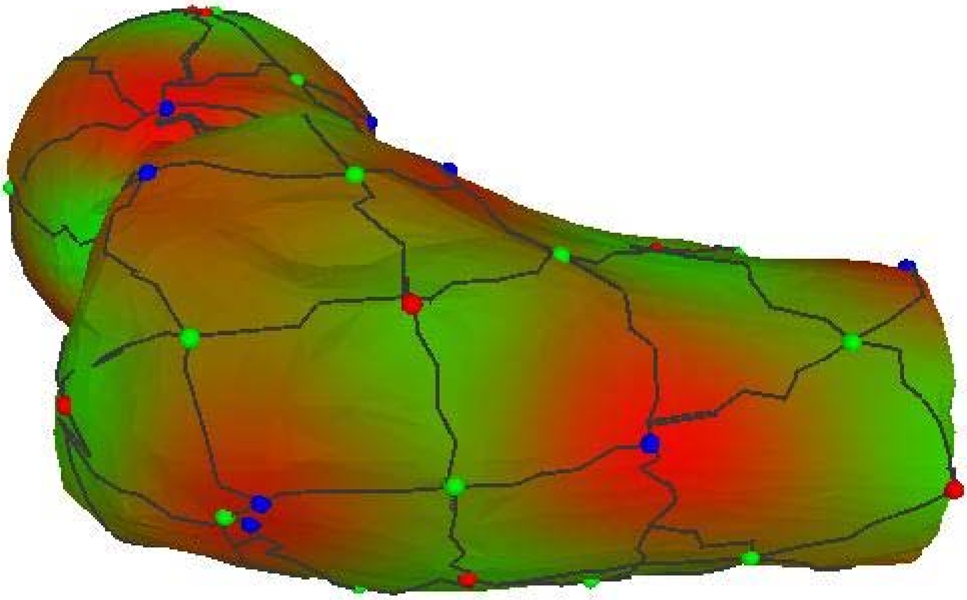
'q', 'x' a 'Esc'	- ukončí program
'1'	- vykreslí se model se spektrem
'2'	- vykreslí se model se spektrem a s kritickými body
'3'	- vykreslí se model se spektrem a Morsův-Smalův komplex
'4'	- vykreslí se Morsův-Smalův komplex

4 Výsledky

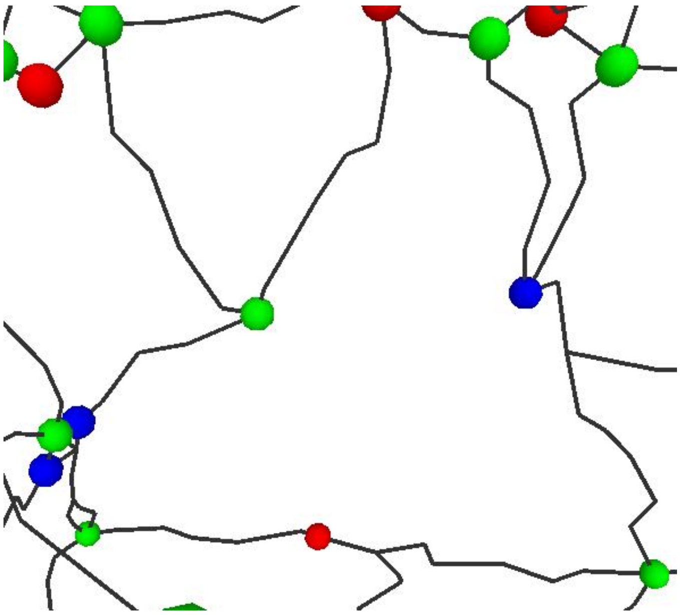
Implementovaný algoritmus jsem zkusil aplikovat na modely (obr. 6). Z obrázku 7 je patrné, že v některých oblastech je Morsův-Smalův komplex opravdu kvalitní. Ale již na obrázku 8, můžeme vidět že v jiných oblastech modelu je komplex značně nekvalitní. A při aplikaci topologické optimalizace došlo ještě k dalšímu zhoršení kvality (obr. 9). Tato chyba bude s největší pravděpodobností způsobena chybou při numerickém výpočtu vlastních čísel a vektorů. To dokazuje i obrázek 12. U výpočtu vlastních vektorů tohoto modelu byl použit datový typ float, zaokrouhlovací chybu ještě umocnil asi dvojnásobný počet vrcholů oproti předchozímu modelu.



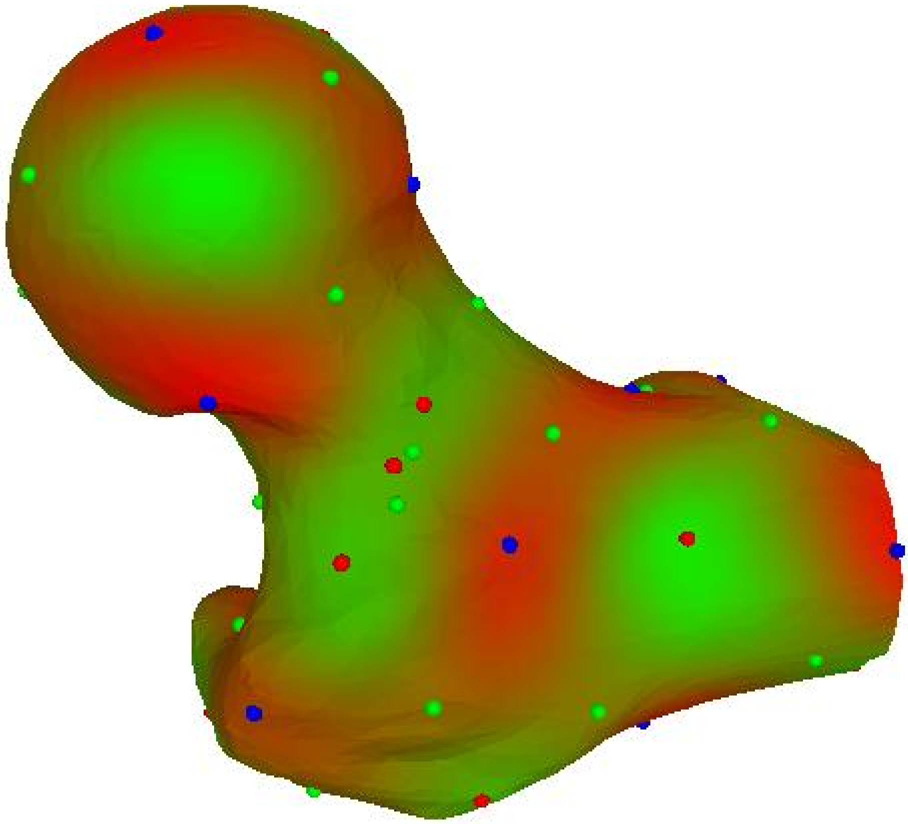
Obrázek 6: Vlastní pole Laplaceova operátoru



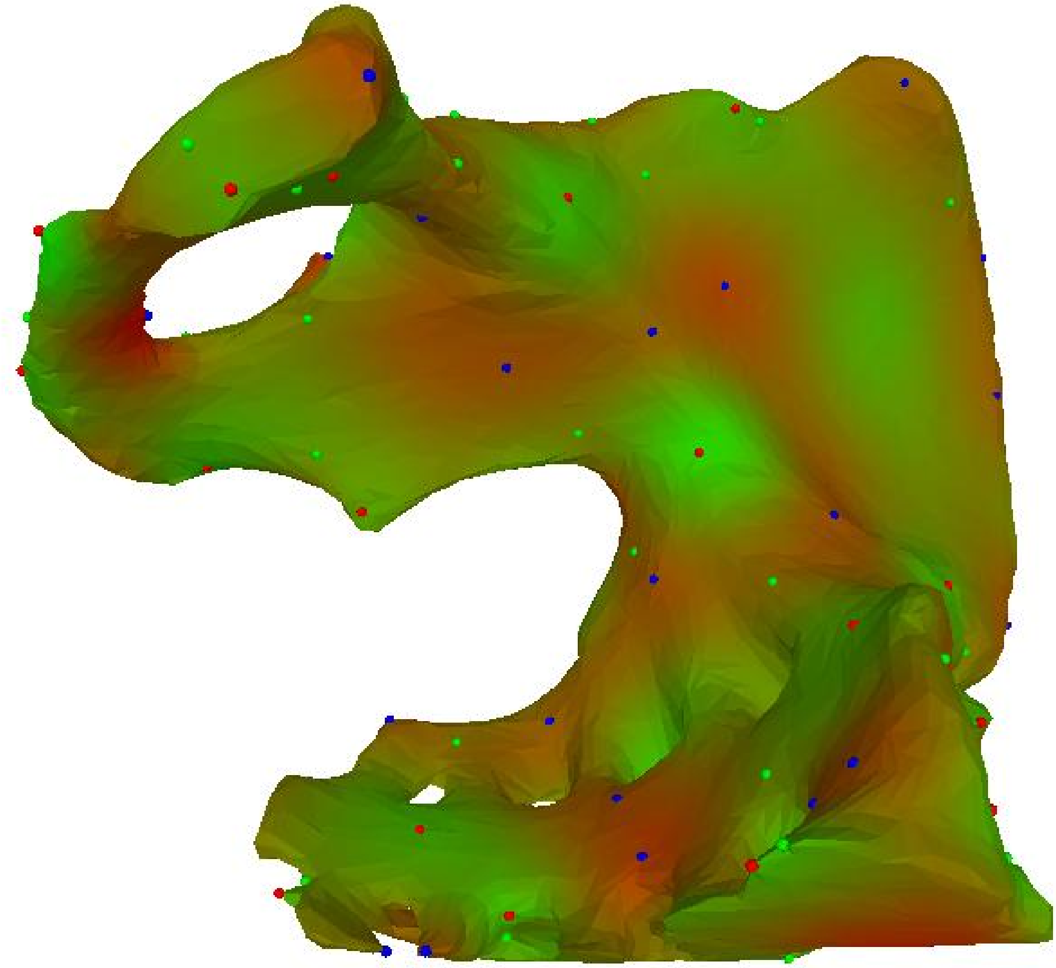
Obrázek 7: Morsův-Smalův komplex



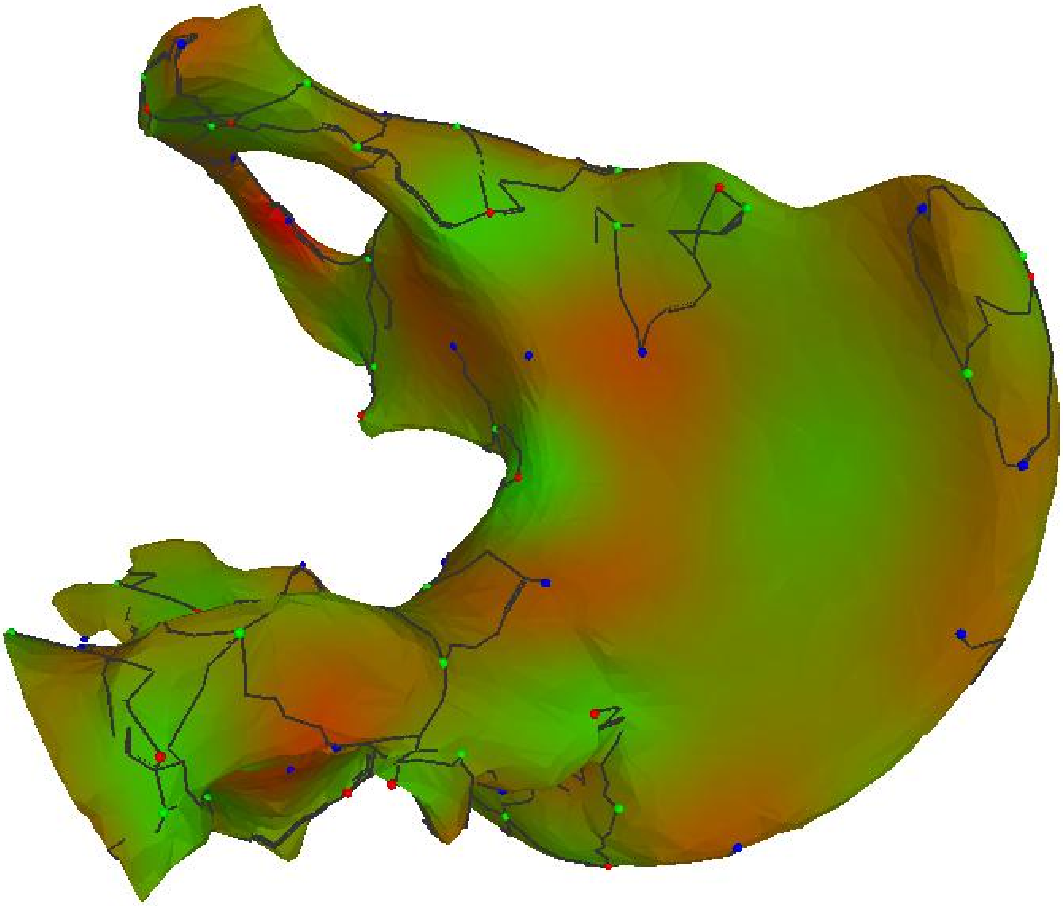
Obrázek 8: Chyba v komplexu



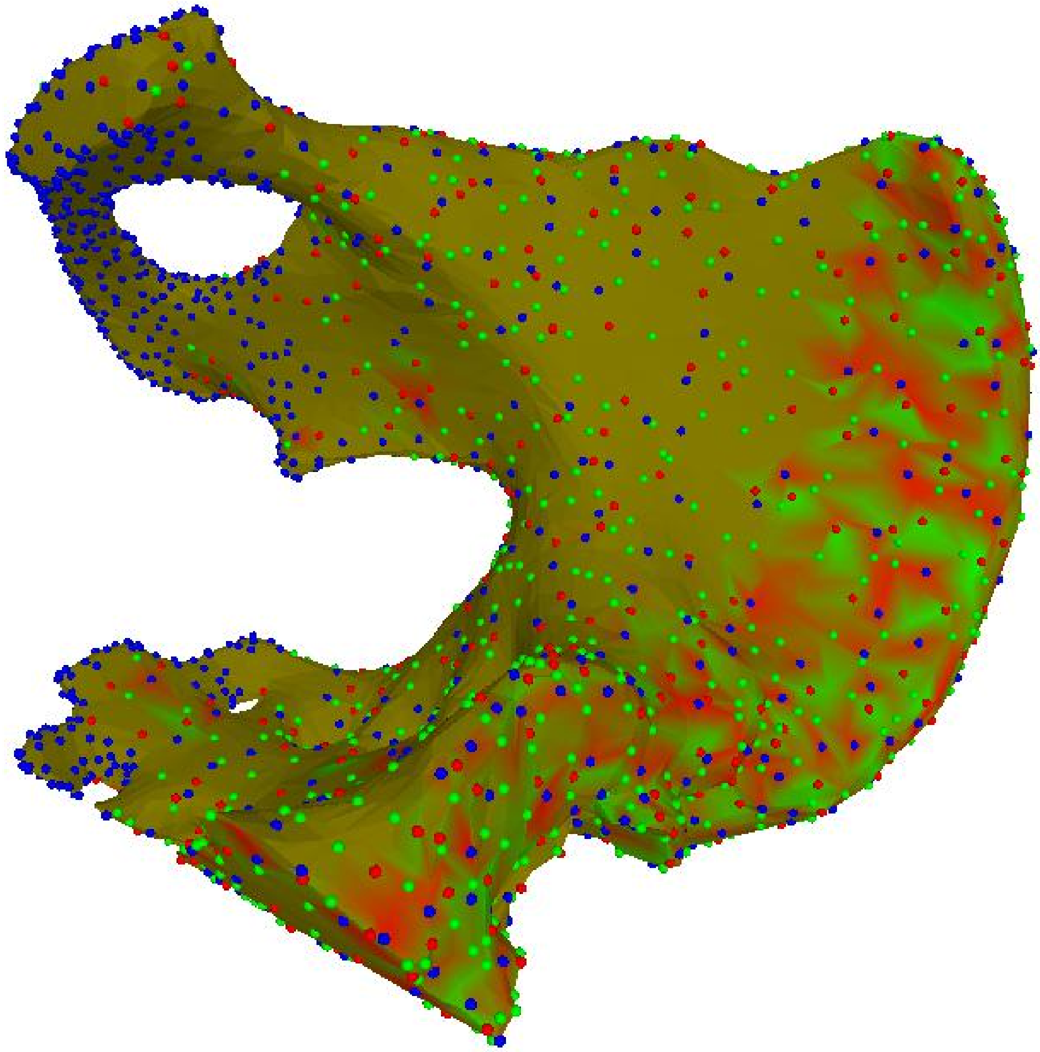
Obrázek 9: Zde byla použita topologická optimalizace s trvalostí 0,5, jak je na modelu vidět, byly zrušeny podstatné extremy v horní části modelu, a šum uprostřed modelu zrušen nebyl.



Obrázek 10: Spektrum modelu a jeho kritické body. Při výpočtu použit datový typ double



Obrázek 11: Očividné nedostatky Morsova-Smalova komlexu. Při výpočtu použit datový typ double



Obrázek 12: Zaokrouhlovací chyba je tak obrovská, že v určitých částech modelu se vyskytují jen maxima. Při výpočtu použit datový typ float

5 Závěr

Převod sítě na kvalitní čtyřúhelníkový základ je nejlepší cestou k převodu trojúhelníkových polygonálních 3D sítí na 3D spline plochy. Z tohoto čtyřúhelníkového základu lze již snadno dalšími metodami získat 3D spline plochy. V této práci jsem se zaměřil na co nejefektivnější získání semi-regulární čtyřúhelníkové sítě. Při implementaci metody využívající vlastních vektorů Laplaceova operátoru bylo dosaženo jistých úspěchů, ale Morsův-Smalův komplex stále obsahuje nadbytečné kritické body a v některých částech modelu vytváří velice nekvalitní oblasti. Toto lze přisoudit nedostatečně vhodné numerické metodě výpočtu vlastních čísel a vektorů. Pro zlepšení výsledků této metody by bylo vhodné použít sofistikovanější metody, např. Arnoldiho iterativní metody. A také užitím vhodnějších optimalizací, které byly nedávno zveřejněny.

Literatura

[1] Kršek P. Dokumentace knihovny VectEntity2
[2] Dong et al.: Quadrangulating a Mesh using Laplacian Eigenvectors, Technical Report No. UIUCDCS-R-2005-2583, 2005
[3] Numerical Recipes in C, kapitola 11, http://www.library.cornell.edu/nr/bookcpdf.html