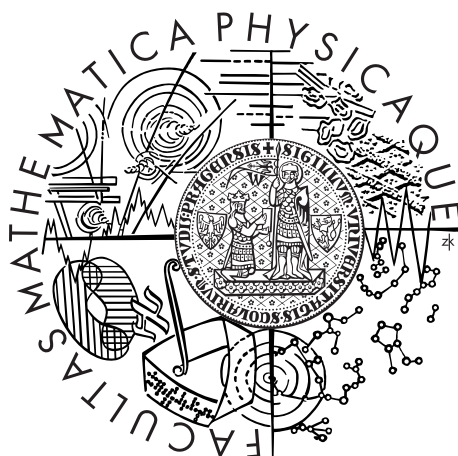


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Bc. Helena Švihlová

Aplikace metody konečných prvků na reálné problémy v hemodynamice

Matematický ústav UK

Vedoucí diplomové práce: RNDr. Ing. Jaroslav Hron, Ph.D.

Konzultant: RNDr. Martin Mádlík, Ph.D.

Studijní program: Matematika

Studijní obor: Matematické modelování ve fyzice a technice

Praha 2013

Na tomto místě bych ráda poděkovala všem, kteří mi se vznikem této práce jakkoliv pomohli. Největší dík patří RNDr. Ing. Jaroslavu Hronovi, Ph.D. za jeho trpělivost, odborné rady a čas, který mi věnoval. Děkuji RNDr. Martinu Mádlíkovi, Ph.D. za pomoc zvláště v počátcích práce a za poskytnutí programu Fstrin. V neposlední řadě děkuji své rodině za podporu v průběhu celého studia.

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Název práce: Aplikace metody konečných prvků na reálné problémy v hemodynamice

Autor: Bc. Helena Švihlová

Katedra: Matematický ústav UK

Vedoucí diplomové práce: RNDr. Ing. Jaroslav Hron, Ph.D., Matematický ústav UK

Abstrakt:

V této diplomové práci je studováno proudění nestlačitelných tekutin v geometriích mozkových cév postižených aneurysmatem. Aneurysma je lokální rozšíření tepny. Toto onemocnění je nebezpečné v případě, kdy dojde k prasknutí aneurysmatu a vylití krve do mozku. Potřeba výpočtu přesného rychlostního pole a zejména rozložení tlaku v geometriích cév postižených aneurysmatem je motivována právě otázkou, které aneurysma může být náchylné k prasknutí a je třeba je sledovat.

K výpočtu proudění je použita metoda konečných prvků. Jedním z důležitých kroků při jejím použití je dobrá diskretizace oblasti. Moderní počítačová tomografie (CT) umožňuje pořizovat celé série prostorově navazujících rovinných snímků a je třeba na základě těchto dat vytvářet odpovídající třírozměrné modely tkání. Součástí práce je popis získání výpočetních sítí ze segmentace pořízené CT skenem, možností jejich zhlazení a úprav.

V teoretické části jsou nejprve zformulovány použité rovnice včetně diskuze zadání vhodných okrajových podmínek až po nalezení slabé formulace úlohy a její diskretizace.

V části pro numerické výsledky jsou uvedena spočtená rychlostní a tlaková pole pomocí různých konečných prvků a dopočteno je i smykové napětí na stěnách, které hraje důležitou roli při vzniku a vývoji aneurysma. Uvedena jsou srovnání přístupů zhlazování sítí, použitých konečných prvků a použitých programů.

Klíčová slova:

hemodynamika, metoda konečných prvků, CT sken, aneurysma

Title: Application of finite element method to real problems in hemodynamics

Author: Bc. Helena Švihlová

Department: Mathematical Institute of Charles University

Supervisor: RNDr. Ing. Jaroslav Hron, Ph.D., Mathematical Institute of Charles University

Abstract:

The incompressible fluid flow around the geometries of cerebral artery aneurysms is studied in this thesis. The aneurysm is a local extension of a vessel. This disease is dangerous only in the case of rupture. Then the blood is released into the brain. The need of accurate computation of the velocity and pressure fields in this geometries is motivated exactly by the question which aneurysm has tendency to rupture.

The finite element method (FEM) is used for the computation of the flow. A good domain discretization is one of the main step in FEM. Modern computed tomography is able to produce series of the two-dimensional images and it is necessary to create an appropriate three-dimensional model of the tissue. This thesis includes the description of the mesh generation and the ways to smooth and improve the meshes.

In the theoretical part the equations of fluid flow are formulated. A suitability of a choice of boundary conditions is discussed. Weak formulation for the equations and its discretization are presented.

In the practical part velocity and pressure fields are computed by the various finite elements. Wall shear stress which plays an important role in the evolution of an aneurysm is also computed on the introduced meshes. Comparison of mesh smoothing filters, used finite elements and used programs is presented.

Keywords:

hemodynamics, finite element method, CT scan, aneurysm

Obsah

Úvod	1
1 Matematický popis proudění v cévách	2
1.1 Popis problému	2
1.2 Zadání rovnic a okrajových podmínek	2
1.3 Odvození variační formulace	6
1.3.1 Variační formulace pro stacionární rovnice	6
1.3.2 Diskretizace časové derivace	7
1.4 Diskretizace rovnic metodou konečných prvků a použitý software .	8
1.5 Použité konečné prvky a stupně volnosti	10
2 Příprava sítě	13
2.1 Získání voxelové sítě	13
2.2 Získání sítě	15
2.3 Zhlazení dat	17
2.3.1 Zhlazení diskrétním konvolučním jádrem	17
2.3.2 Zhlazení sítě pomocí zhlazovacího filtru	19
2.4 Úprava sítě	22
3 Numerické výsledky	30
3.1 Srovnání vytvořených sítí	30
3.2 Srovnání konečných prvků	33
3.3 Spočtená rychlostní a tlaková pole	35
3.4 Spočtená normálová a smyková napětí	39
Závěr	44
Seznam použité literatury	45
Seznam obrázků	48
Seznam tabulek	49
Seznam použitých zkratk a matematických symbolů	50
Příloha 1 - seznam funkcí adresáře Sítě	52

Úvod

Cílem této práce je popsat výpočty proudění nestlačitelných tekutin v reálných geometriích, což zahrnuje získání dostatečně přesné a hladké výpočetní sítě z CT skenu, zformulování rovnic včetně zadání vhodných okrajových a počátečních podmínek až po nalezení slabé formulace úlohy a její diskretizace metodou konečných prvků. Součástí práce jsou numerické výsledky, kde je proudění aplikováno na geometrie mozkových cév postižených aneurysmatem.

Aneurysma, česky výduť, je lokální rozšíření tepny, nejčastěji abdominální nebo mozkové. Je to onemocnění, které se nejčastěji vyskytuje na slabých stěnách tepen v místech, kde dochází k největšímu napětí stěn. Aneurysma je nebezpečné v případě, kdy dojde k jeho prasknutí. Pak dojde k vylití krve a může dojít i k zablokování tepny pevnými částmi aneurysmatu. [26]

S rostoucí oblibou CT skenu je často diagnostikováno aneurysma, ale riziko prasknutí je malé. Vystává tedy otázka, která aneurysmata jsou náchylná k prasknutí. Ve studiu vzniku a vývoje aneurysmat hraje velkou roli lokální smykové napětí na stěnách. [7] Potřeba výpočtu přesného rychlostního pole a rozložení tlaku a smykového napětí v cévách postižených aneurysmatem je motivována právě otázkou, které aneurysma může být náchylné k prasknutí a je třeba je sledovat. Problém proudění v geometriích cév postižených aneurysmatem se v současnosti zabývá řada článků, mezi nimi [22], [23], [28].

K řešení tohoto problému je použita metoda konečných prvků, což je univerzální metoda pro řešení variačně formulovaných diferenciálních rovnic. Výhodou použití této metody je hlavně její aplikovatelnost na široké spektrum rovnic, možnost řešení úloh pro obecný geometrický tvar těles a možnost kontrolovat vlastnosti diskretizace výběrem aproximujících konečně dimenzionálních prostorů.

V této práci jsou nejprve v první kapitole představeny použité rovnice a diskutovány možnosti zadání okrajových podmínek. Rovnice jsou pak převedeny na variační tvar a je uvedena jejich časová a prostorová diskretizace.

Hlavním teoretickým výsledkem této práce je popis získávání výpočetních sítí z CT skenu, možností jejich zhlazení a úprav pro zadání okrajových podmínek, což je obsahem druhé kapitoly.

Konečně ve třetí kapitole jsou představeny získané sítě. Všechna naměřená data včetně segmentací cév byla poskytnuta v rámci spolupráce s MUDr. Alešem Hejčlem, PhD. z Neurochirurgické kliniky Masarykovy nemocnice v Ústí nad Labem. V sekci 3.1 je diskutována kvalita těchto sítí a možnosti vylepšení. Na těchto sítích jsou dále uvedena spočtená rychlostní a tlaková pole pomocí různých konečných prvků, které jsou časově srovnány v sekci 3.2. Dopočteno je i normálové a smykové napětí na stěnách, jak je uvedeno v sekci 3.4. Pro kontrolu a porovnání efektivnosti řešení použitého akademického softwaru jsou pak v sekci 3.3 uvedeny i výsledky napočtené volně šiřitelným programem FEniCS.[18]

1. Matematický popis proudění v cévách

1.1 Popis problému

Jak již bylo zmíněno v úvodu, práce se zabývá prouděním nestlačitelných tekutin v geometriích mozkových aneurysmat. Většina specializovaných obvodů oběhové soustavy jako je mozkový obvod je postavena paralelně vedle sebe, takže každý orgán dostává plně okysličenou krev. Oběhová soustava člověka slouží k rychlému konvektivnímu transportu kyslíku, živin a odpadních látek, hormonů a tepla po celém těle. Vaskulární systém roznese kyslík do celého těla, difuzně pak na velmi krátké vzdálenosti.[17]

Krev je viskózní nestlačitelná neneutonská kapalina. V této práci je pro popis proudění použit pouze newtonovský model s konstantní viskozitou. Cílem této práce je zobrazení výpočtů proudění na sítích získaných z reálných dat, k čemuž není potřeba sofistikovanějšího modelu. Přidání závislosti viskozity na rychlosti smyku je uvažováno jako budoucí práce.

Z použitých Navierových-Stokesových rovnic je získána informace o rozložení rychlosti a tlaku a je možné dopočítat i normálová a smyková napětí na stěně cévy. Cévní stěna je tvořena třemi částmi, vnější adventitia obsahuje spojovací tkáň a nervová vlákna, prostřední media hladké svalstvo a kolagen a vnitřní intima je vystlána endoteliálními buňkami.

„Endoteliální buňky cévní stěny jsou mechanicky citlivé na lokální smykové napětí a přeměňují abnormální smykové napětí na stěnách na biochemické signály, které upravují buněčnou strukturu buňky. To má za následek mimo jiné ztenčení stěny aneurysmatu.“ [16]

Tvar a umístění každého aneurysmatu je jedinečné, proto je při výpočtu rozložení tlaku a smykového napětí na stěnách užitečné použití geometrií získaných z CT skenu. V následující sekci uvedeme použité rovnice včetně zvolených okrajových a počátečních podmínek.

1.2 Zadání rovnic a okrajových podmínek

V této sekci představíme použité Navierovy-Stokesovy rovnice a okrajové podmínky pro rychlost. Zadáváme okrajové podmínky pro rychlost, protože vedle geometrie jsme schopni od neurochirurgů získat i změnu rychlosti krve v závislosti na čase, ale nemáme informaci o tlaku.

Na vstupu používáme Dirichletovu okrajovou podmínku s časovou závislostí odvozenou od skutečné závislosti rychlosti na čase a prostorovou závislostí napodobující klasický parabolický vstupní profil pro laminární proudění.

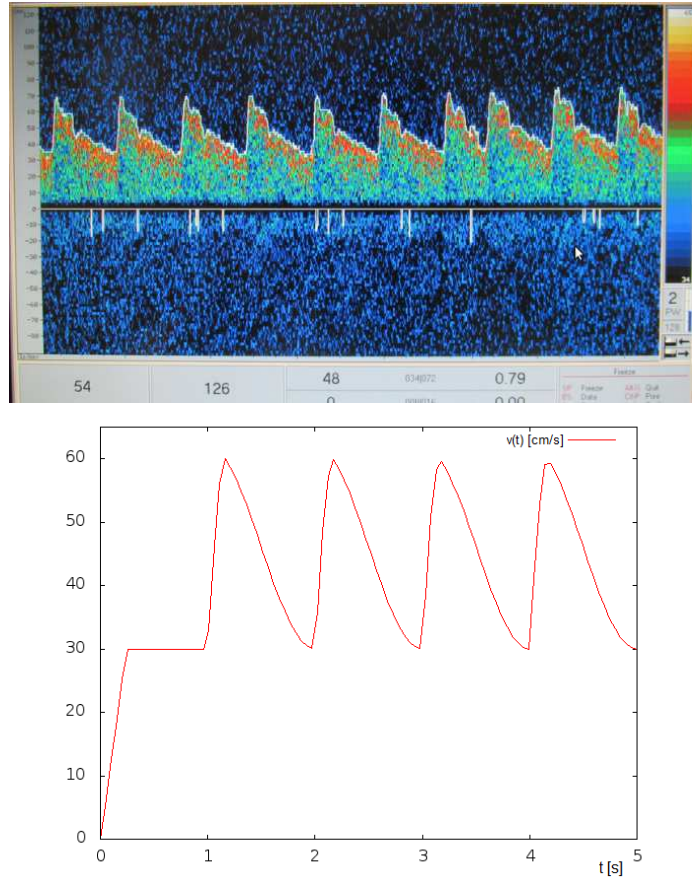
Pro výpočet použijeme Navierovy-Stokesovy rovnice

$$\begin{aligned}
 \frac{\partial u}{\partial t} + (\nabla u)u - \nu \Delta u + \nabla p &= \mathbf{f} & \text{v } (0, T) \times \Omega, \\
 \operatorname{div} u &= 0 & \text{v } (0, T) \times \Omega, \\
 u &= \mathbf{0} & \text{na } (0, T) \times \Gamma_0, \\
 u &= u_{in} & \text{na } (0, T) \times \Gamma_{in}, \\
 \mathbf{T}n &= \mathbf{0} & \text{na } (0, T) \times \Gamma_{out}, \\
 u &= u_0 & \text{na } \{0\} \times \Omega,
 \end{aligned} \tag{1.1}$$

kde Ω je omezená oblast v \mathbf{R}^3 s po částech polygonální hranicí prvního stupně, Γ_{in} je vstupní rovina oblasti, Γ_{out} je sjednocení rovin oblasti pro výstup a $\partial\Omega = \Gamma_0 \cup \Gamma_{in} \cup \Gamma_{out}$. u a p jsou hledaná rychlostní a tlaková pole, \mathbf{T} je Cauchyho tenzor napětí daný vztahem (1.2) s $\mathbf{L} = \nabla u$, $\nu = 2,8 \cdot 10^{-3} g s^{-1} mm^{-1}$ je konstantní dynamická viskozita a n je pole vnějších jednotkových normál pro výstupní roviny. Počáteční podmínku bereme nulovou $u_0 = 0$ a okrajová podmínka $u_{in} = v_t(t)u_X(x, y, z)$ bude zadána níže. Vnější síly zanedbáváme, tedy $\mathbf{f} = 0$.

$$\mathbf{T} = -pI + \nu(\mathbf{L} + \mathbf{L}^T) \tag{1.2}$$

Skutečná křivka zachycující průběh velikosti rychlosti v závislosti na čase a použitá aproximující křivka $v(t)$ jsou vidět na obrázku 1.1.



Obrázek 1.1: Skutečná závislost rychlosti v [cm/s] na čase v [s] a použitá závislost rychlosti u_t v [cm/s] na čase v [s].

Během času $t \in [0, 1]$ se funkce $v_t(t)$ dostane do hodnoty 30cm/s, pak ji v každém dalším intervalu aproximujeme goniometrickými funkcemi. Pro maximální dosaženou výšku v_{max} a počáteční hodnotu periodického proudění v_{min} je funkce zadána

$$v_t(t) = \begin{cases} \min(6(v_{max} - v_{min})t, v_{min}) & t < 1, \\ (v_{max} - v_{min}) \sin(3\pi s) + v_{min} & t \geq 1, s \in [0, \frac{1}{6}], \\ (v_{max} - v_{min}) \frac{\cos(\pi s) + 1}{1 + \frac{\sqrt{3}}{2}} + v_{min} & t \geq 1, s \in [\frac{1}{6}, 1], \end{cases} \quad (1.3)$$

kde $s = [t]$ je dolní celá část aktuálního času t .

Pro funkci v_t z obrázku 1.1 je $v_{max} = 60$ a $v_{min} = 30$.

Vzestup pro $t < 1$ je určen tak, aby byl stejně strmý jako vzestup na intervalu $[s, s + \frac{1}{6}]$, aby pro každý časový krok řešení předchozího bylo dobrým počátečním přiblížením pro Newtonovu metodu. V čase $t = 0$ začínáme nulovou hodnotou, bereme proto za počáteční přiblížení $u_0 = 0$.

Protože používáme nulovou Dirichletovu podmínku na krajních stěnách oblasti, je vhodné na stěně vstupu zadat analogii parabolického vstupu. K tomu vytvoříme funkci u_X s hodnotami

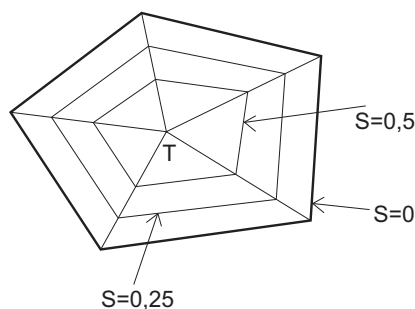
$$u_X(x, y, z) = \begin{cases} 0 & \text{na hranici vstupu} \\ 1 & \text{v těžišti.} \end{cases}$$

Výsledná zadaná vstupní rychlost je pak rovna $u_{in}(x, y, z, t) = v_t(t)u_X(x, y, z)$.

Zbývá tedy předepsat funkci u_X , jejímž definičním oborem jsou vrcholy na vstupu. Krajní vrcholy tohoto vstupu tvoří konvexní mnohoúhelník, což je hlavní předpoklad pro dále popsanou konstrukci. Mnohoúhelník označíme M s hranicí ∂M .

Začneme tím, že rovinu vstupu posuneme do roviny $z = 0$ tak, aby těžiště mnohoúhelníku bylo v počátku soustavy souřadnic. To je výhodné hlavně pro vizualizaci výsledku a zjednodušení popisu, není to však nutné pro další postup.

Každý vrchol $X[x, y, z]$ vstupní roviny spojíme s těžištěm T a označíme $Y \Rightarrow TX \cap \partial M$. Bodu X přiřadíme parametr $s = \frac{|YX|}{|TX|}$ takový, že na úsečce TY platí, že v bodě Y je $s = 0$, v bodě T je $s = 1$. Každému bodu na vstupu takto přiřadíme parametr podle vzdálenosti od těžiště, jak je ukázáno na obrázku 1.2.



Obrázek 1.2: Rovina vstupu s čarami spojujícími body s parametry $s = 0.5$ a $s = 0.25$.

Nakonec položíme

$$u_X = s^2, \quad (1.4)$$

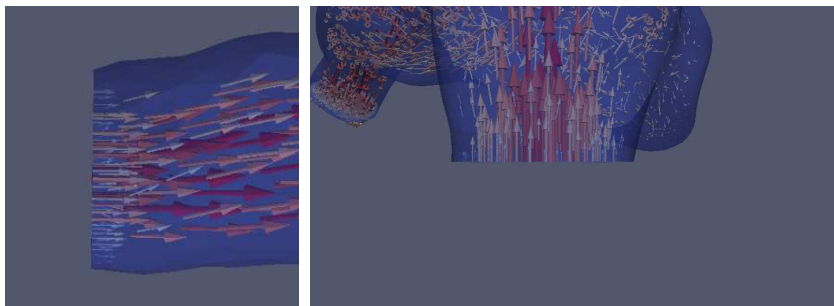
kde s je výše popsáný parametr.

Jinou možností je využít posunutí roviny vstupu do roviny $z = 0$ a skutečnosti, že těžiště krajních bodů je umístěno v počátku. Za předpokladu, že vstup roviny lze považovat za kruhovitý, je možné použít jiný parabolický profil

$$u_X(x, y, 0) = \frac{\max(r^2 - x^2 - y^2, 0)}{r^2}, \quad (1.5)$$

kde $r = \min_{A \in \partial M} \|A\|$.

Napočtená rychlostní pole u vstupu pro obě takto zvolené prostorové závislosti rychlosti na vstupu jsou ukázána na obrázku 1.3.



Obrázek 1.3: Průřezy spočtených proudění u vstupu při použití nejprve prostorové závislosti (1.4), poté (1.5).

Výše uvedenými rovnicemi a okrajovými podmínkami je úloha přesně definována a můžeme pokračovat převodem rovnic na variační tvar potřebný pro metodu konečných prvků.

1.3 Odvození variační formulace

Základem metody konečných prvků je variační formulace a následná diskretizace rovnic (1.1). K tomu je zapotřebí nejprve představit základní prostory, kterými jsou Lebesgueův prostor $L^2(\Omega)$ a Sobolevův prostor $H^1(\Omega)$ včetně jejich rozšíření v třírozměrném prostoru $[L^2(\Omega)]^3$ a $[H^1(\Omega)]^3$, kde tedy

$$\begin{aligned} L^2(\Omega) &= \{v : \Omega \rightarrow \mathbf{R} \text{ měřitelná funkce } \int_{\Omega} v^2 dx < \infty\}, \\ H^1(\Omega) &= \{v \in L^2(\Omega), \frac{\partial v}{\partial x} \in L^2(\Omega)\}, \\ [L^2(\Omega)]^3 &= \{v = (v^1, v^2, v^3) : \Omega \rightarrow \mathbf{R}^3, v^j \in L^2(\Omega), j = 1, 2, 3\}, \\ [H^1(\Omega)]^3 &= \{v = (v^1, v^2, v^3) : \Omega \rightarrow \mathbf{R}^3, v^j \in H^1(\Omega), j = 1, 2, 3\}. \end{aligned} \tag{1.6}$$

1.3.1 Variační formulace pro stacionární rovnice

Uvažujme stacionární Navier-Stokesovy rovnice

$$\begin{aligned} (\nabla u)u - \nu \Delta u + \nabla p &= \mathbf{0} & \text{v } (0, T) \times \Omega, \\ \operatorname{div} u &= 0 & \text{v } (0, T) \times \Omega, \\ u &= \mathbf{0} & \text{na } (0, T) \times \Gamma_0, \\ u &= u_{in} & \text{na } (0, T) \times \Gamma_{in}, \\ \mathbf{T}n &= \mathbf{0} & \text{na } (0, T) \times \Gamma_{out}, \end{aligned} \tag{1.7}$$

kde $u, p, \nu, \mathbf{T}, u_{in}$ jsou stejné jako v rovnici (1.1) a bereme okrajové podmínky (1.3) a (1.5).

Budeme hledat dvojici řešení $(u, p) \in V \times Q$, kde prostory V a Q uvažujeme takové, že

$$\begin{aligned} V &= \{v \in [H^1(\Omega)]^3, v = 0 \text{ na } \Gamma_0, v = u_{in} \text{ na } \Gamma_{in}\}, \\ Q &= L^2(\Omega). \end{aligned} \tag{1.8}$$

Pro testovací funkce je třeba použít lineární vektorový prostor, vezmeme tedy

$$\widehat{V} = \{v \in [H^1(\Omega)]^3, v = 0 \text{ na } \Gamma_0 \cup \Gamma_{in}\}. \tag{1.9}$$

Přenosobním první rovnice testovací funkcí $v \in \widehat{V}$, integrací přes oblast Ω a použitím per-partes získáváme

$$-\nu \int_{\partial\Omega} \frac{\partial u}{\partial n} v dS + \nu \int_{\Omega} \nabla u \nabla v dx + \int_{\Omega} (\nabla u)u v dx + \int_{\partial\Omega} p n \cdot v dS - \int_{\Omega} p \operatorname{div} v dx = 0.$$

Oba integrály přes hranici oblasti jsou nulové díky tvaru testovacího prostoru \widehat{V} a Neumannovým okrajovým podmínkám.

Druhou rovnici přenásobíme testovací funkcí $q \in Q$. Integrací získáme

$$-\int_{\Omega} \operatorname{div} u q \, dx = 0.$$

Definice 1. Slabým řešením rovnic (1.7) nazveme dvojici $(u, p) \in V \times Q$ splňující

$$\begin{aligned} \nu \int_{\Omega} \nabla u \nabla v \, dx + \int_{\Omega} (\nabla u) u v \, dx - \int_{\Omega} p \operatorname{div} v \, dx &= 0 \quad \forall v \in \widehat{V}, \\ - \int_{\Omega} \operatorname{div} u q \, dx &= 0 \quad \forall q \in Q. \end{aligned} \quad (1.10)$$

1.3.2 Diskretizace časové derivace

Pro diskretizaci časové derivace začneme rozdělením časového intervalu na časové kroky, kde t^k značí k -tý časový krok.

Velikost časového kroku $\Delta t = \Delta t_k = t_{k+1} - t_k$ bereme konstantní pro všechny časy k .

Časová derivace

$$\frac{\partial u}{\partial t} = F$$

je aproximována Crankovo-Nicholsonové metodou jako

$$0 = \frac{\partial u}{\partial t} - F \approx \frac{u^{k+1} - u^k}{\Delta t} - \frac{1}{2} [F^{k+1} + F^k], \quad (1.11)$$

kde $u^k = u(t^k)$ a $F^k = F(t^k)$.

Tato metoda je metodou druhého řádu je vhodná pro diskretizaci Navierových-Stokesových rovnic [13].

V případě rovnic (1.1) je i -tá složka funkce F rovna

$$F_i = \nu \Delta u_i - (\nabla u_i) u - \frac{\partial p}{\partial x_i}, \quad i = 1, 2, 3.$$

Dosazením schéma (1.11) do rovnic (1.1) získáme rovnice

$$(u_i^{k+1} - u_i^k) = \frac{\Delta t}{2} \left[\nu (\Delta u_i^{k+1} + \Delta u_i^k) - [(\nabla u_i^{k+1}) u^{k+1} + (\nabla u_i^k) u^k] - 2 \frac{\partial p^{k+1}}{\partial x_i} \right],$$

kde $i = 1, 2, 3$ a tlak bereme na každém časovém kroku konstantní, proto $p^k = p^{k+1}$.

Tuto rovnici opět přenásobíme testovací funkcí $v \in \widehat{V}$, kde \widehat{V} je definován v (1.9) a podobně jako pro stacionární problém získáme rovnice (1.12).

$$\begin{aligned} \int_{\Omega} (u_i^{k+1} - u_i^k) v_i - \frac{\Delta t}{2} \nu \int_{\Omega} (\nabla u_i^{k+1} + \nabla u_i^k) \nabla v_i \, dx \\ - \frac{\Delta t}{2} \int_{\Omega} [(\nabla u_i^{k+1}) u^{k+1} + (\nabla u_i^k) u^k] v_i \, dx - \Delta t \int_{\Omega} \frac{\partial p^{k+1}}{\partial x_i} v_i \, dx = 0 \quad \forall v \in \widehat{V}, \end{aligned} \quad (1.12)$$

pro $i = 1, 2, 3$, kde $k = 0, 1, 2, \dots$ a $u^0 = p^0 = 0$.

Druhá rovnice (1.10) zůstává formálně stejná

$$- \int_{\Omega} \operatorname{div} u^{k+1} q \, dx = 0 \quad \forall q \in Q, \quad (1.13)$$

kde Q je definováno v (1.8).

Definice 2. Slabým řešením úlohy (1.1) v k -tém časovém kroku nazveme dvojici $(u^{k+1}, p^{k+1}) \in V \times Q$ splňující rovnice (1.12) a (1.13), kde V, Q jsou definovány v (1.8), $k = 0, 1, 2, \dots$ a $u^0 = p^0 = 0$.

1.4 Diskretizace rovnic metodou konečných prvků a použitý software

Abychom mohli rovnice (1.1) řešit metodou konečných prvků, je třeba diskretizovat prostory V, Q (1.8) a rovnice (1.12) a (1.13).

Nejprve v této sekci ukážeme diskretizaci rovnic, konkrétní použité prostory pak představíme v sekci 1.5.

Hledanou rychlost u můžeme vyjádřit ve tvaru $u = \bar{u} + \hat{u}$, kde $\hat{u} \in \widehat{V}$ a $\bar{u} \in V$ je dané a splňuje okrajové podmínky. Těchto \bar{u} je nekonečně mnoho, ale výsledný tvar u nezávisí na jeho výběru. [11] V následujícím ztotožníme $u := \hat{u}$.

Mějme konečně dimenzionální prostory \widehat{V}_h a Q_h , které aproximují prostory \widehat{V} a Q . Pak lze rovnice (1.12) a (1.13) přeformulovat ve smyslu následující definice.

Definice 3. Řekneme, že dvojice (u_h^{k+1}, p_h^{k+1}) řeší diskrétní variační úlohu, pokud $(u_h^{k+1}, p_h^{k+1}) \in V_h \times Q_h$ a platí

$$\int_{\Omega} (u_{hi}^{k+1} - u_{hi}^k) v_i - \frac{\Delta t}{2} \nu \int_{\Omega} (\nabla u_{hi}^{k+1} + \nabla u_{hi}^k) \nabla v_i \, dx$$

$$- \frac{\Delta t}{2} \int_{\Omega} [(\nabla u_{hi}^{k+1}) u_h^{k+1} + (\nabla u_{hi}^k) u_h^k] v_i \, dx - \Delta t \int_{\Omega} \frac{\partial p_h^{k+1}}{\partial x_{hi}} v_i \, dx = 0 \quad \forall v \in \widehat{V}_h,$$

pro $i = 1, 2, 3$, kde $k = 0, 1, 2, \dots$ a $u^0 = 0$,

$$- \int_{\Omega} \operatorname{div} u_h^{k+1} q \, dx = 0 \quad \forall q \in Q_h. \quad (1.14)$$

Označme $\{\phi_i\}_{i=1}^N$ bázi prostoru \widehat{V}_h , kde $N = \dim \widehat{V}_h$ a $\{\theta_i\}_{i=1}^M$ bázi prostoru Q_h s $M = \dim Q_h$.

Za testovací funkce diskrétní variační úlohy (1.14) bereme báze diskrétních aproximujících prostorů.

Označme

$$\begin{aligned} u_{hi} &= \sum_{j=1}^N U_{ij} \phi_j \\ p_h &= \sum_{l=1}^M P_l \theta_l \end{aligned} \quad (1.15)$$

$i = 1, 2, 3$, kde U_{ij} nazveme souřadnice i -té složky funkce u_h a P_l souřadnice funkce p_h .

Úlohu (1.14) lze pak zapsat jako $\mathbf{F}(U_{1j}, U_{2j}, U_{3j}, P_l) = 0$.

Získáváme tedy v každém časovém kroku nelineární algebraickou soustavu $3N+M$ rovnic pro $3N+M$ neznámých. Ta je řešena pomocí Newtonovy metody, která však nemusí konvergovat při nevhodně zvoleném počátečním přiblížení. V prvním kroku bereme nulové počáteční přiblížení, což je i přesným řešením, v každém dalším bereme za počáteční řešení výsledek předchozího časového kroku, což může vést k divergenci pro nevhodně zvolený časový krok.

Jeden krok Newtonovy metody pro úlohu $\mathbf{F}(\mathbf{X}) = \mathbf{0}$ lze zapsat jako

$$\mathbf{X}^{n+1} = \mathbf{X}^n - J^{-1} \mathbf{F}(\mathbf{X}^n), \quad (1.16)$$

kde $J = \frac{\partial \mathbf{F}}{\partial \mathbf{X}}(\mathbf{X}^n)$ je Jacobiho matice, která je spočtena metodou konečných diferencí

$$J_{ij} = \frac{\mathbf{F}_i(\mathbf{X}^n + \alpha \mathbf{e}_j) - \mathbf{F}_i(\mathbf{X}^n - \alpha \mathbf{e}_j)}{2\alpha}, \quad (1.17)$$

kde \mathbf{e}_j jsou vektory kanonické báze v \mathbf{R}^{3N+M} a $\alpha \in \mathbf{R}$.[2]

Newtonova metoda vede k soustavě lineárních rovnic. Pro jejich řešení je použit LU rozklad implementovaný v knihovně MUMPS [3].

K numerickým výpočtům byl použit program vyvinutý na Matematickém ústavu MFF UK pro řešení nestlačitelných Navier-Stokesových rovnic Fstrin [2], který pro řešení lineárního algebraického problému využívá knihovnu PETSc [6].

1.5 Použité konečné prvky a stupně volnosti

Abychom mohli představit použité konečné prvky pomocí referenčního konečného prvku, je třeba nejprve zadefinovat, co myslíme konečným prvkem a afinní ekvivalencí. Potom můžeme přejít k popisu tří konečných prvků použitých v této práci.

Definice 4. Konečným prvkem rozumíme trojici (K, V, \mathcal{L}) , kde K je omezená uzavřená podmnožina \mathbf{R}^3 s neprázdným vnitřkem a po částech spojitou hranicí, V je prostor funkcí definovaných na K dimenze n a konečně $\mathcal{L} = l_1, l_2, \dots, l_q(n)$ je množina stupňů volnosti definovaných na K taková, že $\text{span } \mathcal{L} = V^*$. [11]

Definice 5. Řekneme, že dva konečné prvky $(\widehat{K}, \widehat{V}, \widehat{\mathcal{L}})$ a (K, V, \mathcal{L}) jsou afinně ekvivalentní, pokud existuje regulární afinní zobrazení $F_K : \widehat{K} \rightarrow K$.

Definice 5 nám umožní definovat stupně volnosti na daném referenčním prvku. Libovolný prvek lze z referenčního prvku odvodit, pokud je s ním afinně ekvivalentní.[11]

Všechny použité prvky jsou simplicialní, a tedy množina K je čtyřstěn.

Pro každý ze tří použitých prvků uvedeme konečně dimenzionální prostory funkcí V_h a Q_h definované na celé oblasti Ω , které aproximují prostory V a Q a na referenčním prvku ukážeme lokální stupně volnosti pro daný prvek. Pro prostor \widehat{V}_h platí vztah (1.18).

$$\widehat{V}_h = V_h \cup v_h = 0 \text{ na } \Gamma_0 \cup \Gamma_{in}. \quad (1.18)$$

Přesnější definice použitých prvků lze nalézt v [18].

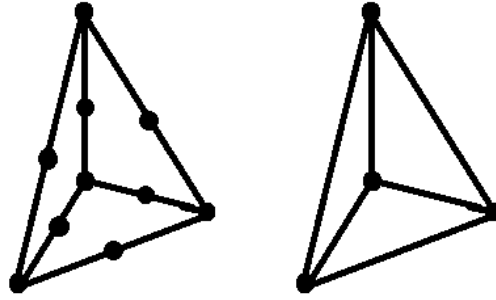
Taylorův-Hoodův prvek (TaHo)

Tento prvek, také často značený P_2P_1 prvek, používá konečně dimenzionální aproximaci prostoru V a Q danou vztahy (1.19).

$$V_h = \left\{ v_h \in [C^1(\Omega)]^3, v_h|_K \in [P_2(K)]^3 \quad \forall K \in T_h \right\}, \quad (1.19)$$

$$Q_h = \{ v_h \in C^0(\Omega), v_h|_K \in P_1(K) \quad \forall K \in T_h \}.$$

Lokálními stupni volnosti na jednom prvku jsou pro každou složku rychlosti hodnoty funkce ve vrcholech a středech stran, pro tlak pouze hodnoty funkce ve vrcholech, jak je vidět na obrázku 1.5. Lokální počet stupňů volnosti pro tento prvek je 34.



Obrázek 1.4: Prvek TaHo: Lokální stupně volnosti pro rychlost a tlak.

Crouzeixův-Raviartův prvek (CrRa)

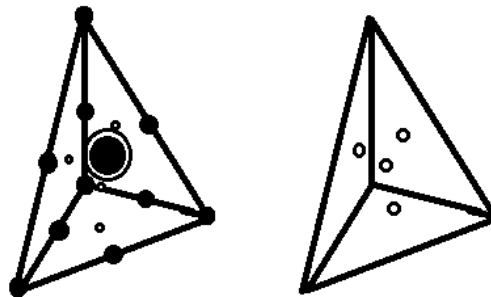
Tento prvek, také označovaný $P_2^{++}P_1^{disc}$ prvek, používá konečně dimenzionální aproximaci prostoru V a Q danou vztahy (1.20).

$$V_h = \left\{ v_h \in [C^1(\Omega)]^3, v_h|_K \in [P_2^{++}(K)]^3 \quad \forall K \in T_h \right\} \quad (1.20)$$

$$Q_h = \left\{ v_h \in L^2(\Omega), v_h|_K \in P_1(K) \quad \forall K \in T_h \right\}$$

Prostor $[P_2^{++}(K)]^3 = [P_2(K)]^3 \oplus [B_1(K)]^3 \oplus [B_2(K)]^3$, kde B_1 má v \mathbf{R}^3 dimenzi 1 a za stupeň volnosti bere hodnotu funkce v těžišti a B_2 má v \mathbf{R}^3 dimenzi 4 a za stupně volnosti bere hodnoty funkce ve středech stran.

Lokální stupně volnosti pro Crouzeixův-Raviartův prvek jsou vidět na obrázku 1.5, jejich počet pro tři složky rychlosti a tlak je 49.



Obrázek 1.5: Prvek CrRa: Lokální stupně volnosti pro rychlost a tlak.

MINI element (MINI)

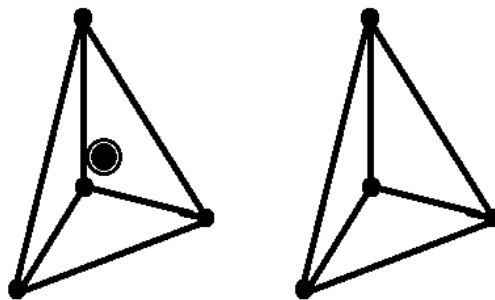
Tento prvek, také značený jako $P_1^+ P_1$, byl zaveden v [4] pro řešení Stokesova problému pro smíšenou metodu konečných prvků. Používá konečně dimenzionální aproximaci prostoru V a Q danou vztahy (1.21).

$$V_h = \left\{ v_h \in [C^0(\Omega)]^3, v_h|_K \in [P_1^+(K)]^3 \quad \forall K \in T_h \right\} \quad (1.21)$$

$$Q_h = \{ v_h \in C^0(\Omega), v_h|_K \in P_1(K) \quad \forall K \in T_h \}$$

Prostor $[P_1^+(K)]^3 = [P_1(K)]^3 \oplus [B_1(K)]^3$, kde B_1 má v \mathbf{R}^3 dimenzi 1 a za stupeň volnosti bere hodnotu funkce v těžišti.

Lokální stupně volnosti pro MINI prvek jsou vidět na obrázku 1.5, jejich počet pro tři složky rychlosti a tlak je 19.



Obrázek 1.6: Prvek MINI: Lokální stupně volnosti pro rychlost a tlak.

2. Příprava sítě

V této kapitole se zaměříme na přípravu sítě pro metodu konečných prvků získanou z reálných dat. Kvalita sítě má velký vliv na přesnost a účinnost výpočtů založených na řešení parciálních diferenciálních rovnic. Pro metodu konečných prvků se v odhadech interpolační chyby přirozeně vyskytují některé indikátory kvality sítě, které závisí na tvaru sítě, velikosti elementů a úhlech triangulace [15].

Nejprve popíšeme postup získání CT skenu a voxelové reprezentace dat, v dalších sekcích představíme software zabývající se teselací a zhlazováním sítí a ukážeme také možnosti zhlazování na úrovni voxelů. Na závěr této kapitoly budeme diskutovat problém definování vstupu a výstupu na reálné geometrii pro zadávání okrajových podmínek. K tomu budeme potřebovat zadefinovat některé základní pojmy.

Definice 6. Voxelem (zkratka pro *volumetric element*) rozumíme konečný objem reprezentovaný kvádrem (nejčastěji pravidelným čtyřbokým hranolem), jehož rozměry jsou dány rozlišením pixelu a tloušťkou jednotlivého řezu CT skenu.

Definice 7. Pixel (zkratka pro *picture element*) je nejmenší rozlišitelný prvek dvourozměrného obrazu, který nese základní jednotku informace odpovídající většinou intenzitě dopadajícího paprsku.[27]

Definice 8. Segmentace obrazu je proces, při němž je každý voxel z objemových dat označen buď jako součást segmentovaného objektu nebo jako pozadí. [5]
V případě segmentace cévy jde o rozdělení na část uvnitř cévy s nenulovou hodnotou intenzity ve voxelu a vně cévy s hodnotou 0. Segmentace je v tomto textu ztotožňována s tou částí dat, která odpovídá části uvnitř cévy.

Definice 9. Binárními daty nebo také černobílými daty rozumíme data v segmentaci, která nabývají pouze dvou hodnot, nejčastěji nul a jedniček.

Definice 10. Šedými daty nebo také rozmazanými daty rozumíme ta data, která mají spojitý obor hodnot.

2.1 Získání voxelové sítě

V této části se budeme zabývat získáním CT skenu a jeho zpracováním na pole černobílých dat vyjadřující umístění voxelů v třírozměrném obrazu.

Při skenování projde mozkem pod různými úhly velké množství rentgenových paprsků. Výsledný CT sken je soubor dvourozměrných řezů, kde každý reprezentuje konkrétní tloušťku celkového obrazu. Řezy zobrazují propustnost materiálu na jednotlivých pixelech vyjádřenou CT číslem, které měří koeficient zeslabení paprsku při průchodu tkání uvnitř každého voxelu.[14]

Koeficient zeslabení popisuje, do jaké míry se intenzita energie paprsku snižuje při průchodu konkrétním materiálem. Velký koeficient zeslabení znamená, že paprsek při průchodu zeslábne rychle, malý, že je materiál vůči paprsku relativně transparentní. Při použití pro rentgenové paprsky se značí μ a má jednotku cm^{-1} .

Míře zeslabení rentgenového paprsku se říká také rentgenová hustota. CT číslo se škáluje pomocí Haunsfieldovy stupnice, která dává destilované vodě za normálního tlaku a teploty hodnotu 0HU a vzduchu -1000HU. Pro materiál s koeficientem zeslabení μ je pak

$$\text{HU} = 1000 \frac{\mu - \mu_{\text{voda}}}{\mu_{\text{voda}} - \mu_{\text{vzduch}}}.$$

V této stupnici má např. tuk hodnotu -100HU, sval a krev +40HU a kosti přes +400HU. [20]

Takto získané řezy jsou dvourozměrnými poli celých čísel odpovídající celé skenované oblasti mozku. Síť pro metodu konečných prvků se získává ze segmentace cévy s aneurysmatem, která je vyjádřena třírozměrným binárním polem. Označme toto pole $I(i, j, k)$ a segmentaci S . Ztotožníme-li voxel s jeho pořadím v segmentaci (i, j, k) , platí, že

$$I(i, j, k) = \begin{cases} 1, & (i, j, k) \in S, \\ 0, & \text{jinak.} \end{cases}$$

Nyní si projdeme získání pole $I(i, j, k)$ z dat získaných CT skenem. Následující odstavce jsou napsány podle [9].

Vycházíme z třírozměrného pole indexovaného (i, j, k) , kde pro každou hladinu k máme jeden řez CT skenu vyplněný celočíselnými CT čísly. První a druhý rozměr pole jsou dány počtem pixelů, třetí je dán počtem řezů. Tato čísla převedeme na reálná čísla v intervalu $[0, 1]$. Získanou matici označme $I_0(i, j, k)$, z ní pak nejprve vyřízneme tu část skenu, která nás zajímá. Označme ji R . Získáme tak menší pole

$$I_1(i, j, k) = \begin{cases} I_0(i, j, k), & (i, j, k) \in R, \\ 0, & \text{jinak.} \end{cases}$$

Dalším krokem je zostření obrazu a zvýšení kontrastu mezi světlejšími a tmavšími voxely. Prvky výsledné matice $I_2(i, j, k)$ jsou stále v rozmezí $[0, 1]$, pro segmentaci však potřebujeme binární data. V dalším kroku tedy následuje vymezení intervalu $R_1 = [I_{\min}, I_{\max}]$, hodnot pole, která zůstanou v segmentaci ve smyslu

$$I_3(i, j, k) = \begin{cases} 1, & I_2(i, j, k) \in R_1, \\ 0, & \text{jinak.} \end{cases}$$

Interval R_1 vymezuje určitou intenzitu obrazu, kterou chceme z obrazu vyříznout. Začneme s počátečním voxel (seed voxel) x_0 a meze intervalu R_1 určíme tak, aby zahrnoval všechny voxely ze stejné topologické oblasti obrazu jako x_0 . Odhady těchto mezí nejsou automatizovány a je třeba přistupovat individuálně ke každé segmentaci.

Následuje zhlazení pomocí diskrétního konvolučního operátoru. Označme pro každý voxel jeho šest nejbližších sousedů N_0 a N_1 označme dvacet voxelů, které s ním sousedí přes vrchol nebo přes hranu. Konvoluční jádro zhlazení má rozměry $3 \times 3 \times 3$ a pro voxel se souřadnicemi (x, y, z) má tvar

$$I_4(i, j, k) = \begin{cases} -26, & (i, j, k) = (x, y, z), \\ 2, & (i, j, k) \in N_0, \\ 1, & (i, j, k) \in N_1. \end{cases}$$

Zhlazování pomocí konvolučních operátorů se věnuje sekce 2.3.1.

Takto získaná data stále obsahují nerovnosti a šumy, které lze odstranit statistickým filtrováním. Označme S_1 segmentaci voxelů s hodnotou 1 a S_0 zbylé voxelů s hodnotou 0. Platí

$$I_5(i, j, k) = \begin{cases} 1, & \sum_{x \in N_0} [x \in S_1] > 3, \\ 0, & \sum_{x \in N_0} [x \in S_0] > 3, \\ I_4(i, j, k), & \sum_{x \in N_0} [x \in S_1] = 3, \end{cases}$$

kde

$$[x \in S] = \begin{cases} 1, & x \in S, \\ 0, & \text{jinak.} \end{cases}$$

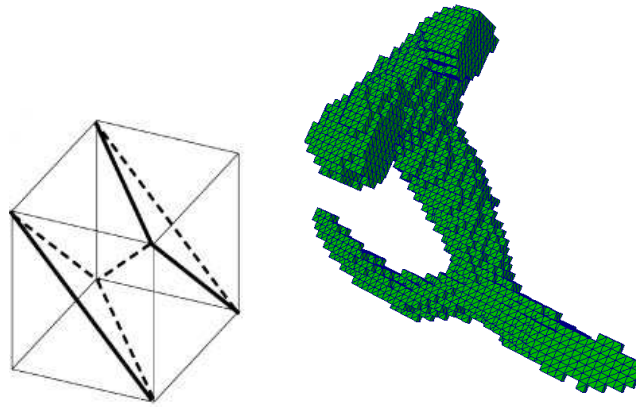
Často se také používá operátor, který zaplňuje drobné konkávní kraje obrazu. Tato operace je užitečná pro odstranění dalších šumů, může však vést až k zaplnění všech konkávních částí celého obrazu.[9]

Voxelová segmentace $I(i, j, k)$ stále nemusí být ideální pro další zpracování, proto se používá i vykreslování voxelové segmentace a interaktivní odstraňování jednotlivých voxelů nebo i celých vrstev pro získání rovného vstupu a výstupu. Tato možnost je diskutována v sekci 2.4.

2.2 Získání sítě

V minulé kapitole jsme popsali získání voxelové reprezentace dat. Následuje proces teselace, tedy generování trojúhelníkové sítě. Na obrázku 2.1 je ukázáno rozdělení krychle voxelu na šest čtyřstěnů a výsledná síť, kdy je toto aplikováno na každý voxel segmentace.

Takto vytvořená síť je základem pro další generování sítě. Jedním z možných přístupů je přímá teselace, jejíž popis lze najít v [9]. Jiným přístupem je vytvořit povrchy tvořené voxelů se stejnou hodnotou intenzity nazývaných zde isopovrchy, jak je popsáno v [9] a [10], který je použit i pro potřeby této práce.



Obrázek 2.1: Rozdělení krychle voxelu na 6 čtyřstěnnů a získaná voxelová síť

Pro všechny vrcholy vytvořené sítě čtyřstěnnů se určí jejich hodnota na základě intenzity v daném voxelu. Na jednotlivých hranách se pak použije lineární kombinace hodnot ve vrcholech hrany s váhou odpovídající lineárně vzdálenosti bodu od těchto vrcholů. Je předepsána hodnota intenzity, pro kterou bude generován povrch. Různé hodnoty intenzity pro generování povrchu výrazně neovlivňují teselaci, pro binární data se nejčastěji používá hodnota 0.5. U každého čtyřstěnnu se zkoumá, kolik hran obsahujících tuto hodnotu se v něm vyskytuje. Pokud je tento počet tři, resp. čtyři, bude z tohoto čtyřstěnnu vygenerován jeden, resp. dva, povrchové trojúhelníky. Trojúhelníky jsou orientovány tak, aby jejich normála ukazovala ve směru rostoucí intenzity. [9]

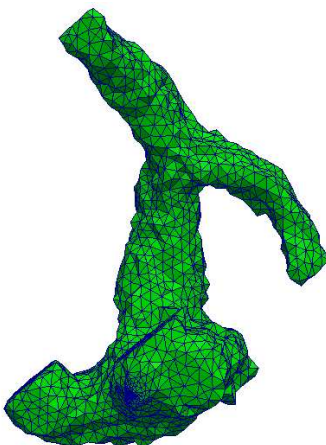
Pokud tedy hodnoty ve vrcholech jsou pouze 1 nebo 0, použijí se pouze hrany, které spojují vrcholy s různými hodnotami. Ty pak mají hodnotu 0.5 uprostřed, takže čtyřstěnnu, ze kterých jsou získávány povrchové trojúhelníky jsou rozřezávány v polovinách svých hran.

K tvorbě sítě čtyřstěnnů pro metodu konečných prvků z voxelové segmentace slouží nekomerční program iso2mesh [10] využívající volně šiřitelnou knihovnu CGAL. [1]

„Iso2mesh je víceúčelový generátor sítí na tvorbu konečně prvkové povrchové nebo objemové sítě z třírozměrných medicínských binárních nebo šedých dat.“ [10]

Dvě základní subrutiny programu použité pro generaci sítí v této práci jsou funkce vol2surf a vol2mesh, které z daných černobílých nebo šedých dat vytvoří povrchovou nebo objemovou síť. Mezi další funkce tohoto programu patří také rekonstrukce voxelové reprezentace dat z dané sítě, skripty pro vylepšování třírozměrných sítí a další.

Na obrázku 2.2 je vygenerovaná povrchová síť bez dalších úprav. Objemová síť je vytvořena pomocí programu Tetgen[25], který je externě využíván programem iso2mesh. Tento 3D generátor však nezachovává počet a pořadí povrchových trojúhelníků. Jinou alternativou je použití programu Netgen.[12] Parametry těchto sítí jsou uvedeny v sekci 3.1, úprava těchto sítí pak v sekci 2.4.



Obrázek 2.2: Nezhlazená vygenerovaná síť.

2.3 Zhlazení dat

Síť získaná segmentací dat je příliš hrubá na výpočty i vizualizaci. Proto je zapotřebí použít nějaký zhlazovací algoritmus. Ten se používá jak na úrovni voxelů, kde používáme zhlazení diskretním konvolučním jádrem, tak na úrovni sítě, kde používáme zhlazení sítě pomocí zhlazovacího filtru. V této sekci představíme obě varianty a ukážeme jejich použití na data získaná CT skenem včetně kombinace obou přístupů.

2.3.1 Zhlazení diskretním konvolučním jádrem

Zhlazení diskretním konvolučním jádrem se používá k rozostření získaného obrazu a odstranění šumů. U voxelů to znamená získání spojitého spektra hodnot, tedy přesun od binárních černobílých dat k šedým datům. Pro tento účel slouží v Matlabu funkce *smooth3*, která používá pro zhlazení třírozměrné diskretní konvoluční jádro. Používají se dva možné filtry podle použitého jádra, *gaussian* a *box*. [19]

Diskretní konvoluční jádro pro rozmazání dat má rozměry $n \times n \times n$, kde n je liché číslo. Pro oba filtry se v Matlabu zadává rozměr jádra n . Hodnoty tohoto jádra vyjadřují vliv příslušného sousedního voxelu na výslednou hodnotu konvoluce. Součet všech hodnot jádra je roven jedné.

Číslo $\frac{n-1}{2}$ vyjadřuje počet hladin sousedů, přes které se vypočítává hodnota pro daný voxel, pro $n = 3$ se je tedy hodnota počítána přes jeden sousední voxel.

Definice 11. Diskretní konvoluce pro třírozměrné pole I délky $M \times N \times P$ a jádra K o rozměrech $n \times n \times n$ je definována vztahem

$$(I * K)(x, y, z) = \sum_{i, j, k=1, \dots, n} I \left(x - \frac{n+1}{2} + i, y - \frac{n+1}{2} + j, z - \frac{n+1}{2} + k \right) K(i, j, k)$$

s tím, že pro rozměry pole I , které nejsou definovány, se bere hodnota 0.

Ukažme si tvar tohoto jádra pro $n = 3$. Kdyby sousedé neměli žádný vliv na výslednou hodnotu, platilo by v konvolučním jádru, že jediná nenulová hodnota je číslo 1 na pozici odpovídající voxelu, jak ukazuje tabulka 2.1.

0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0

Tabulka 2.1: Příklad konvolučního jádra, kdy sousedé nemají žádný vliv.

Filtr *box* průměruje hodnoty přes všechny sousedy se stejnou vahou, všechny jeho hodnoty jsou tedy rovny hodnotě $\frac{1}{n^3}$.

Pro počítání hodnot Gaussova konvolučního jádra se používá Gaussovo rozdělení dané rovnice (2.1), resp. (2.2). Zadává se hodnota směrodatné odchylky σ .

$$\widehat{G}(x) = \frac{1}{(2\pi)^{1/2}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (2.1)$$

$$G(x, y, z) = \widehat{G}(x)\widehat{G}(y)\widehat{G}(z) = \frac{1}{(2\pi)^{3/2}\sigma^3} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}} \quad (2.2)$$

Hodnoty jádra jsou různé ve středu jádra, které odpovídá počítanému voxelu označenému \widehat{V} a pro sousední voxely. Struktura jádra je zobrazena v tabulce 2.2, kde

T je označení pro daný voxel \widehat{V} ,

F značí sousední voxel \overline{V} , pro který $\overline{V} \cap \widehat{V} = \text{stěna}$,

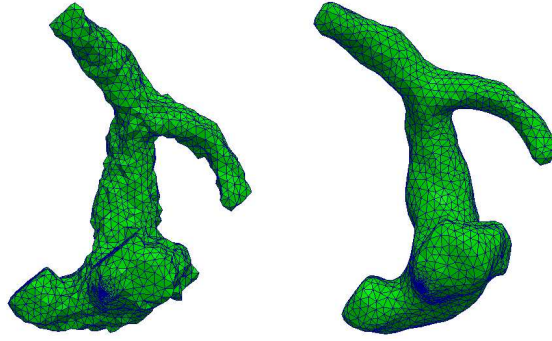
E značí sousední voxel \overline{V} , pro který $\overline{V} \cap \widehat{V} = \text{hrana}$ a konečně

V značí sousední voxel \overline{V} , pro který $\overline{V} \cap \widehat{V} = \text{vrchol}$.

V	E	V	E	F	E	V	E	V
E	F	E	F	T	F	E	F	E
V	E	V	E	F	E	V	E	V

Tabulka 2.2: Struktura třírozměrného konvolučního jádra.

Třírozměrné konvoluční jádro $K(i, j, k)$ se konstruuje z jednorozměrného $\widehat{K}(i)$ tak, že $K(i, j, k) = \widehat{K}(i)\widehat{K}(j)\widehat{K}(k)$. Následuje obrázek 2.3, kde jsou k porovnání vygenerovaná síť nezhlazená a zhlazená Gaussovým konvolučním jádrem.



Obrázek 2.3: Nezhlazená vygenerovaná síť, síť zhlazená pomocí Gaussova konvolučního jádra velikosti 9, $\sigma = 1$.

2.3.2 Zhlazení sítě pomocí zhlazovacího filtru

Cíle zjemňování sítí pro medicínské výpočty se dají rozdělit do dvou kategorií. První je čistě vizuální zlepšení sítě, redukce hranatých a schodovitých částí obrazu a druhou je, jak moc se získaná povrchová síť odlišuje od originálu. Do této kategorie patří zachování objemu a vzdáleností a také problematika ukončovacích kritérií. [5]

Pro získání hladší sítě se používají tři zhlazovací algoritmy: Laplaceův filtr, zpětný Laplaceův filtr a filtr lowpass. Jedná se o iterační metody, kde se pro vrchol P povrchové sítě s původní polohou p^0 jeho poloha v k -té iteraci p^k přepočítává z jeho současné polohy p^{k-1} a poloh jeho sousedů q_i^{k-1} , $i = 1, \dots, n$, kde n je počet sousedů vrcholu P .

Podle definice souseda se určuje řád metody. Pokud se berou pouze nejbližší sousedé vrcholu, tedy vrcholy, které tvoří s daným vrcholem hranu, mluvíme o metodě 1. řádu. Používá se i metoda 2. řádu, kde se za sousedy označí i nejbližší sousedé nejbližších sousedů daného vrcholu.

Laplaceův filtr

Laplaceův filtr nebo také relaxační filtr je nejjednodušší zhlazovací algoritmus. Jedná se o jednokrokovou metodu, která každý vrchol v každé iteraci posune do geometrického středu svých sousedů. Ukončovacím kritériem je získání požadované hladkosti.

Výhodou tohoto algoritmu je jednoduchost, nevýhodou však je, že objekt se při každé iteraci výrazně zmenšuje, proto se zavádí váhový faktor λ , který má regulovat vliv sousedních vrcholů. Pro daný vrchol P povrchové sítě s původní polohou p^0 se jeho nová poloha v k -té iteraci p^k přepočítává z jeho současné polohy p^{k-1} a poloh jeho sousedů q_i^{k-1} , $i = 1, \dots, n$, pomocí rovnice (2.3), kde λ je zadávaný parametr a n je počet sousedů vrcholu P .

$$p^k = (1 - \lambda)p^{k-1} + \frac{\lambda}{n} \sum_{i=1}^n q_i^{k-1} \quad (2.3)$$

Zpětný Laplaceův filtr

Zpětný Laplaceův filtr kombinuje Laplaceův filtr a zpětný chod, který má eliminovat zmenšování objemu tím, že vrcholy posouvá zpět směrem k původní poloze. Je to dvoukroková metoda, kde se pro daný vrchol P s původní polohou p^0 jeho nová poloha v k -té iteraci p^k přepočítává z jeho současné polohy p^{k-1} a poloh jeho sousedů q_i^{k-1} , $i = 1, \dots, n$, kde n je počet sousedů vrcholu P . Tento filtr byl představen v [29].

V prvním kroku se spočte nová poloha \hat{p}^k pomocí Laplaceova filtru se zvoleným parametrem λ jako v rovnici (2.4). Tato poloha je napočtena pro všechny vrcholy sítě.

$$\hat{p}^k = (1 - \lambda)p^{k-1} + \frac{\lambda}{n} \sum_{i=1}^n q_i^{k-1} \quad (2.4)$$

Ve druhém kroku se nejprve spočtou hodnoty b_p^k a $b_{q_i}^k$ jako v (2.5), kde pro $\lambda = 1$ by b_p^k bylo rovno posunutí vrcholu v prvním kroku.

$$\begin{aligned} b_p^k &= \hat{p}^k - [(1 - \lambda)p^0 + \lambda p^{k-1}] \\ b_{q_i}^k &= \hat{q}_i^k - [(1 - \lambda)q_i^0 + \lambda q_i^{k-1}] \end{aligned} \quad (2.5)$$

Výsledná poloha v k -tém kroku \hat{p}^k se nakonec spočte

$$\begin{aligned} p^k &= p^{k-1} - d^k, \\ d^k &= (1 - \alpha)b_p^k + \frac{\alpha}{n} \sum_{i=1}^n b_{q_i}^k. \end{aligned} \quad (2.6)$$

Filtr lowpass

V lowpass filtru nebo také λ/μ filtru se střídají dva Laplaceovy operátory (2.3), jeden s kladným a druhý se záporným váhovým faktorem λ . První Laplaceův operátor se použije s libovolným λ , druhý s $\lambda = \mu$, kde μ je nastaveno na hodnotu o něco menší než je $-\lambda$. Opět se tedy střídají vyhlazovací a zpětný krok, aby se zabránilo zmenšování objemu. Algoritmus využívá další parametr μ , který je pevně nastaven na $\mu = -1.02 \lambda$, jak je doporučováno v [5].

Srovnání podle [5]

Vhodné vyhlazení protáhlých objektů typu větvičí se cévy není možné s jednokrokovým zhlazovacím algoritmem, tedy není vhodný Laplaceův filtr. Podle [5] a [21] nemá smysl používat metody 2. řádu, protože příliš zmenšují objem, hlavně malých a oddělených částí obrazu. Doporučují tedy zpětný Laplaceův filtr nebo filtr lowpass, kde lowpass lépe zachovává objem. Také výsledné hodnoty mají lepší vztah k původnímu originálnímu objektu. Nicméně i s tímto filtrem se malé objekty zmenšují a velké mají tendenci se zvětšovat. Nejlépe dopadly váhové faktory $\lambda = 0,5$ a počet iterací v rozmezí 10 - 20.

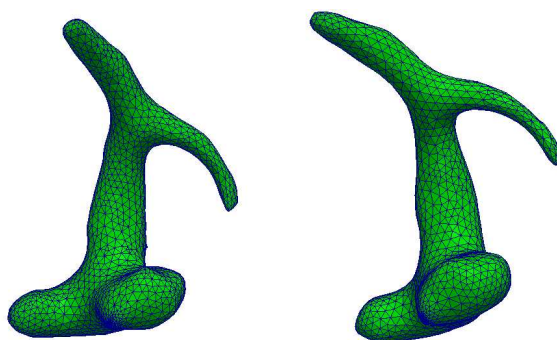
Pro zpětný Laplaceův filtr doporučují hodnoty parametrů $\alpha = 0$ a $\lambda = 0,2$.

Pro vizuální srovnání těchto filtrů jsme opět použili program iso2mesh [10]. Obrázek 2.6 ukazuje výsledky použití všech tří operátorů vždy bez použití zhlazení diskretního konvolučního jádra i s jeho použitím a tabulka 2.3 ukazuje použité

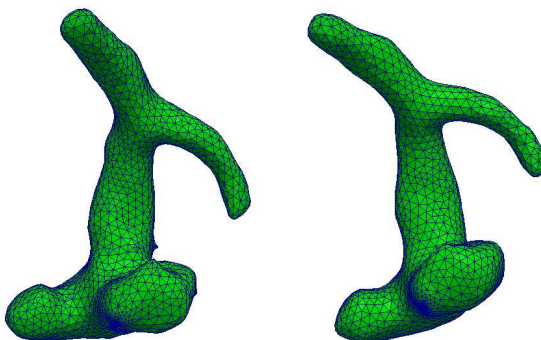
parametry λ a α . Na obrázku je mimo jiné vidět i to, že Laplaceův filtr bez zpětného kroku nezachovává objem. Opticky nejlépe dopadl zpětný Laplaceův filtr s rozmazáním dat pomocí konvolučního Gaussova jádra.

filtr	počet iterací	λ	α
Laplaceův	10	0.5	-
zpětný Laplaceův	10	0.2	0.0
lowpass	10	0.5	-

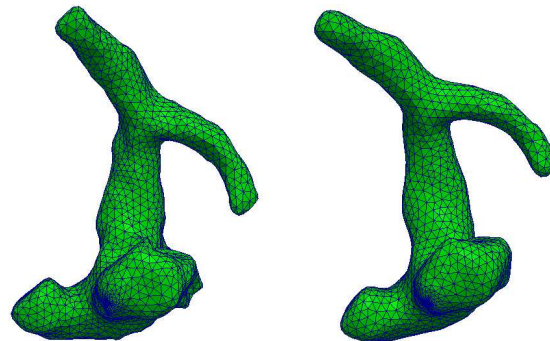
Tabulka 2.3: Parametry použité pro zhlazovací filtry.



Obrázek 2.4: Laplaceův filtr bez použití zhlazení Gaussovým konvolučním jádem a s jeho použitím.



Obrázek 2.5: Zpětný Laplaceův filtr bez použití zhlazení Gaussovým konvolučním jádem a s jeho použitím.



Obrázek 2.6: Filtr lowpass bez použití zhlazení Gaussovým konvolučním jádem a s jeho použitím.

2.4 Úprava sítě

V předchozích sekcích jsme ukázali, jak získat kvalitní hladkou objemovou síť pro výpočty proudění. Zbývá vytvoření vstupních a výstupních rovin pro zadávání okrajových podmínek. Toto se obvykle zajišťuje už při segmentaci vhodným natočením obrazu podle os souřadnic, kdy zarovnáním voxelů vznikne rovná stěna. Využívá se také interaktivního manuálního odstraňování jednotlivých voxelů.[9]

Nevýhodou tohoto přístupu je hlavně nemožnost libovolného ořezu. V této sekci představíme jednoduché oříznutí povrchové sítě a generování rovin vstupů a výstupů ze smyčky krajních vrcholů získaných tímto řezem.

Protne-li rovina řezu povrchový trojúhelník, pak pro každý z jeho vrcholů může nastat jedna ze dvou situací: vrchol se nachází buď v rovině řezu nebo uvnitř oblasti a má být ponechán nebo zůstal mimo oblast a má být odstraněn. Označení takových vrcholů pro další použití je uvedeno v tabulce 2.4.

poloha vrcholu	značení vrcholu	vrchol má být ponechán
na rovině řezu	V_0	ano
uvnitř oblasti	V_+	ano
mimo oblast	V_-	ne

Tabulka 2.4: Označení vrcholů trojúhelníka v závislosti na jejich poloze vůči rovině řezu.

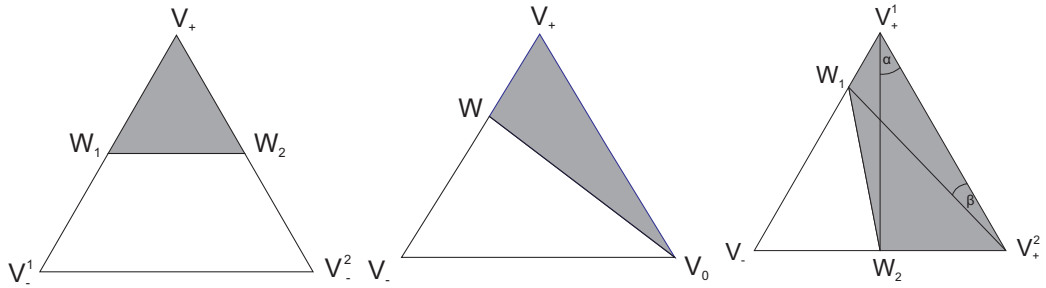
Pro povrchové trojúhelníky tímto vznikly tři možnosti:

- 1, **trojúhelník neobsahuje vrchol** V_+ a má být celý odstraněn
- 2, **trojúhelník neobsahuje vrchol** V_- a má zůstat nezměněn
3. **trojúhelník obsahuje oba tyto vrcholy** a má být ořezán, což je případ trojúhelníků $V_-V_0V_+$, $V_-V_-V_+$ a $V_-V_+V_+$.

První typ trojúhelníka je při řezu odstraněn, druhý ponechán beze změny. V třetím případě případě z původního trojúhelníka vznikají nové trojúhelníky. Ty však mohou být moc malé nebo mohou obsahují příliš malý úhel. Protože však jde o malý počet takových trojúhelníků, je možné upravit konce řezaného povrchu, aby neobsahovaly tyto nevyhovující trojúhelníky. Je třeba poznamenat, že předpokládáme, že původní řezaná síť je kvalitní v tom smyslu, že úhly povrchových trojúhelníků jsou blízké 60° .

Pro poslední typ trojúhelníka získáme vrcholy, které vznikly protnutím roviny řezu a jeho hrany, resp.hran. Ty označíme jako W , resp W_1 a W_2 . Možnosti řezu trojúhelníka jsou ukázány na obrázku 2.7.

Pokud vznikl čtyřúhelník $V_{1+}V_{2+}W_1W_2$, rozdělíme jej na dva trojúhelníky tak, aby vznikly ty větší úhly.



Obrázek 2.7: Řez trojúhelníka.

Takto vznikly dva typy trojúhelníků podle polohy jejich vrcholů. Pokud vrchol leží na hranici řezu, označíme ho 0, pokud leží uvnitř sítě, označíme ho 1. První typ trojúhelníka je trojúhelník 001, který má dva vrcholy na hranici řezu a jeden uvnitř sítě, a druhý typ je 011, který má jeden vrchol na hranici řezu a dva uvnitř sítě. Ukažme si tedy pro oba typy, jaké trojúhelníky mohou vznikat, které jsou nevhovující a jak je odstranit.

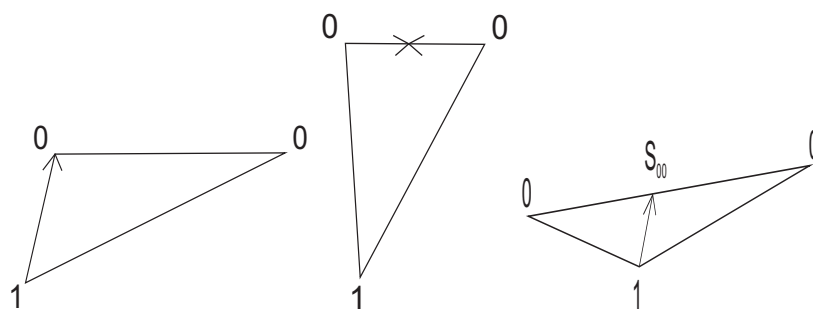
Typ 001

Jeden úhel u vrcholu 0 označíme α , druhý β , úhel u vrcholu 1 označíme γ . Pokud žádný z těchto trojúhelníků není příliš malý, je trojúhelník ponechán, v opačném případě je odstraněn. Úhel je příliš malý, pokud je menší než zadávaný minimální tolerovaný úhel α_{\min} .

Trojúhelník je odstraňován vždy tak, že jsou jeden nebo dva jeho vrcholy posunuty do jiné polohy. Všechny možnosti, kdy má být trojúhelník typu 001 odstraněn včetně toho, který vrchol má být kam posunut, jsou uvedeny v tabulce 2.5, kde S_{00} značí střed hrany tvořené dvěma vrcholy 0 a označení $A \rightarrow B$ vyjadřuje, že vrchol A má být přesunut do pozice vrcholu B. Pro větší názornost je naznačení přesunu uvedených vrcholů do nových pozic zobrazeno na obrázku 2.8.

$\alpha < \alpha_{\min}$	$\beta < \alpha_{\min}$	$\gamma < \alpha_{\min}$	přesun vrcholů
libovolné	libovolné	ano	$0 \rightarrow S_{00}, 0 \rightarrow S_{00}$
ano	ano	ne	$1 \rightarrow S_{00}$
ano	ne	ne	1 u úhlu $\beta \rightarrow S_{00}$
ne	ano	ne	1 u úhlu $\alpha \rightarrow S_{00}$

Tabulka 2.5: Odstraňování nevhovujících trojúhelníků typu 001 podle velikostí jejich vnitřních úhlů.



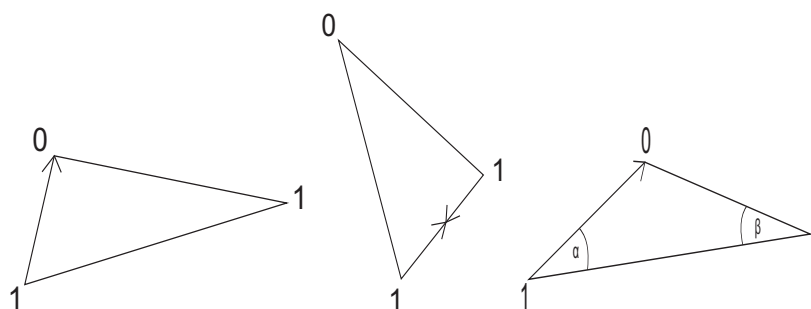
Obrázek 2.8: Přesuny vrcholů v trojúhelnících typu 001 podle tabulky 2.5.

Typ 011

Jeden úhel u vrcholu 1 označíme α , druhý β , úhel u vrcholu 0 označíme γ . Všechny možnosti, kdy má být trojúhelník typu 011 odstraněn včetně toho, který vrchol má být kam posunut, jsou uvedeny v tabulce 2.4, kde S_{11} značí střed hrany tvořené dvěma vrcholy 1 a označení $A \rightarrow B$ vyjadřuje, že vrchol A má být přesunut do pozice vrcholu B. Pro větší názornost je naznačení přesunu uvedených vrcholů do nových pozic zobrazeno na obrázku 2.9.

$\alpha < \alpha_{\min}$	$\beta < \alpha_{\min}$	$\gamma < \alpha_{\min}$	přesun vrcholů
libovolné	libovolné	ano	$1 \rightarrow S_{11}, 1 \rightarrow S_{11}$
ano	ano	ne	$1 \rightarrow 0$ u úhlu $\max(\alpha, \beta)$
ano	ne	ne	1 u úhlu $\beta \rightarrow 0$
ne	ano	ne	1 u úhlu $\alpha \rightarrow 0$

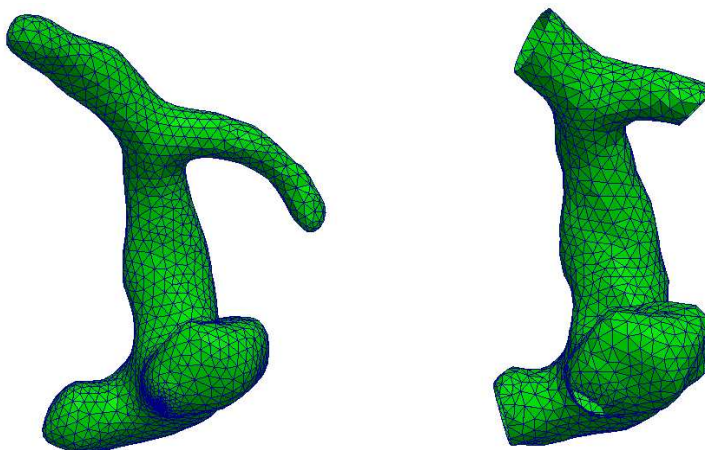
Tabulka 2.6: Odstraňování nevyhovujících trojúhelníků typu 011 podle velikostí jejich vnitřních úhlů.



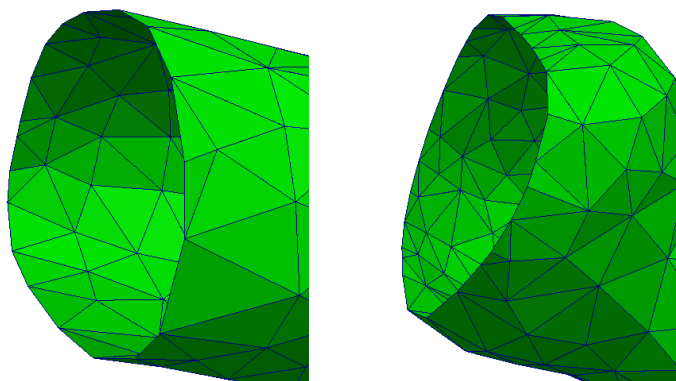
Obrázek 2.9: Přesuny vrcholů v trojúhelnících typu 011 podle tabulky 2.4

Následuje zhlazení sítě zpětným Laplaceovým filtrem, který je popsán v sekci 2.3.2. Vrcholy na vstupu jsou označeny jako nepohyblivé, aby byla zachována rovina řezu.

Následují obrázky 2.10 a 2.11, kde je ukázána povrchová síť zhlazená zpětným Laplaceovým filtrem a Gaussovým konvolučním jádrem před a po řezu. Jsou také ukázány povrchové trojúhelníky na okrajích řezu.



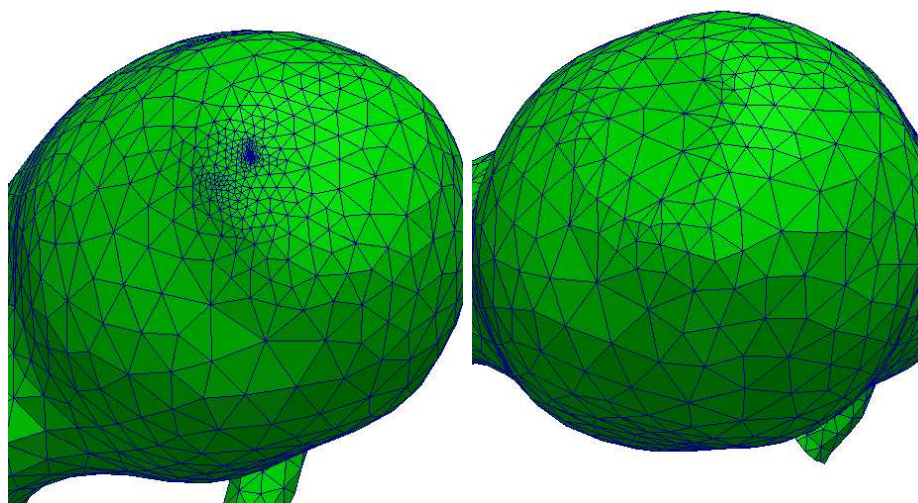
Obrázek 2.10: Síť před řezem a po řezu.



Obrázek 2.11: Detaily řezu.

Pomocí řezu lze také odstranit minimální trojúhelníky, které vznikají při teselaci, jak je vidět například na obrázku 2.12. Tyto trojúhelníky vznikají v místě počátečního generování sítě jako šum, tedy vznikají pouze na jednom místě sítě, kde nejsou žádoucí. Za minimální trojúhelník bereme trojúhelník, kde poměr jeho obsahu a průměrného obsahu je menší než nějaká určená hodnota, která je v této práci určena jako $\frac{1}{50}$. Tato hodnota je dostatečně velká k tomu, aby všechny takto definované minimální trojúhelníky ležely blízko sebe a šlo je odstranit jedním řezem. Takto upravené sítě již takové trojúhelníky neobsahují, jak je ukázáno v tabulce 3.2.

Nejprve tedy provedeme řez rovinou vybranou tak, aby obsahovala co nejvíce minimálních trojúhelníků. Pak je vygenerována vstupní rovina a zhlazena filtrem lowpass, kde za pohyblivé jsou označeny i vrcholy na hranici této roviny. Obrázek 2.12 zachycuje oblast s minimálními trojúhelníky před a po úpravě.



Obrázek 2.12: Minimální trojúhelníky vzniklé na síti a jejich odstranění pomocí řezu.

Popis úpravy sítě

Všechny následující úpravy jsou napsány v programu Matlab [19] s využitím knihoven programu iso2mesh [10]. Použit je také generátor dvojrozměrných sítí triangle [24]. Další funkce používané k řezu a úpravám sítě jsou uvedeny v příloze.

Vstupem pro úpravu je povrchová síť získaná tak, jak je popsáno v sekci 2.2. První částí úpravy je ořezání sítě, jak bylo popsáno výše v této sekci a vygenerování smyček vrcholů, které tvoří roviny vstupu a výstupu. Z těch je pak sestaven vstupní soubor pro program triangle[24], který ze zadaných hran mnohoúhelníka vygeneruje jeho triangulaci. Zadává se také například maximální tolerovaný obsah vzniklých trojúhelníků a minimální požadovaný úhel triangulace.

K ořezání sítě je třeba zadat roviny řezu. Tyto roviny se zadávají do pole *plains* pomocí normály roviny a bodu R, který na ní leží. Normála nemusí být orientovaná, zadává se i bod M, který leží na té části povrchové sítě, která má být odstraněna. Bod M je zadáván proto, aby nebyly odřezány i jiné části cévy, kterými řez prochází, a byla tedy odřezána pouze daná komponenta.

Každý řádek pole *plains* odpovídá jedné rovině řezu a má tvar

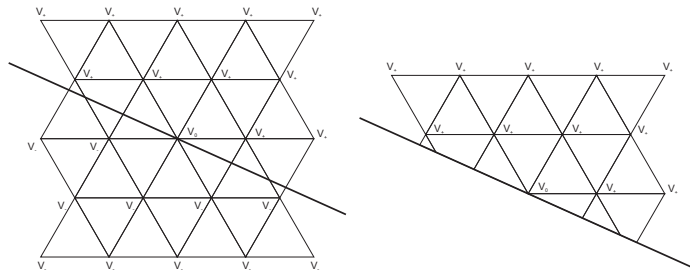
$$n_1 \ n_2 \ n_3 \ R_1 \ R_2 \ R_3 \ M_1 \ M_2 \ M_3 \ \text{číslo stěny.}$$

V tomto poli musí různým řezům odpovídat jiné číslo stěny. Není přípustné číslo stěny 1, které je určeno pro řezaný vstup.

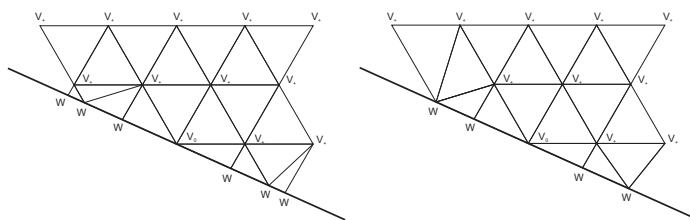
Dále je pro roviny řezu potřeba určit parametr *tol_zero* a cosinus minimálního tolerovaného úhlu *maxcos*. Minimální úhel je nejmenší přípustný úhel triangulace a číslo *tol_zero* je takové, že bod P leží v rovině $\rho : \vec{n}x + d = 0$, pokud $|\vec{n}P + d| < \text{tol_zero}$. Pro *tol_zero* není vhodné použít příliš malé číslo, v této práci je používáno 10^{-3} .

Pro tyto vstupní údaje program nejprve pro každou rovinu určí, které trojúhelníky má odstranit a které ořezat, jak bylo popsáno v této sekci výše. Pro trojúhelníky, které mají být ořezány, vygeneruje nově vznikající trojúhelníky. V těch zkoumá vzniklé minimální úhly a podle nich rozhoduje, zda a jak má být odstraněn daný trojúhelník, jak je ukázáno v tabulkách 2.5 a 2.4. Všechny takto vzniklé trojúhelníky tímto procesem projdou třikrát, což vylepší nežádoucí efekty na okolních neřezaných trojúhelnících.

Ilustrační příklad takového řezu je uveden na obrázcích 2.13 a 2.14.



Obrázek 2.13: Řez trojúhelníkové sítě.



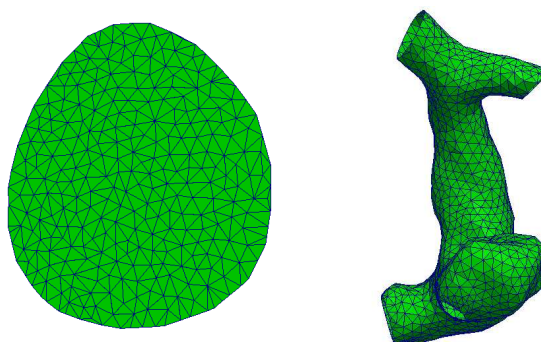
Obrázek 2.14: Vygenerování nových vrcholů a jejich posunutí.

Pokud je zadávaný minimální úhel triangulace α_{\min} dostatečně malý, například $\alpha_{\min} = 10^\circ$, nejsou ani sousední trojúhelníky příliš změněny. V této práci je použito $\alpha_{\min} = 20^\circ$, což nemusí zaručovat, že sousední trojúhelník nebyl posunut tak, aby vznikl menší úhel. Následné použití zhlazovacího zpětného Laplaceova filtru popsaného v sekci 2.3.2 posune vnitřní vrcholy tak, že výsledná síť je kvalitou srovnatelná s původní, jak lze vidět na tabulce 3.2. Vrcholy na vstupu jsou označeny jako nepohyblivé, aby byla zachována rovina řezu.

Druhou částí programu je použití programu triangle.[24] Zde stačí pouze zadat příkazy, které jsou v první části programu vypsané na obrazovku. Program vygeneruje pole krajních hran vstupu a převede obecnou rovinu vstupu na rovinu $z = 0$. Souřadnice libovolného krajního vrcholu A jsou převedeny do počátku $[0\ 0\ 0]$ a souřadnice jiných dvou krajních vrcholů vstupu B a C, které jsou s vrcholem A kolineární, jsou převedeny na $[|AB|\ 0\ 0]$ a $[x_C\ y_C\ 0]$, kde $x_C = \frac{b^2 - a^2 + c^2}{2c}$ a $y_C = \sqrt{b^2 - x^2}$.

Pro následné „přilepení“ vzniklých stěn je třeba si zapamatovat souřadnici vrcholu A, matici přechodu zpět k třídídimenzionálním souřadnicím a reálné indexy krajních vrcholů vstupu, které byly přečíslovány tak, aby šly vzestupně od jedné.

Příklad výstupu programu triangle a řez sítě před přidáním vygenerovaných rovin vstupu je uveden na obrázku 2.15.

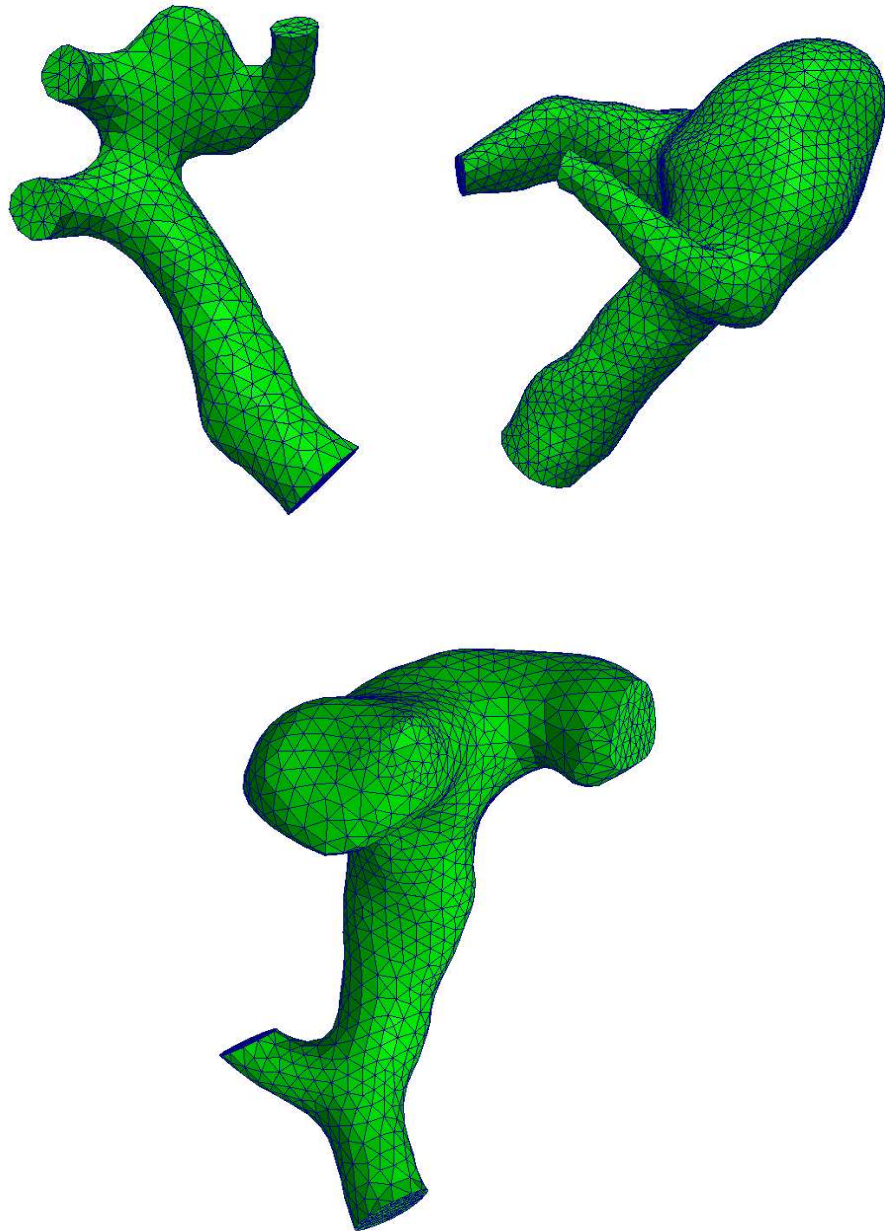


Obrázek 2.15: Příklad výstupu programu triangle a řez sítě před přidáním vygenerovaných vstupních rovin.

Všechny dosud použité úpravy jsou úpravy na povrchové síti, zbývá tedy vygenerovat objemovou síť. To bylo popsáno v sekci 2.2.

Při generování objemové sítě se nezachová očíslování trojúhelníků podle toho, na které stěně leží. Funkce zajišťující očíslování daných rovin a orientaci trojúhelníků jsou součástí adresáře Sítě, vzniklého pro potřeby této práce. Seznam funkcí adresáře Sítě je uveden v příloze. Při použití této funkce se zadává pole podobné poli *plains*, kde již můžeme použít libovolné očíslování stěn.

Výsledné síť po ořezání, přidání vstupních rovin, vygenerování objemových elementů a očíslování stěn jsou uvedeny na obrázku 2.16.



Obrázek 2.16: Výsledné síť použité k výpočtu.

3. Numerické výsledky

V této kapitole ukážeme spočtená rychlosti a tlaková pole na vytvořených sítích včetně dopočteného normálového a smykového napětí. Uvedeno bude také srovnání použitého akademického softwaru[2] a volně šiřitelného programu FEniCS pro řešení parciálních diferenciálních rovnic.[18]

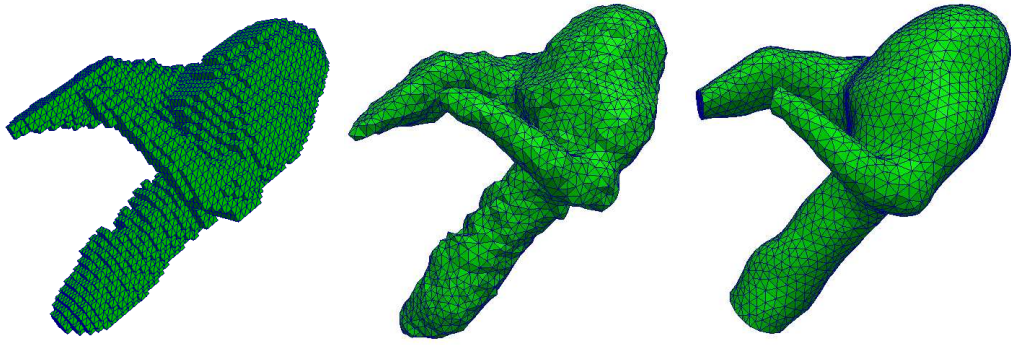
Nejprve však pro troje použítá data ukážeme získané sítě a provedeme jejich srovnání podle minimálního úhlu v povrchových trojúhelnících, obsahu minimálního povrchového trojúhelníka a objemu minimálního čtyřstěnu. K vizualizaci výsledků byly použity programy Netgen[12] a ParaView[8].

3.1 Srovnání vytvořených sítí

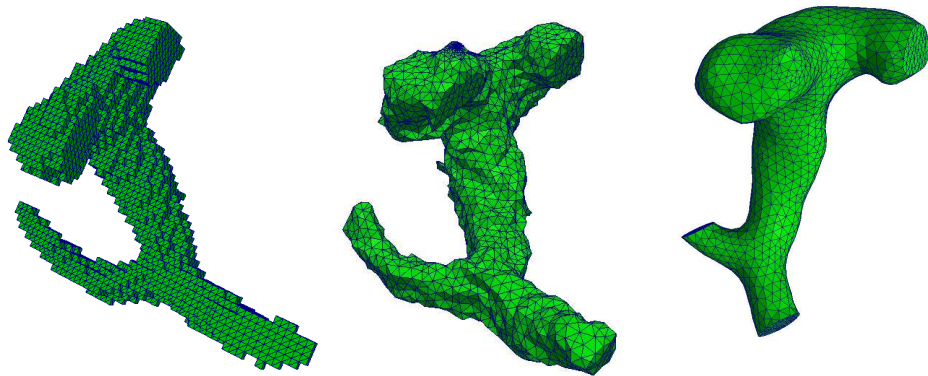
Na obrázcích 3.1, 3.2 a 3.3 jsou pro troje použítá data, která označme po řadě DATA1, DATA2 a DATA3 ukázány voxelové reprezentace segmentace získané z CT scanu, nezhlazené a neořezané sítě získané teselací pomocí isopovrchů a konečně výsledné sítě zhlazené pomocí diskrétního konvolučního jádra a zpětného laplaciánu použité pro výpočty. Na výsledných sítích již byly odstraněny minimální trojúhelníky vzniklé teselací, jak je popsáno v sekci 2.4.



Obrázek 3.1: DATA1: Segmentace získaná z CT scanu s 2737 voxely, vygenerovaná nezhlazená povrchová síť s 1679 vrcholy a 3354 trojúhelníky a zhlazená ořezaná objemová síť s 1204 vrcholy, 4022 čtyřstěny a 8864 trojúhelníky.



Obrázek 3.2: DATA2: Segmentace získaná z CT scanu s 7690 voxely a vygenerovaná nezhlazená povrchová síť s 2454 vrcholy a 4904 trojúhelníky a zhlazená ořezaná objemová síť s 3482 vrcholy, 15186 čtyřstěny a 32214 trojúhelníky.



Obrázek 3.3: DATA3: Segmentace získaná z CT scanu s 5350 voxely a vygenerovaná nezhlazená povrchová síť s 2232 vrcholy a 4458 trojúhelníky a zhlazená ořezaná objemová síť s 2514 vrcholy, 10140 čtyřstěny a 21784 trojúhelníky.

Pro povrchovou síť pro DATA3 vzniklou teselací isopovrchů jsou různé zhlazovací filtry a použití diskrétního konvolučního jádra srovnány podle kvality vzniklé sítě. Parametry určující kvalitu sítě jsou zde minimální úhel v triangulaci a minimální obsah trojúhelníka.

Tabulka 3.1 ukazuje pro DATA3 minimální úhel v povrchové síti a jeho procentuální poměr k průměrnému minimálnímu úhlu v povrchových trojúhelnících. Dále ukazuje minimální obsah povrchového trojúhelníku a jeho procentuální poměr k průměrnému obsahu povrchového trojúhelníka.

Porovnávají jsou sítě pro data nezhlazená nebo zhlazená zpětným laplaciánem (LHC) nebo filtrem lowpass (LOW), jak pro zhlazená konvolučním jádrem (GAUSS), tak pro nezhlazená.

zhlazovací filtr	zhlazení diskr.konv.	úhel [rad]		obsah [mm ²]	
		min.	poměr	min.	poměr
ne	GAUSS	0.53	62.4%	1.55e-04	0.02%
ne	ne	0.52	62.1%	2.26e-04	0.04%
LHC	GAUSS	0.38	43.6%	2.24e-04	0.04%
LHC	ne	0.31	35.7%	3.47e-04	0.06%
LOW	GAUSS	0.37	41.9%	2.08e-04	0.03%
LOW	ne	0.21	24.5%	3.58e-04	0.06%

Tabulka 3.1: Srovnání povrchové sítě pro různá zhlazení.

Předchozí tabulka 3.1 ukazuje, že bez ohledu na použité zhlazení síť vzniklá teselací obsahuje velké množství malých trojúhelníků. Tyto trojúhelníky vznikají v místě počátečního generování sítě jako šum. Vyskytují se však na jednom místě, takže se jich lze zbavit i pomocí řezu, jak je uvedeno v sekci 2.4.

Za malý označíme takový trojúhelník, kde poměr jeho obsahu a průměrného obsahu je menší než nějaká určená hodnota, která je v této práci určena jako $\frac{1}{50}$, což odpovídá 2%. Tato hodnota je dostatečně velká k tomu, aby všechny takto definované malé trojúhelníky ležely blízko sebe a šlo je odstranit jedním řezem. Takto upravené sítě již takové trojúhelníky neobsahují, jak je ukázáno v tabulce 3.2.

Tabulka 3.2 ukazuje opět velikost minimálního obsahu trojúhelníka, dále průměrný obsah a jejich poměr v procentech. To je ukázáno na třech použitých datech před a po ořezání povrchové sítě včetně její opravy. Pro tyto povrchové sítě byly následně vygenerovány sítě objemové. Tabulka 3.2 pro tyto sítě ukazuje také objem minimálního a průměrného čtyřstěnu a jejich poměr v procentech.

data	řez	obsah[mm ²]			objem[mm ³]		
		min.	prům.	poměr	min.	prům.	poměr
DATA1	před	9.5e-05	0.59	0.016%	3.8e-06	1.9	0.0002%
DATA1	po	0.071	0.64	11%	0.069	2.4	2.8%
DATA2	před	0.00031	0.64	0.049%	1.5e-05	2.5	0.0006%
DATA2	po	0.015	0.66	2.2%	0.02	2.6	0.77%
DATA3	před	0.00022	0.63	0.036%	1.3e-05	2.4	0.0006%
DATA3	po	0.044	0.67	6.5%	0.039	2.7	1.5%

Tabulka 3.2: Parametry povrchové a objemové sítě před a po řezu.

3.2 Srovnání konečných prvků

Nejprve ukážeme srovnání použitých konečných prvků podle konvergence a časové náročnosti na nestacionární neperiodický problém daný rovnicemi (1.1), kde byla použita časová závislost rychlosti na vstupu $v_t = 1000t$ s časovým krokem 0.01. Prostorová závislost rychlosti na vstupu je dána rovnicí (1.5). Použito bylo 16 procesorů.

Celkový výpočetní čas (CPU) a maximální čas, kdy byla dosažena konvergence jsou uvedeny v tabulce 3.4, parametry použitých sítí jsou uvedeny v tabulce 3.3.

data	počet vrcholů	počet čtyřstěnů
DATA1	2331	8777
DATA2	4349	19583
DATA3	3689	15658

Tabulka 3.3: Parametry sítí pro výpočet neperiodického proudění.

data	konečné prvky	počet zkonvergovaných kroků	maximální dosažená rychlost [mm/s]	CPU [s]
DATA1	TaHo	60	600	3843
	CrRa	60	600	27120
	MINI	70	700	3512
DATA2	TaHo	59	590	19462
	CrRa	62	620	72113
	MINI	59	590	9550
DATA3	TaHo	65	650	14054
	CrRa	42	420	55350
	MINI	67	670	7144

Tabulka 3.4: Maximální dosažená rychlost a celkový čas výpočtu pro neperiodické proudění.

Pro periodická proudění daná rovnicemi (1.1) s prostorovou závislostí rychlosti na vstupu danou rovnicí (1.5) a periodickou časovou závislostí rychlosti na vstupu danou vztahem (1.3) pro $v_{min} = 300\text{mm/s}$ a $v_{max} = 600\text{mm/s}$ byl z důvodu časové náročnosti použit MINI element. Periodické proudění bylo počítáno na 8 procesorech. Celkový čas výpočtu pro jednotlivá data s časovým krokem 0.01 a koncovým časem 2.0 je uveden v tabulce 3.5. Vedle celkového výpočetního času je uvedena i časová náročnost spočtení jacobíanu (NEW) a vyřešení lineárního algebraického problému (LIN). Je uvedeno srovnání na ořezaných sítích před a po odstranění minimálních trojúhelníků popsanych v sekci 2.4.

data	řez	CPU [s]	LIN [s]	NEW [s]	počet řádků počítané matice	počet elementů sítě
DATA1	před	1380	658	695	23384	5639
DATA1	po	802	364	411	16888	4022
DATA2	před	7726	5537	2134	59384	15092
DATA2	po	8016	5324	2602	59488	15186
DATA3	před	4309	2768	1494	45760	11480
DATA3	po	4191	2181	1891	40480	10140

Tabulka 3.5: Časová náročnost periodického proudění pro MINI element.

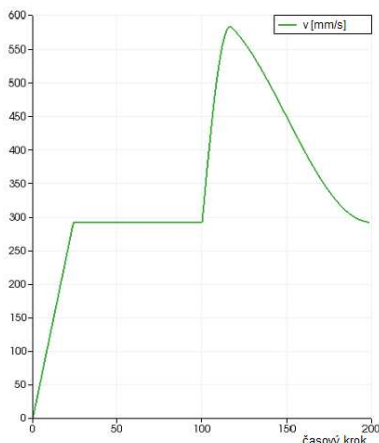
Pro stejné rovnice (1.1) a stejné okrajové podmínky (1.5), (1.3) s parametry $v_{min} = 200\text{mm/s}$ a $v_{max} = 400\text{mm/s}$ bylo použito 8 procesorů. Velikost časového kroku je 0.01, koncový čas 2.0. V tabulce 3.6 je ukázán celkový čas výpočtu, včetně velikosti sestavených matic. Vedle celkového výpočetního času je uvedena i časová náročnost spočtení jacobíanu a vyřešení lineárního algebraického problému.

data	počet čtyřstěňů	konečné prvky	počet řádků poč.matic	CPU [s]	LIN [s]	NEW [s]
DATA1	4022	MINI	16888	726	320	387
		TaHo	22952	2570	1478	1061
		CrRa	76496	14653	10575	3470
DATA2	15186	MINI	59488	6435	4362	1946
		TaHo	75456	22613	18028	4498
		CrRa	274920	158493	144430	13862
DATA3	10140	MINI	40480	2652	1656	973
		TaHo	52528	10641	7439	3109
		CrRa	186352	49826	41744	7562

Tabulka 3.6: Časová náročnost periodického proudění pro různé prvky.

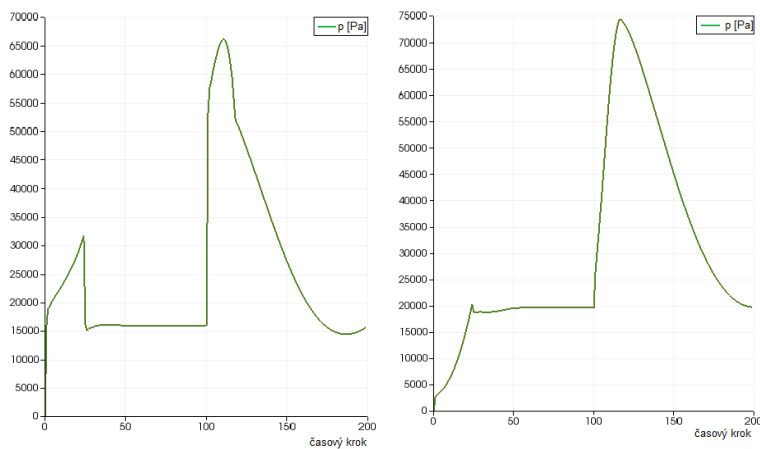
3.3 Spočtená rychlostní a tlaková pole

V této sekci budou ukázány výpočty rychlosti a tlaku na získaných sítích. Použity jsou rovnice (1.1), rychlost na vstupu je zadána vztahy (1.5), (1.3), kde $v_{min} = 300\text{mm/s}$ a $v_{max} = 600\text{mm/s}$. Bylo použito 8 procesorů a prvek MINI element popsáný v sekci 1.5. Velikost časového kroku je 0.01, koncový čas 2.0. Na obrázku 3.4 je zobrazena zadávaná časová závislost rychlosti na vstupu.

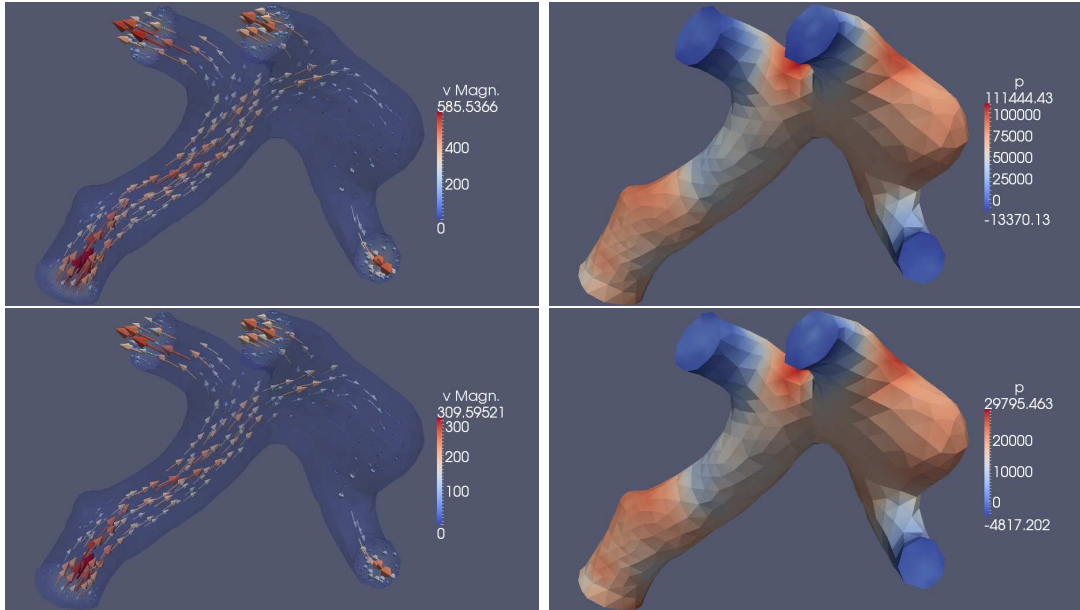


Obrázek 3.4: Časová závislost rychlosti na vstupu.

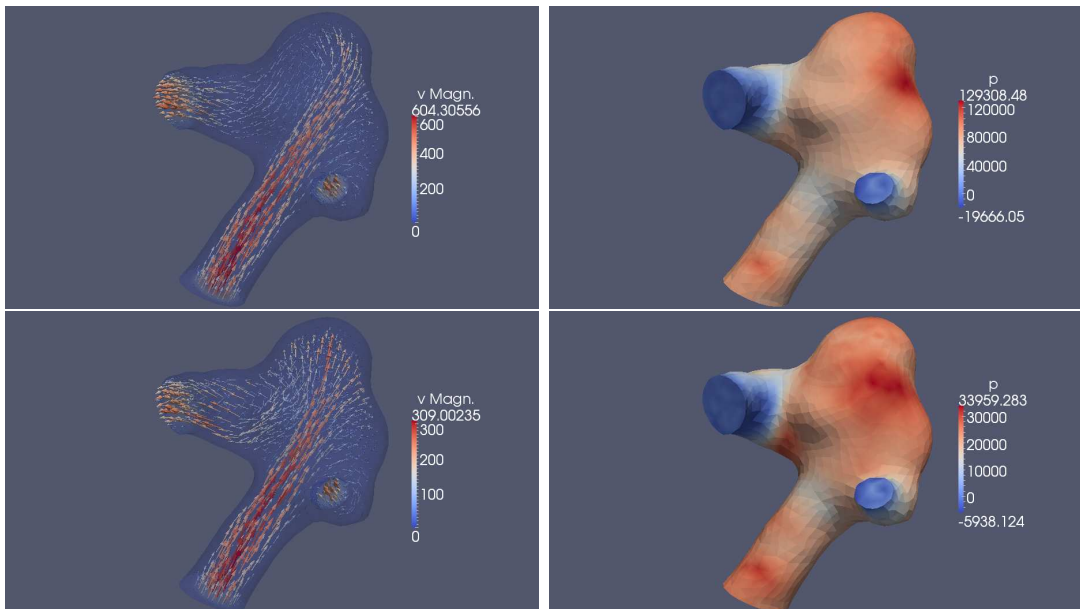
Budou ukázána rozložení rychlosti a tlaku ve dvou časových krocích. Nejprve v časovém kroku 117 odpovídajícímu maximální zadávané rychlosti na vstupu 600mm/s a pak v časovém kroku 199 odpovídajícímu konci proudění s rychlostí 300mm/s. Rychlost a tlak jsou ukázány po řadě DATA1, DATA2 a DATA3 na obrázcích 3.6, 3.7 a 3.8. Pro DATA1 je ukázána i časová závislost spočteného tlaku na vstupu v bodě těžiště vstupního mnohoúhelníka a na povrchu aneurysma na obrázku 3.5.



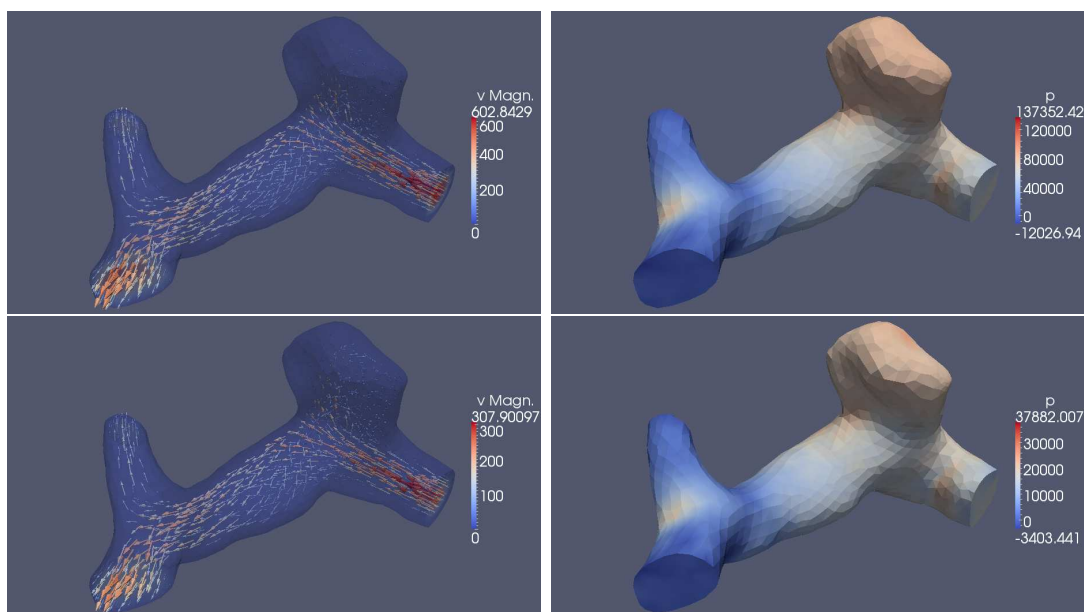
Obrázek 3.5: DATA1: Tlak na vstupu a v aneurysmě.



Obrázek 3.6: DATA1: Rozložení rychlosti v [mm/s] a tlaku p [Pa] v časových krocích 117 a 199.



Obrázek 3.7: DATA2: Rozložení rychlosti v [mm/s] a tlaku p [Pa] v časových krocích 117 a 199.



Obrázek 3.8: DATA3: Rozložení rychlosti v [mm/s] a tlaku p [Pa] v časových krocích 117 a 199.

Výsledky pro výpočet rychlosti a tlaku byly porovnány s programem FEniCS [18]. Pro používaný akademický software Fstrin [2] jsou použity konečné prvky TaHoa MINI, pro program FEniCS je použit pouze prvek TaHo.

Použity byly rovnice (1.1), kde rychlost na vstupu je specifikována vztahy (1.5), (1.3) s $v_{min} = 200\text{mm/s}$ a $v_{max} = 400\text{mm/s}$. Velikost časového kroku je 0.01, koncový čas 2.0. Zadávaná časová rychlost na vstupu je vidět na obrázku 3.4.

Numerické výsledky vyšly stejné. V tabulkách 3.8 a 3.9 je uveden celkový čas výpočtu na 1 a 8 procesorech. Velikosti počítaných matic a počty objemových elementů sítě jsou uvedeny v tabulce 3.7.

data	konečné prvky	počet řádků počítané matice	počet elementů sítě
DATA1	TaHo	22952	4022
DATA1	MINI	16888	4022
DATA2	TaHo	75456	15186
DATA2	MINI	59488	15186
DATA3	TaHo	52528	10140
DATA3	MINI	40480	10140

Tabulka 3.7: Parametry výpočtu pro porovnání programů Fstrin a FEniCS.

data	program	konečné prvky	CPU [s]
DATA1	Fstrin	TaHo	4268
	Fstrin	MINI	4012
	FEniCS	TaHo	2326
DATA2	Fstrin	TaHo	122045
	Fstrin	MINI	29351
	FEniCS	TaHo	45343
DATA3	Fstrin	TaHo	51989
	Fstrin	MINI	14805
	FEniCS	TaHo	14816

Tabulka 3.8: Porovnání časové náročnosti programů Fstrin a FEniCS pro výpočet na 1 procesoru.

data	program	konečné prvky	CPU [s]
DATA1	Fstrin	TaHo	2570
	Fstrin	MINI	726
	FEniCS	TaHo	778
DATA2	Fstrin	TaHo	22613
	Fstrin	MINI	6435
	FEniCS	TaHo	8400
DATA3	Fstrin	TaHo	10641
	Fstrin	MINI	2652
	FEniCS	TaHo	3420

Tabulka 3.9: Porovnání časové náročnosti programů Fstrin a FEniCS pro paralelní výpočet na 8 procesorech.

3.4 Spočtená normálová a smyková napětí

Jak bylo uvedeno v sekci 1.1, cévní stěny reagují na abnormální smykové napětí na stěnách. Z toho důvodu je dopočtení normálových a smykových napětí důležitou součástí výpočtů proudění. Program tedy během svého výpočtu nepočítá pouze tlak p a rychlost v , ale i gradient rychlosti \mathbf{L} . Z něj je pak dopočten Cauchyho tenzor napětí daný vztahem (3.1) a vektor napětí daný vztahem (3.2), kde

$$\nu = 2,8 \cdot 10^{-3} g s^{-1} mm^{-1}$$

je konstantní dynamická viskozita a n je vnější jednotková normála v daném vrcholu povrchové sítě. Ta je dopočítána jako průměr vnějších jednotkových normál ke stěnám daným povrchovými trojúhelníky, které obsahují daný vrchol.

$$\mathbf{T} = -p\mathbf{I} + \nu(\mathbf{L} + \mathbf{L}^T) \quad (3.1)$$

$$s = -\mathbf{T}n \quad (3.2)$$

Normálové napětí s_n je skalární veličina daná vztahem (3.3) a smykové napětí s_t je skalární veličina definovaná jako velikost vektoru, jak je uvedeno ve vztahu (3.4).

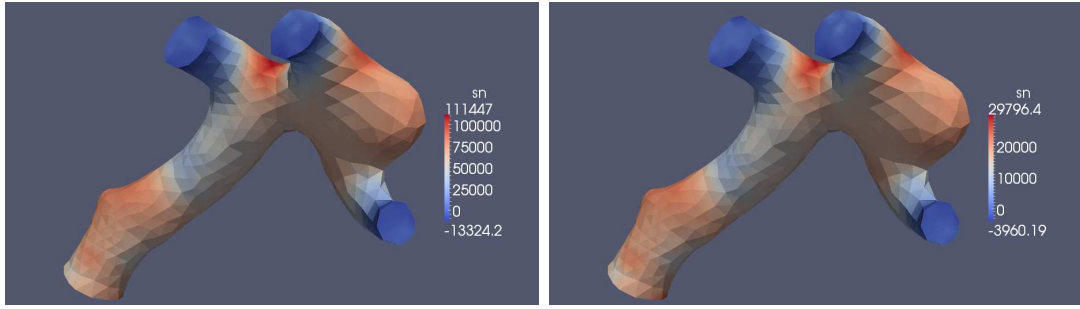
$$s_n = s \cdot n \quad (3.3)$$

$$s_t = |s - s_n n| \quad (3.4)$$

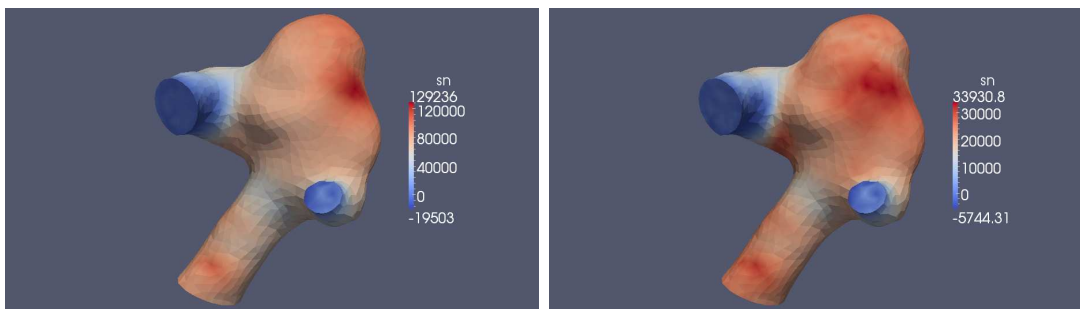
Použity jsou rovnice (1.1), rychlost na vstupu je zadána vztahy (1.5), (1.3) s $v_{min} = 300\text{mm/s}$ a $v_{max} = 600\text{mm/s}$. Bylo použito 8 procesorů a prvek MINI element popsany v sekci 1.5. Velikost časového kroku je 0.01, koncový čas 2.0. Časovou závislost rychlosti zadávané na vstupu je vidět na obrázku 3.4.

Na obrázcích 3.9, 3.10 a 3.11 jsou ukázána napočtená normálová napětí s_n ve dvou časových krocích. Nejprve v časovém kroku 117 odpovídajícímu maximální zadávané rychlosti na vstupu 600mm/s a pak v časovém kroku 199 odpovídajícímu konci proudění s rychlostí 300mm/s. s_n je počítáno bodově pro každý vrchol na povrchu sítě.

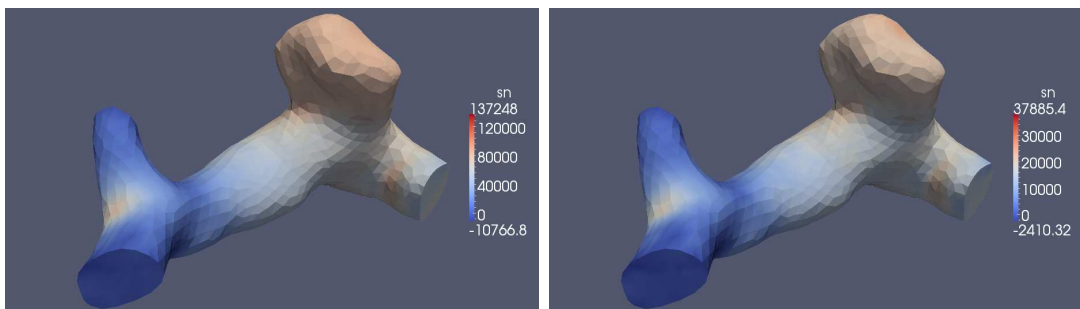
Pro stejný výpočet jsou zobrazena i napočtená smyková napětí s_t ve třech časových krocích. První časový krok 117 odpovídá maximální zadávané rychlosti na vstupu 600mm/s, druhý časový krok 158 maximální zadávané rychlosti na vstupu 450mm/s a třetí časový krok 199 odpovídá konci proudění s maximální zadávanou rychlostí 300mm/s. s_t je počítáno bodově pro každý vrchol na povrchu sítě a ukázáno na obrázcích 3.12, 3.13 a 3.14. Řešení jsou zobrazena pro větší názornost i v logaritmické stupnici.



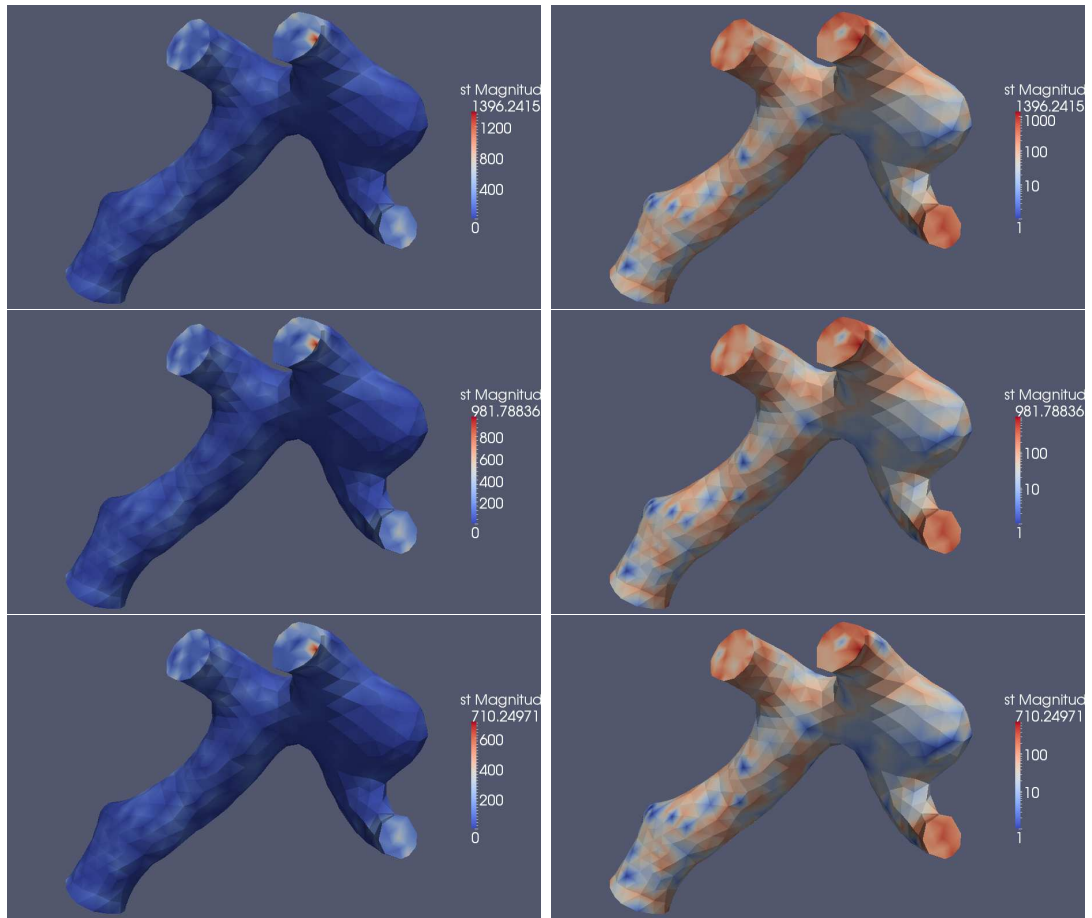
Obrázek 3.9: DATA1: Rozložení normálového napětí s_n [Pa] v časových krocích 117 a 199.



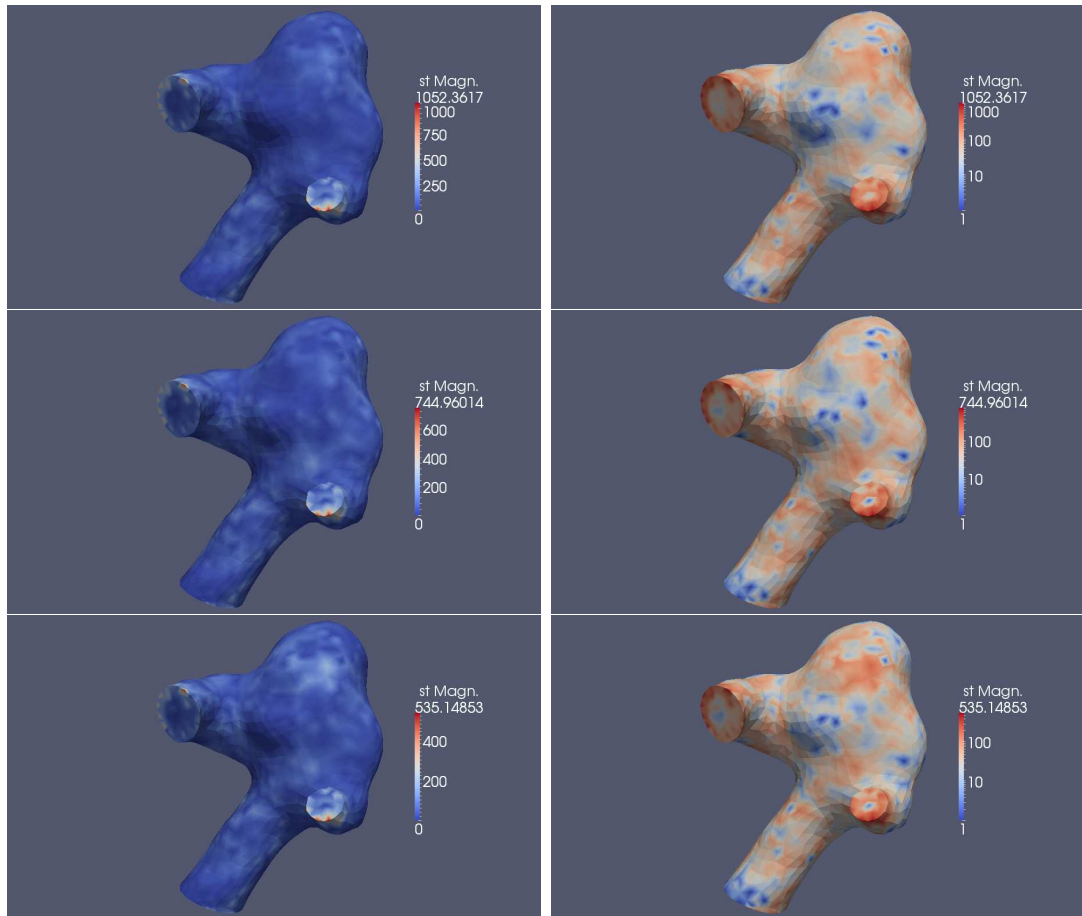
Obrázek 3.10: DATA2: Rozložení normálového napětí s_n [Pa] v časových krocích 117 a 199.



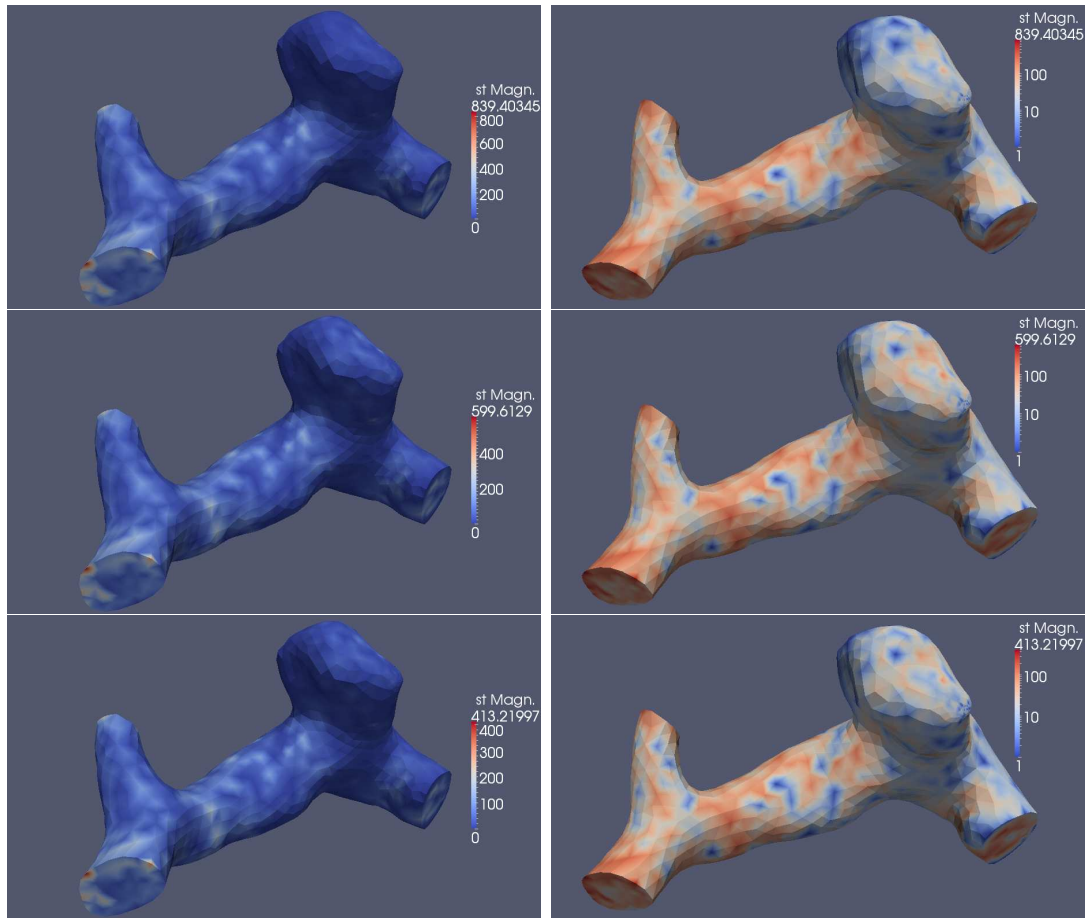
Obrázek 3.11: DATA3: Rozložení normálového napětí s_n [Pa] v časových krocích 117 a 199.



Obrázek 3.12: DATA1: Rozložení smykového napětí s_t [Pa] v časových krocích 117, 158 a 199 nejprve v klasické, pak v logaritmické stupnici.



Obrázek 3.13: DATA2: Rozložení smykového napětí s_t [Pa] v časových krocích 117, 158 a 199 nejprve v klasické, pak v logaritmické stupnici.



Obrázek 3.14: DATA3: Rozložení smykového napětí s_t [Pa] v časových krocích 117, 158 a 199 nejprve v klasické, pak v logaritmické stupnici.

Závěr

Cílem této práce bylo popsat postup spočtení proudového pole v reálných geometriích cév postižených aneurysmatem.

V teoretické části této práce byly nejprve ukázány použité rovnice a diskutovány použité okrajové podmínky. Byl také popsán postup úpravy rovnic od získání variační formulace přes časovou diskretizaci až po tvar nelineárního algebraického problému a konečné prvky použité pro prostorovou diskretizaci včetně vhodných aproximačních prostorů.

Hlavním výsledkem této části byl popis získání, zhlazení a přípravy výpočetní sítě z hrubých zobrazovacích informací získaných pomocí výpočetní tomografie. To bylo realizováno ve druhé kapitole, kde jsme nejprve popsali, jak je získána voxelová segmentace požadované oblasti na základě intenzity rentgenového paprsku procházejícího tkání. Následně byla diskutována potřeba zhlazení sítě pomocí diskretního konvolučního jádra a pomocí zhlazovacích filtrů, kde jsme vybrali nejvhodnější postup pro zadaná data. Konečně bylo provedeno ořezání okrajů povrchové sítě pomocí vytvořeného programu a přidány roviny vstupů vygenerované programem pro tvorbu dvourozměrné sítě. Následně byla vygenerována objemová síť.

V části pro numerické výsledky byly zobrazeny takto získané sítě a bylo ukázáno, že jsou dobré kvality pro další výpočty. Podařilo se také spustit výpočet rychlostního a tlakového pole včetně dopočtení normálových a smykových napětí. Pro implementaci byl použit software na výpočty klasickou metodou konečných prvků vyvinutý na MFF UK. Okrajové podmínky bylo třeba přizpůsobit reálným informacím o daném proudění, které jsme měli k dispozici.

Je otevřenou otázkou a prací do budoucna, jakým způsobem zadávat okrajové podmínky. Reálně máme k dispozici pouze několik naměřených bodových hodnot rychlosti uvnitř cévy. Všechny naměřené hodnoty a použité segmentace byly poskytnuty Neurochirurgickou klinikou Masarykovy nemocnice v Ústí nad Labem.

V třetí části práce jsou také uvedeny časové náročnosti použitých konečných prvků, kde numericky se všechny zdají být dostatečně vhodné pro výpočet. Pro tyto prvky bylo nejprve spuštěno neperiodické nestacionární proudění, kde se také zkoumal počet zkonvergovaných časových kroků při konstantně zvyšované vstupní rychlosti. Pro různé prvky je pak spuštěn i periodický výpočet. U něj je sledován nejen celkový čas výpočtu, ale také časová náročnost spočtení jakobiánu a lineárního algebraického problému. Vedle použitého akademického softwaru Fstrin[2] byl výpočet pro Taylorův-Hoodův prvek na získaných sítích spuštěn i v programu FEniCS[18].

Na výsledky této práce lze také navázat při zapojení interakce elastických stěn pro popis proudění, kde je potřeba použití vícepovrchových sítí, získávaných obdobně jako jednopovrchové sítě uvedené v této práci.

Jsou zde samozřejmě také otázky urychlení výpočtu, kde jsou možnosti zlepšení paralelního výpočtu. Je zde také možnost použití jiných řešičů pro lineární algebraický problém. V případě interakce dosahuje velikost řešené matice řádu 10^6 a více a její podmíněnost má velký vliv na chování lineárních řešičů.

Dalším zajímavým problémem je použití sofistikovanějších modelů pro popis proudění krve a jejich aplikace na získané sítě.

Seznam použité literatury

- [1] CGAL, Computational Geometry Algorithms Library. <http://www.cgal.org>.
- [2] Jaroslav Hron a Martin Mádlík. Fluid-structure interaction with applications in biomechanics. *Nonlinear Analysis: Real World Applications*, 8(5):1431 – 1458, 2007.
- [3] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.
- [4] D. N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21(4):337–344 (1985), 1984.
- [5] Ragnar Bade, Jens Haase, and Bernhard Preim. Comparison of fundamental mesh smoothing algorithms for medical surface models. In *In Simulation und Visualisierung 2006*, pages 289–304, 2006.
- [6] Satish Balay, Jed Brown, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Web page, 2013. <http://www.mcs.anl.gov/petsc>, naposledy navštíveno 1.8.2013.
- [7] J.-D. Boissonnat, R. Chaine, P. Frey, G. Malandain, S. Salmon, E. Saltel, and M. Thiriet. From arteriographies to computational flow in saccular aneurisms: the INRIA experience. *MED. IMAGE ANAL.*, 9:133–143, 2005.
- [8] Steven P. Callahan, Juliana Freire, Carlos E. Scheidegger, Cláudio T. Silva, Huy T. Vo, and Vistrails Inc. Towards Provenance-Enabling ParaView.
- [9] J.R. Cebal and R. Lohner. From Medical Images to CFD Meshes. In *Proc. 8th Int. Meshing Roundtable, South Lake Tahoe*, pages 321–331, 1999.
- [10] Qianqian Fang and David A. Boas. Tetrahedral mesh generation from volumetric binary and gray-scale images. In *Proceedings of the Sixth IEEE international conference on Symposium on Biomedical Imaging: From Nano to Macro*, ISBI'09, pages 1142–1145, Piscataway, NJ, USA, 2009. IEEE Press.
- [11] Jaroslav Haslinger. *Metoda konečných prvků pro řešení variačních rovnic a nerovnic*. Praha : Státní pedagogické nakladatelství, 1981.
- [12] Reinhold Huber, Reinhold Huber, Helmut A. Mayer, Helmut A. Mayer, Roland Schwaiger, and Roland Schwaiger. netGEN - A Parallel System Generating Problem-Adapted Topologies of Artificial Neural Networks by means of Genetic Algorithms, 1995.
- [13] Volker John, Gunar Matthies, and Joachim Rang. A comparison of time-discretization/linearization approaches for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 195(44–47):5995 – 6010, 2006.

- [14] Janko D. Jovanović and Miomir Lj. Jovanović. Finite element modeling of the vertebra with geometry and material properties retrieved from CT-scan data. *Facta Universitatis - series Mechanical Engineering*, 8(1):19 – 26, 2010.
- [15] Patrick M. Knupp. Remarks on Mesh Quality. <https://cfwebprod.sandia.gov/cfdocs/CCIM/docs/reno07paper.pdf>, naposledy navštíveno 1.8.2013.
- [16] O. Steinbach L. John, P. Pustějovská. On the influence of the wall shear stress vector form on hemodynamic indicators. <http://www.numerik.math.tugraz.at/berichte/Bericht0613.pdf>, naposledy navštíveno 1.8.2013, 2013.
- [17] J.R. Levick. *An Introduction to Cardiovascular Physiology*. A Hodder Arnold Publication. Hodder Arnold, 2010.
- [18] Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.
- [19] MATLAB. *version 7.8.0 (R2009a)*. The MathWorks Inc., Natick, Massachusetts, 2009.
- [20] Roberto Molteni. From CT Numbers to Hounsfield Units in Cone Beam Volumetric Imaging: the effect of artifacts. www.aaomr.org/resource/resmgr/2012_Uploads/AAOMR2011_medium.pdf, naposledy navštíveno 1.8.2013.
- [21] University of Magdeburg. Comparison of mesh smoothing algorithms and strategies for medical surface models. <http://www.wisg.cs.uni-magdeburg.de/cv/projects/LST/smoothing>, naposledy navštíveno 1.8.2013, 2006.
- [22] Makoto Ohta, Stephan G. Wetzel, Philippe Dantan, Caroline Bachelet, Karl O. Lovblad, Hasan Yilmaz, Patrice Flaud, and Daniel A. Rüfenacht. Rheological Changes After Stenting of a Cerebral Aneurysm: A Finite Element Modeling Approach. *CardioVascular Interventional Radiology*, 28(6):768, 2005.
- [23] Stéphanie Salmon, Marc Thiriet, and Jean-Frédéric Gerbeau. Medical image – based computational model of pulsatile flow in saccular aneurysms. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 37(4):663–679, 2003.
- [24] Jonathan Richard Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.
- [25] Hang Si and Klaus Gärtner. Meshing Piecewise Linear Complexes by Constrained Delaunay Tetrahedralizations. In *In Proceedings of the 14th International Meshing Roundtable*, pages 147–163. Springer, 2005.

- [26] Massachusetts Medical Society. Unruptured Intracranial Aneurysms — Risk of Rupture and Risks of Surgical Intervention. *New England Journal of Medicine*, 339(24):1725–1733, 1998. PMID: 9867550.
- [27] C.J. Solomon and T.P. Breckon. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. Wiley-Blackwell, 2010.
- [28] Y. Ventikos, E. C. Holland, T. J. Bowker, P. N. Watton, N. M. P. Kakalis, M. Megahed, F. Zhu, P. E. Summers, and J. V. Byrne. Computational modelling for cerebral aneurysms: risk evaluation and interventional planning. *British Journal of Radiology*, 82:62–71, 2009.
- [29] J. Vollmer, R. Mencl, and H. Müller. Improved laplacian smoothing of noisy surface meshes. In *Computer Graphics Forum*, pages 131–138, 1999.

Seznam obrázků

1.1	Skutečná a použitá časová závislost rychlosti na vstupu	3
1.2	Zadávání prostorové závislosti rychlosti na vstupu	4
1.3	Průřezy spočtených proudění u vstupu	5
1.4	Lokální stupně volnosti pro prvek TaHo.	11
1.5	Lokální stupně volnosti pro prvek CrRa.	11
1.6	Lokální stupně volnosti pro prvek MINI.	12
2.1	Rozdělení krychle voxelu na 6 čtyřstěnů a získaná voxelová síť	16
2.2	Nezhlazená vygenerovaná síť.	17
2.3	Síť zhlazená pomocí Gaussova konvolučního jádra velikosti 9, $\sigma = 1$	19
2.4	Použití Laplaceova filtru	21
2.5	Použití zpětného Laplaceova filtru	21
2.6	Použití filtru lowpass	21
2.7	Řez trojúhelníka.	23
2.8	Přesuny vrcholů v trojúhelnících typu 001	24
2.9	Přesuny vrcholů v trojúhelnících typu 011	24
2.10	Síť před řezem a po řezu.	25
2.11	Detaily řezu.	25
2.12	Minimální trojúhelníky vzniklé na síti a jejich odstranění	26
2.13	Řez trojúhelníkové sítě.	27
2.14	Vygenerování nových vrcholů a jejich posunutí.	27
2.15	Příklad výstupu programu triangle a řez sítě	28
2.16	Výsledné síť použité k výpočtu.	29
3.1	DATA1: Parametry sítí v průběhu jejich získávání	30
3.2	DATA2: Parametry sítí v průběhu jejich získávání	31
3.3	DATA3: Parametry sítí v průběhu jejich získávání	31
3.4	Časová závislost rychlosti na vstupu.	35
3.5	DATA1: Tlak na vstupu a v aneurysmě.	35
3.6	DATA1: Rozložení rychlosti a tlaku	36
3.7	DATA2: Rozložení rychlosti a tlaku	36
3.8	DATA3: Rozložení rychlosti a tlaku	37
3.9	DATA1: Rozložení normálového napětí	40
3.10	DATA2: Rozložení normálového napětí	40
3.11	DATA3: Rozložení normálového napětí	40
3.12	DATA1: Rozložení smykového napětí	41
3.13	DATA2: Rozložení smykového napětí	42
3.14	DATA3: Rozložení smykového napětí	43

Seznam tabulek

2.1	Příklad konvolučního jádra, kdy sousedé nemají žádný vliv.	18
2.2	Struktura třírozměrného konvolučního jádra.	18
2.3	Parametry použité pro zhlazovací filtry.	21
2.4	Označení vrcholů trojúhelníka	22
2.5	Odstraňování nevyhovujících trojúhelníků typu 001	23
2.6	Odstraňování nevyhovujících trojúhelníků typu 011	24
3.1	Srovnání povrchové sítě pro různá zhlazení.	32
3.2	Parametry povrchové a objemové sítě před a po řezu.	33
3.3	Parametry sítě pro výpočet neperiodického proudění.	33
3.4	Počet zkonvergovaných kroků pro neperiodické proudění	33
3.5	Časová náročnost periodického proudění pro MINI element.	34
3.6	Časová náročnost periodického proudění pro různé prvky.	34
3.7	Parametry výpočtu pro porovnání programů Fstrin a FEniCS.	37
3.8	Porovnání programů Fstrin a FEniCS na 1 procesoru	38
3.9	Porovnání programů Fstrin a FEniCS na 8 procesorech	38

Seznam použitých zkratek a matematických symbolů

\mathbf{R}^3	Euklidovský prostor
u	rychlostní pole
p	tlakové pole
ν	dynamická viskozita
\mathbf{T}	Cauchyho tenzor napětí
n	pole vnějších jednotkových normál
u_{in}	okrajová podmínka pro rychlost na vstupu
u_0	počáteční podmínka pro rychlost na vstupu
v_t	časová závislost rychlosti na vstupu
u_X	prostorová závislost rychlosti na vstupu
v_{max}	maximální hodnota zadávané rychlosti v periodickém proudění
v_{min}	hodnota rychlosti v proudění
Ω	omezená oblast v \mathbf{R}^3 s po částech polygonální hranicí
Γ_{in}	část hranice se vstupní rovinou
Γ_{out}	část hranice s výstupními rovinami
Γ_0	část hranice, na které je předepisována nulová rychlost
t	čas
T	koncový čas
\mathbf{I}	identický tenzor
\mathbf{L}	gradient rychlosti
V, Q	nekonečně dimenzionální prostory funkcí
\widehat{V}	nekonečně dimenzionální lineární prostor funkcí
V_h, Q_h	konečně dimenzionální aproximace prostorů V, Q
t^k	k-tý časový krok
Δt	velikost časového kroku
u^k	hodnota funkce u v k-tém časovém kroku
v, q	testovací funkce
TaHo	Taylorův-Hoodův konečný prvek
CrRa	Crouzeixův-Raviartův konečný prvek
MINI	MINI prvek

CT	computed tomography, výpočetní tomografie
HU	Haunsfield unit, jednotka používaná pro hodnoty CT čísel

LHC	zpětný Laplaceův zhlazovací filtr
LOW	zhlazovací filtr lowpass
GAUSS	Gaussovo zhlazovací konvoluční jádro
CPU	celkový výpočetní čas
LIN	časová náročnost lineárního algebraického problému
NEW	časová náročnost sestavení jakobiánu
s	vektor napětí
s_n	normálové napětí
s_t	smykové napětí

Seznam funkcí adresáře Sítě

$face = create_face(elem, size(node, 1), s, f_new)$

- ze zadané objemové sítě vygeneruje povrchovou síť
- node = pole vrcholů, každý řádek tvoří tři souřadnice jednoho vrcholu
- elem = pole čtyřstěňů, každý řádek tvoří číslo oblasti a indexy jeho čtyř vrcholů, čísluje se od 1, číslo oblasti je pro pevnou síť tvořenou jednou oblastí rovno jedné
- face = pole povrchových trojúhelníků, každý řádek tvoří číslo stěny a indexy jeho tří vrcholů
- s = počet povrchů vznikající sítě, může být 1 nebo 2
- pro s=1 f_new = číslo stěny pro všechny vznikající trojúhelníky
- pro s=2 f_new = číslo stěny, která je spojuje čtyřstěny s jiným číslem oblasti

$[node, elem, face] = create_voxel_mesh(aux)$

- vytvoří čtyřstěnovou síť pro zobrazení voxelů.
- node, elem, face jako v *create_face*
- aux = třírozměrné binární pole segmentace

$[M, A, node_real] = cut(input, output, plains, tol_zero)$

- řez sítě zadanými rovinami
- input = soubor s povrchovou sítí
- output_rez.mesh = soubor, kam se uloží ořezaná síť bez vstupních a výstupních rovin,
- output_rez_cislo.poly = název souboru pro program triangle
- plains, tol_zero jako v sekci 2.4
- M,A = matice přechodu a body pro posunutí pro návrat výstupních 2D rovin zpět do 3D
- node_real = reálné indexy krajních vrcholů výstupních rovin

$[node, elem, face] = loading(input)$

- načte síť ze souboru input ve formátu .fsgeo, .mesh nebo .xml
- node, elem, face jako v *create_face*

$saving(output, node, elem, face)$

- uloží síť do souboru output ve formátu .mesh nebo .xml
- node, elem, face jako v *create_face*

$$[node, face, index] = move(node, face, index)$$

- posouvá vrcholy L s indexy v levém sloupci matice indexů na pozice vrcholů P s indexy v pravém sloupci matice indexů, v node zanikají vrcholy L, ve face zanikají trojúhelníky s hranou LR, dojde k přečíslování indexů
- node, face jako v *create_face*

$$[elem, face] = orient(node, elem, face)$$

- node, elem, face jako v *create_face*
- přehodí pořadí vrcholů v poli čtyřstěnnů, aby byly zorientovány podle pravidla pravé ruky
- přehodí pořadí vrcholů v poli trojúhelníků, aby byly zorientovány podle jim příslušných čtyřstěnnů

$$[face] = reconstructionBC(node, face, BC_set, tol_zero)$$

- očísluje stěny podle zadaných rovin
- node, face jako v *create_face*
- BC_set je ve tvaru normála roviny, vrchol roviny, číslo vznikající stěny
- tol_zero popsán v sekci 2.4

$$[loop, start] = get_loop(edge)$$

- z matice hran $N \times 2$, které tvoří smyčku, například 1-2, 2-4, 4-5, 5-1 vytvoří smyčku 1-2-4-5
- může vzniknout více smyček, indexy jejich začátků jsou pak ve start
- netvoří-li hrany smyčku, vydá chybovou hlášku

$$loop = set_pro,xyz(node, face, f_new_input, tol_zero)$$

- vytvoří smyčku vrcholů, které tvoří hranici mezi číslem stěny f_new_input a stěnou 1
- loop jako v *get_loop*
- node, face jako v *create_face*
- tol_zero popsán v sekci 2.4

$$[outputA, outputB] = split(remA, remB, face, node)$$

- vytvoří dvě povrchové sítě - vnější A a vnitřní B z jedné dvoupovrchové, uloží je do outputA a outputB
- node, face jako v *create_face*
- remA, resp. remB = čísla stěn, které nepatří do A, resp. do B
- zajistí konzistenci sítě pro generování objemové sítě v programu Netgen

$$[node, elem, face] = join(inputA, inputB)$$

- vytvoří jednu dvoupovrchovou objemovou síť ze dvou objemových, které mají stejně očíslované vrcholy na styčné stěně
- node, elem, face jako v *create_face*