



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Numerická simulace problémů lineární elasticity v prostředí OpenFoam

Bakalářská práce

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 - Elektronické informační a řídicí systémy

Autor práce: **Vítězslav Novák**

Vedoucí práce: Ing. Petr Šidlof, PhD.

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum: 15.5.2014

Podpis

Abstrakt

Tato bakalářská práce řeší výpočet mechanického namáhání ocelového přípravku numerickou metodou konečných objemů s použitím opensource softwaru. Pro návrh namáhaného přípravku a generování výpočetní sítě je použit parametrický CAD Gmsh a samotný výpočet probíhá pomocí řešiče solidEquilibriumFoam CFD balíku OpenFoam. Výsledná pole napětí a deformací jsou vizualizována programem Paraview. Geometrie namáhaného přípravku je dostatečně jednoduchá, aby bylo možné spočítat přesné analytické řešení. Tyto výsledky jsou pak vzájemně porovnány s tenzometrickým měřením na reálném přípravku.

Klíčová slova: pružnost, metoda konečných objemů, Gmsh, Openfoam, solidEquilibriumFoam

Abstract:

This bachelor thesis solves computation of mechanical stress of iron beam, by numeric method of finite volumes, using opensource software. For design and generation of computational mesh network, parametric CAD Gmsh is used. The calculation itself is done by solidEquilibriumFoam solver of OpenFoam CFD software package. Resulting fields of tensions and deformations are visualized by Paraview. Geometry of the beam is simple enough to be possible to find exact analytical solution. Results are then mutually compared with real tensometric measuring.

Keywords: elasticity, finite volume method, Gmsh, Openfoam, solidEquilibriumFoam

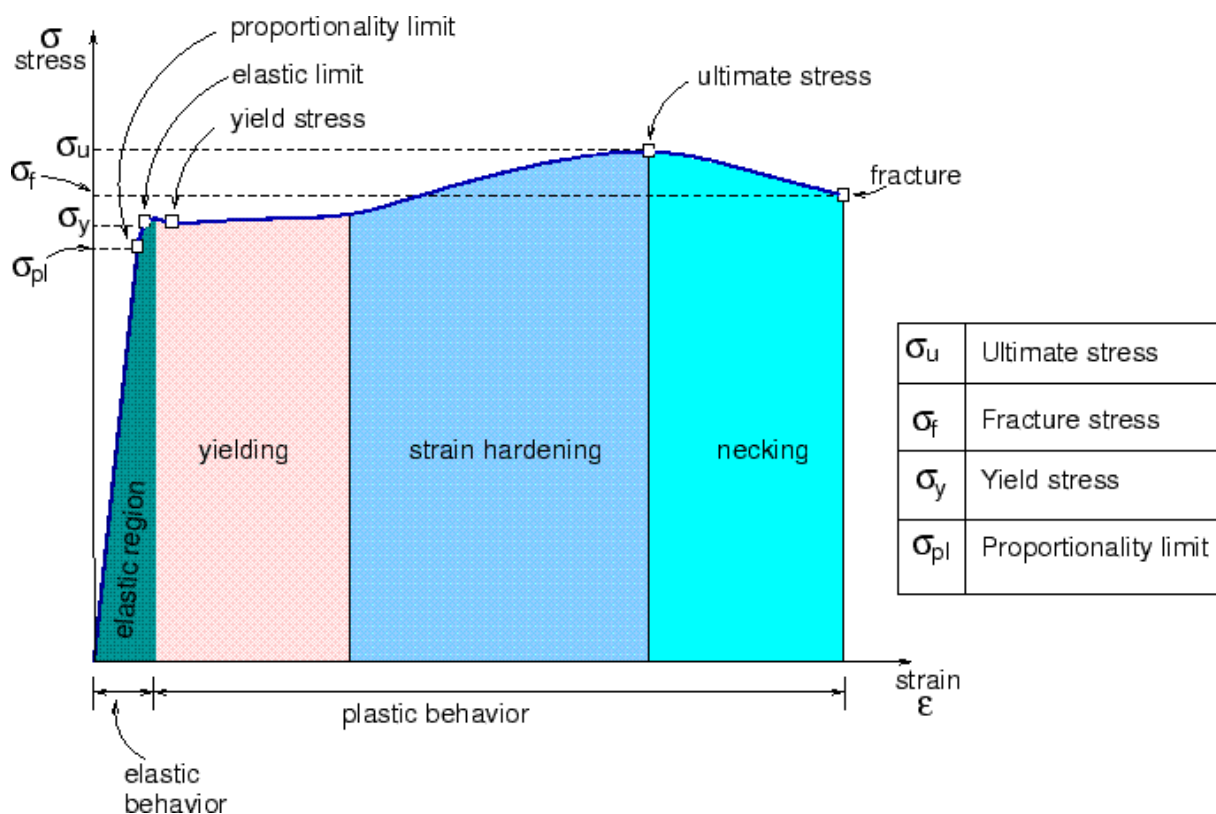
Obsah

1	Úvod.....	8
2	Základy technické pružnosti pevných látek.....	10
2.1	Osové namáhání tyče.....	10
2.2	Ohyb nosníku.....	11
2.2.1	Silové uvolnění.....	11
2.2.1	Vnitřní statické účinky.....	11
3	Numerické metody pro řešení úloh elasticity.....	16
3.1	Přehled metod.....	16
3.1.1	Metoda konečných prvků.....	16
3.1.2	Metoda konečných objemů.....	16
3.1.3	Metoda konečných diferencí.....	16
3.2	Síťování.....	16
4	Výpočet pole napětí a deformace při zatížení nosníku příčnou silou.....	18
4.1	Popis problému, měření.....	18
4.2	Analytické řešení.....	20
4.3	Numerické řešení.....	23
4.3.1	Generování výpočetní sítě pomocí Gmsh.....	23
4.3.2	Základní typy výpočetních sítí.....	27
4.3.3	Výpočet pomocí balíku OpenFoam.....	30
4.3.4	Řešení úlohy pomocí řešiče solidDisplacementFoam.....	30
4.3.5	Vizualizace výsledku výpočtu v Paraview.....	34
5	Porovnání výsledků a závěr.....	37
	Použité materiály.....	39
	Přílohy:.....	40

1 Úvod

V technické praxi se při tvorbě nejrůznějších konstrukcí a součástí, které jsou fyzicky namáhané dostáváme k otázkám, jak budou měnit vlivem tohoto namáhání svůj tvar a jaké v nich bude rozložení napětí, tedy otázkám mechanické pružnosti a zdali toto namáhání nebude mít destruktivní účinky na celistvost materiálu, ze kterého jsou vyrobeny, tedy otázkám mechanické pevnosti. Je jasné, že tyto vlastnosti materiálů mají zásadní vliv na funkčnost a bezpečnost daných konstrukcí a je tudíž nutné vědět předem, jak se dané konstrukce budou za určitých podmínek chovat.

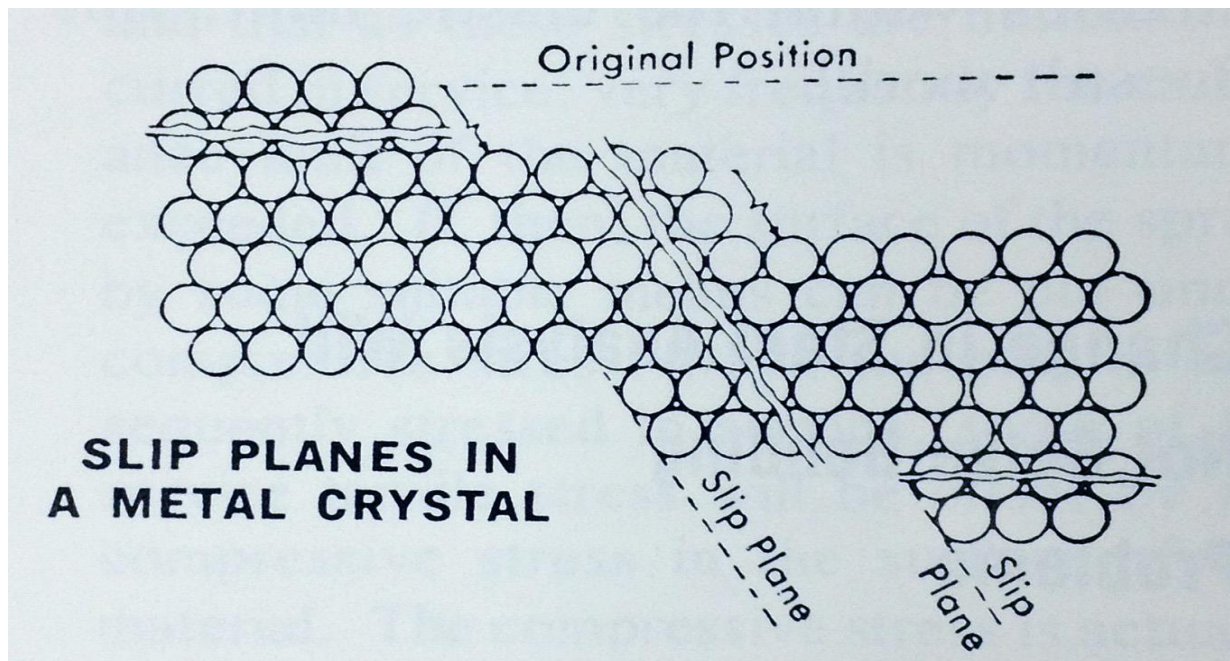
Náhled na chování materiálu, na který mechanicky působíme silou nám velmi dobře zprostředkuje tzv. pracovní diagram materiálu. V našem případě budeme uvádět ocel, jakožto jeden z hlavních konstrukčních materiálů obecně. Zároveň přípravek, na kterém budeme provádět měření v této práci, je také z ocele. Pracovní diagram je záznam silového působení trhačského stroje.



Obr. 1-Pracovní diagram,(převzato z [1])

V této práci se zabýváme silovým působením, které nepřekročí hranici σ_y , což je mezní napětí, při kterém se materiál chová ještě jako elastický. To znamená, že po ukončení silového působení se materiál vrátí do svého původního tvaru. Tato oblast se nazývá oblastí lineární elasticity nebo také oblastí platnosti Hookeova zákona. Materiál je v této oblasti ideálně pružný a závislost velikosti deformace na velikosti síly, která ji způsobuje je lineární. Po překročení této hranice se dostáváme do oblasti plasticity-tvarovatelnosti. Atomy, respektive celé vrstvy atomů změni své

původní polohy a dislokují se do nových rovnovážných poloh jinde, ve kterých pak zůstanou. Plastická deformace je pohyb těchto dislokací.



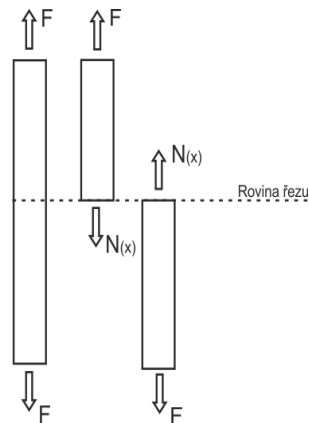
Obr. 2-Dislokace atomových vrstev - plastická deformace (převzato z [2])

Cílem této práce je popsat pole napětí a deformace příčně zatíženého nosníku. Takový výpočet lze analyticky provést pouze na dostatečně jednoduchých geometriích (což je i případ této práce), nicméně v technické praxi je tato podmínka splněna málokdy. S nástupem výpočetní techniky byly vyvinuty aproximativní numerické metody, které převádějí problém analytického řešení pomocí parciálních diferenciálních rovnic na problém řešení rovnic algebraických. Standardem v oblasti simulací problémů lineární elasticity pevných látek je metoda konečných prvků. V této práci se ale pokusíme posoudit vhodnost použití metody konečných objemů, která je běžně používaná zejména pro numerické výpočty proudění tekutin. Výpočet bude proveden pomocí řešiče `solidDisplacementFoam` open-source balíku `OpenFoam`, což je pokročilý software pro výpočty z oblasti dynamiky plynů a kapalin - (CFD - computational fluid dynamics).

2 Základy technické pružnosti pevných látek.

2.1 Osově namáhání tyče

Při popisu jednoosého namáhání prismatické (všude stejný průměr) tyče, nejprve vyjádříme vnitřní statické účinky.



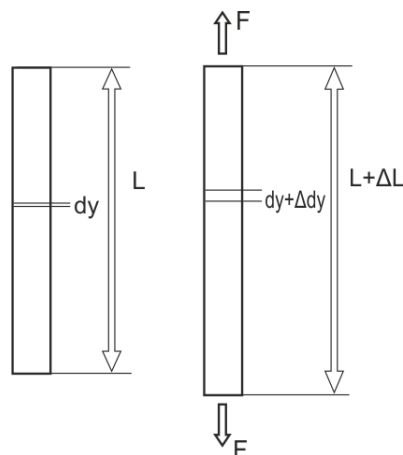
Obr. 3-Silové namáhání tyče

V místě myšleného řezu působí normálová síla $N_{(x)}$, která je kladná pro tah a záporná pro tlak.

Napětí v rovině řezu:

$$\sigma_{(x)} = \frac{N_{(x)}}{S} = \frac{N}{m^2} = Pa$$

Deformace ε ve směru y :



Obr. 4-Deformace tyče

$$\varepsilon = \frac{\Delta L}{L} \quad , \quad \varepsilon_y = \frac{\Delta d_y}{d_y}$$

Deformace ε je poměrová bezrozměrná veličina, ale byl zaveden strain, respektive μ strain. Vztah mezi napětím σ a deformací ε řeší Hookeův zákon: $\sigma = \varepsilon * E$, přičemž E je Youngův modul pružnosti, což je materiálová konstanta. $E_{ocel} = 210 \text{ GPa}$. Čili pro představu, hranice elasticity je pro běžnou konstrukční ocel (viz.pracovní diagram na obr.1) $\sigma_y = 210 \text{ MPa}$, pak

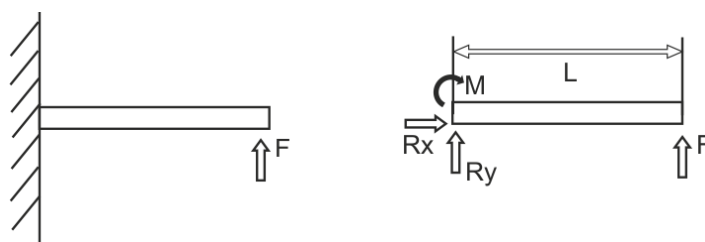
$$\varepsilon = \frac{\sigma_y}{E_{ocel}} = \frac{210 \text{ MPa}}{210 \text{ GPa}} = 0.001,$$

což znamená, že metrovou ocelovou tyč lze elasticky natáhnout maximálně o 1mm tak, aby se pružně vrátila do původního tvaru.

2.2 Ohyb nosníku

Nosníkem rozumíme těleso, které má délku podstatně větší, než příčné rozměry, a které je zatíženo příčnými silami. V našem případě je nosník z jedné strany pevně vetknutý. Pro analytické zjištění pole napětí a deformace a pro určení průhybové křivky $w(x)$ nejprve nosník silově uvolníme a následně popíšeme jaké vyvolají zjištěné síly a momenty v daném nosníku vnitřní statické účinky.

2.2.1 Silové uvolnění

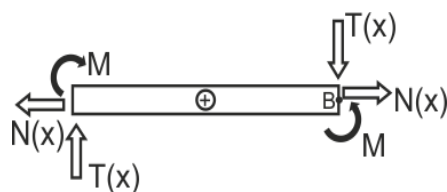


Obr. 5-Silové uvolnění vetknutého nosníku

$$R_y = -F$$

$$M = F \cdot l$$

2.2.1 Vnitřní statické účinky



Obr. 6-Vnitřní statické účinky

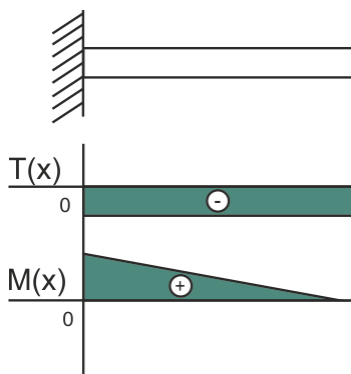
$$T(x) = -F$$

$$M_0(x) = F(L - x) \quad (1)$$

$T(x)$.. Posouvací síla (Shear force)

$M_0(x)$.. Ohybový moment

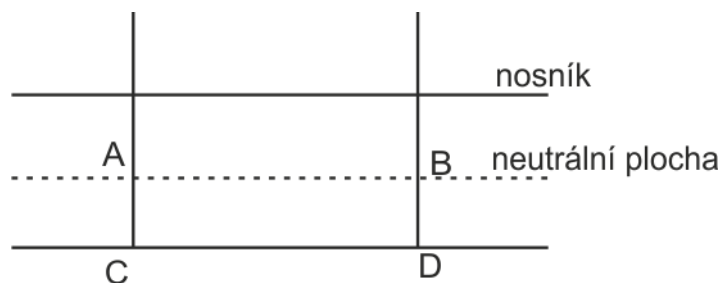
Síla působící na nosník jej tvaruje do konvexního tvaru, čili moment je z konvence kladný.



Obr. 7-Rozložení posouvací síly a ohybového momentu

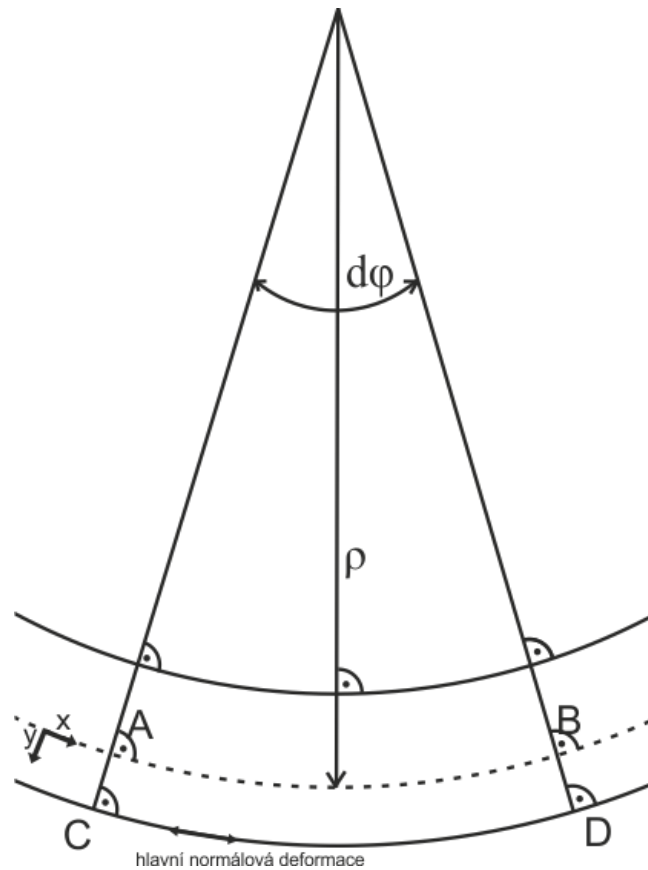
Největší napětí bude v místě s největším momentem $M_0(x)$, tedy v místě vetknutí nosníku.

Při detailním pohledu na místo ohýbání nosníku si můžeme všimnout, že díky ohybu se nám hmota na jedné straně roztahuje a na protější straně smršťuje. Pokud budeme uvažovat předchozí případ ohnutí do konvexního tvaru, pak horní plocha nosníku se bude smršťovat a spodní roztahovat. Pokud procházíme takto ohnutým materiálem nosníku od horní strany k dolní, tak plynule procházíme z oblasti smrštěné do oblasti roztažené s tím, že někde musí existovat místo přechodu, kde materiál není ani smrštěný ani roztažený – neutrální plocha. Díky této vlastnosti je to ideální místo pro umístění osy x souřadného systému pro popis vnitřních statických účinků. Tato plocha prochází těžištěm a její tvar je závislý na celkovém tvaru nosníku.



Obr. 8-Nosník před ohybem

Nosník s vyznačenou neutrální plochou před ohybem. Vzdálenost AB a CD je stejná.



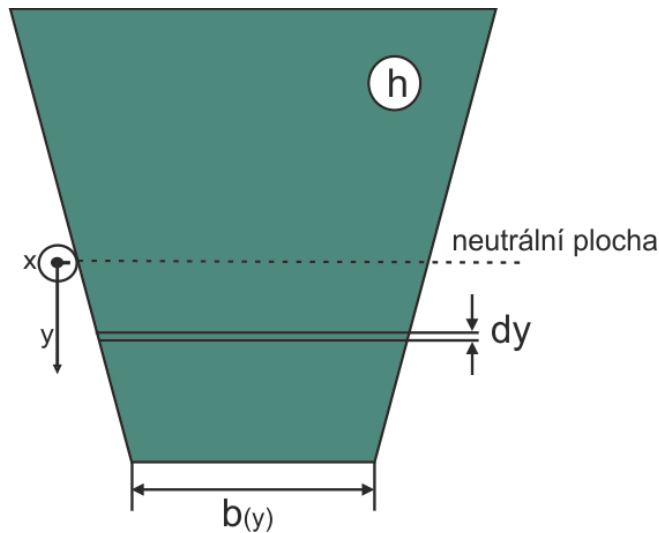
Obr. 9-Ohnutý nosník

Při ohnutém nosníku se vzdálenost bodů CD díky prohnutí zvětší, což vyvolá napětí $\varepsilon_x(y)$. Toto napětí odpovídá právě tomuto zvětšení vzdálenosti bodů CD oproti vzdálenosti bodů AB v neutrální vrstvě. A od B je vzdálen o $d\varphi$ při rádiu ρ . C od D je vzdálen o $d\varphi$ při rádiu $\rho + y$.

$$\varepsilon_x(y) = \frac{|CD| - |AB|}{|AB|} = \frac{(\rho + y)d\varphi - \rho d\varphi}{\rho d\varphi} = \frac{y}{\rho}$$

$$\sigma_x(y) = E \cdot \varepsilon_x(y) = \frac{Ey}{\rho}$$

Pro nalezení momentové rovnice nejprve vyjádříme normálovou sílu $N(x)$, což je vnitřní statický účinek přes celou plochu h průřezu kolmo na plochu papíru.

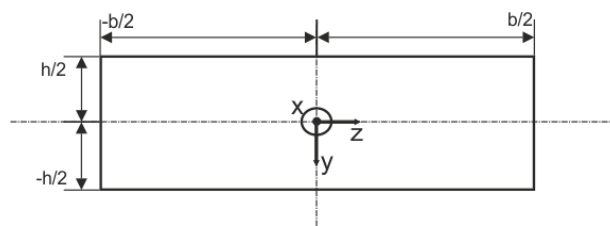


Obr. 10-Profil nosníku s vyznačenou neutrální plochou

$$N(x) = \int_h \sigma_x(y) \cdot b(y) dy = \frac{E}{\rho} \int_h y \cdot b(y) dy = \frac{E}{\rho} \int_S y dS$$

$$M(x) = \int_h \sigma_x(y) \cdot b(y) \cdot y dy = \frac{E}{\rho} \int_h y^2 \cdot b(y) dy = \frac{E}{\rho} \int_S y^2 dS$$

Člen $J(y) = \int_S y^2 dS$ nazveme kvadratický moment setrvačnosti průřezu, který nám definuje vlastnosti konkrétní geometrie průřezu nosníku. V případě nosníku použitého v této práci se jedná o obdélníkový průřez. Pro jeho výpočet je nutné umístit počátek souřadného systému do těžiště profilu nosníku.



Obr. 11-Výpočet kvadratického momentu setrvačnosti

$$J(y) = \int_S y^2 dS = \int_{-\frac{b}{2}}^{\frac{b}{2}} \int_{-\frac{h}{2}}^{\frac{h}{2}} y^2 dy dz = \int_{-\frac{h}{2}}^{\frac{h}{2}} [z \cdot y^2]_{-\frac{b}{2}}^{\frac{b}{2}} dy = \left[\frac{by^3}{3} \right]_{-\frac{h}{2}}^{\frac{h}{2}} = \frac{bh^3}{12}$$

Průhybovou křivku $w(x)$ lze vypočítat dle vztahu:

$$w''(x) \approx \frac{1}{\rho} = \frac{M_0(x)}{EJ(y)} = \frac{F(l-x)}{EJ(y)}$$

Potom rovnice průhybové křivky bude:

$$w(x) = \frac{Flx^2}{2EJ(y)} - \frac{Fx^3}{6EJ(y)} \quad (2)$$

Napětí na nosníku pak:

$$\sigma_x(y) = \frac{M_0(x)y}{J(y)} \quad (3)$$

Deformace:

$$\varepsilon_x(x, y) = \frac{M_0(x)y}{EJ_y} \quad (4)$$

3 Numerické metody pro řešení úloh elasticity

3.1 Přehled metod

3.1.1 Metoda konečných prvků

Metoda konečných prvků je standard v oblasti řešení inženýrských úloh pružnosti a pevnosti pevných látek. Jedná se o numerickou metodu, která slouží k simulaci průběhů napětí a deformací, vlastních frekvencí, proudění tepla nebo šíření elektromagnetického pole. Je to přibližná metoda pro řešení problémů popsaných diferenciálními rovnicemi. Metoda typicky slouží k hledání kriticky namáhaných míst. Její teoretické základy navrhl již v roce 1943 Richard Courant, k první reálné implementaci mohlo ovšem dojít až s nástupem výpočetní techniky (USA 60-tá léta, NASA – program NASTRAN, pro potřeby kosmického programu). Princip spočívá v rozdělení (diskretizaci) spojitého prostoru tělesa (kontinua) do určitého, tedy konečného počtu prvků – podobojmů pomocí geometrické výpočetní sítě. Hledané parametry se pak zjišťují v uzlových bodech sítě.

3.1.2 Metoda konečných objemů

Jako v případě metody konečných prvků používá geometrickou diskretizační výpočetní síť a aproximuje se tok přes hranice kontrolního objemu, přičemž kontrolní objem je prostorový element výpočetní sítě. Používá se v první řadě pro numerické simulace dynamiky tekutin – CFD (Computational fluid dynamics). Jako v případě MKP jsou pohybové, energetické, nebo transportní parciální diferenciální rovnice pro spojitě prostředí diskretizovány do soustav algebraických rovnic. Tuto metodu používá i zde použitý CFD software OpenFoam.

3.1.3 Metoda konečných diferencí

Je to nejstarší metoda pro diskretizaci parciálních diferenciálních rovnic. Podstata této metody je založená na diskretizaci derivačních členů rovnice. Parciální derivace tedy převedeme na difference. Použijeme diferenční vzorce pro numerický výpočet derivace. Odvození diferenčních vzorců provedeme například pomocí Taylorovy řady. Nevýhodou této metody je, že je použitelná jen na dostatečně jednoduchých geometriích.

3.2 Síťování

Vygenerování výpočetní sítě na trojrozměrném modelu je jedním z prvních kroků při přípravě prostorových dat pro numerická řešení různých fyzikálních simulací metodami konečných prvků nebo konečných objemů na daném modelu. Většinou se jedná o pevnostní výpočty, výpočty vedení tepla, simulace složitých problémů lineární elasticity. Celý postup přípravy zadané

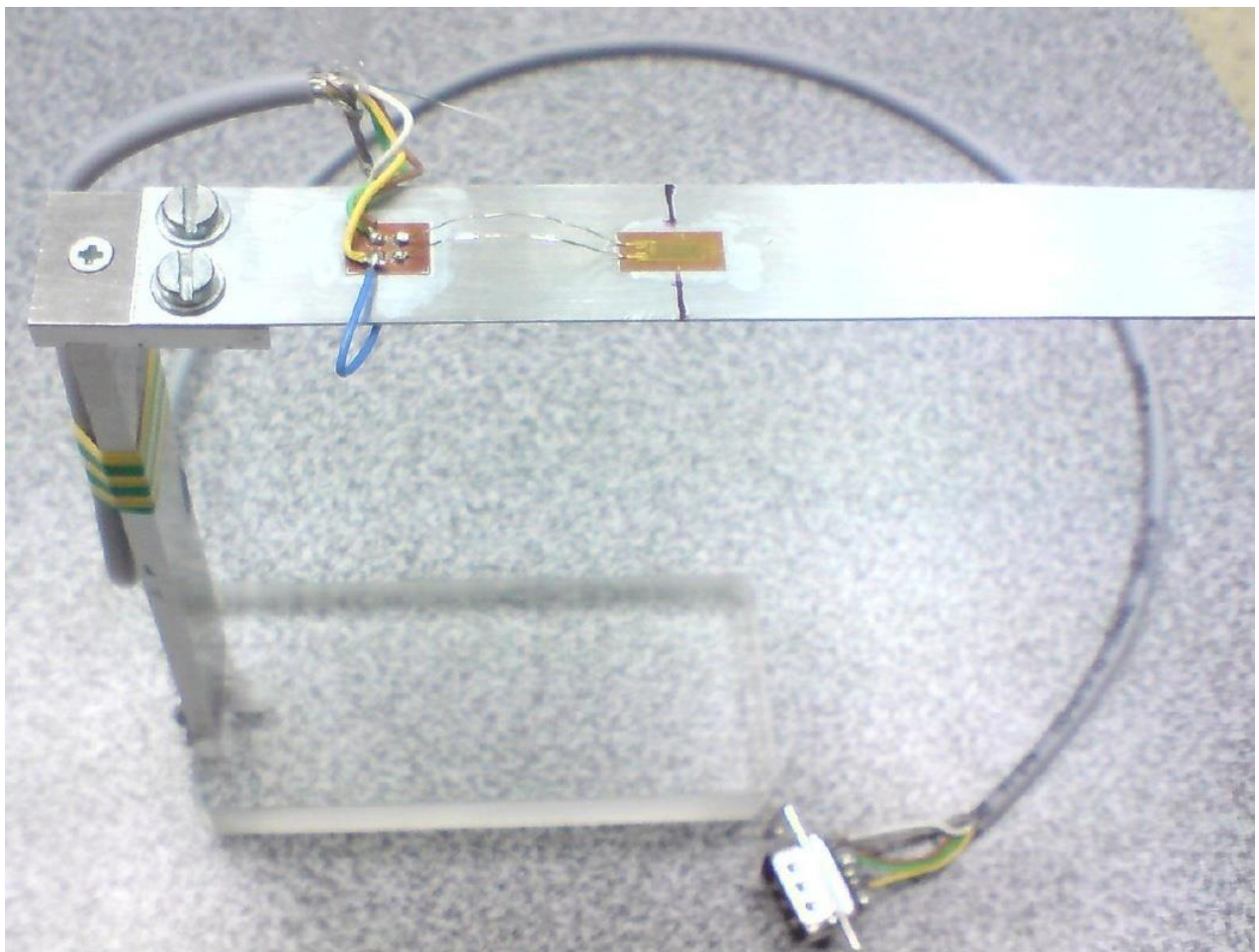
simulace je následující: pomocí některého CAD softwaru vytvoříme geometrický model objektu, na kterém chceme provést simulaci. Celou geometrii tohoto modelu poté proložíme sítí, která nám rozdělí daný objem na jednotlivé elementy-podobjemy, které na sebe beze zbytku navazují. Tvary těchto elementárních podobjemů mohou být různé: čtyřstěny, hranoly-pětistěny, kvádry nebo i obecně šestistěny, které mohou a nemusí mít pravoúhlou strukturu, nebo obecné polyhedrální elementy, to vše s délkami hran přesně definovanými nebo s náhodně proměnnými, dle složitosti geometrie modelu tělesa. Čím jemnější síť, tím je výpočet přesnější, ale samozřejmě i výpočetně a paměťově náročnější. Software, který toto provede se obecně nazývá generátor sítě (mesher). Generátor sítě může být součástí CAD softwaru, ale i samostatný modul. Tímto máme vytvořenou strukturu sousedících domén, na které bude probíhat výpočet. Tento vlastní výpočet zajišťuje software, který se obecně jmenuje solver-řešič. Výsledky takové simulace je třeba nějak prezentovat-zobrazit a to zajišťuje postprocesor-nástroj pro vizualizaci dat. Tedy celý výpočetní řetězec je následující: CAD → Generátor sítě → Řešič → Postprocesor.

Velká komerční softwarová řešení (například ANSYS) zahrnují většinou všechny čtyři moduly. Ale nemusí to tak být. Mohou to být i 4 různé programy, které si navzájem předávají data v různých souborových formátech. Těchto formátů je poměrně mnoho, ať již po komunikaci CAD → Generátor sítě, nebo pro komunikaci Generátor sítě → Řešič, nebo Řešič → Postprocesor. Interní struktury těchto souborů jsou různé: od binárních proprietárních, po lidsky čitelné textové soubory i se skriptovací funkcí. Binární soubory mají velmi rády komerční softwarové produkty. Nesou jim totiž samé výhody: není jim vůbec rozumět (ochrana duševního vlastnictví) a velice rychle se načtou a začlení do struktur systému. Čili při volbě složení výpočetního řetězce je také třeba zajistit, aby výstup z jednoho modulu byl srozumitelný modulu následujícímu. V našem případě pro tvorbu geometrie a sítě a budeme používat software Gmsh [3], což je případ CADu i generátoru zároveň, pro vlastní výpočet balík OpenFoam a pro vizualizaci software Paraview. Všechny tři uvedené softwarové balíky patří do rodiny programů s otevřenými zdrojovými kódy a jejich interoperabilita je prověřena.

4 Výpočet pole napětí a deformace při zatížení nosníku příčnou silou.

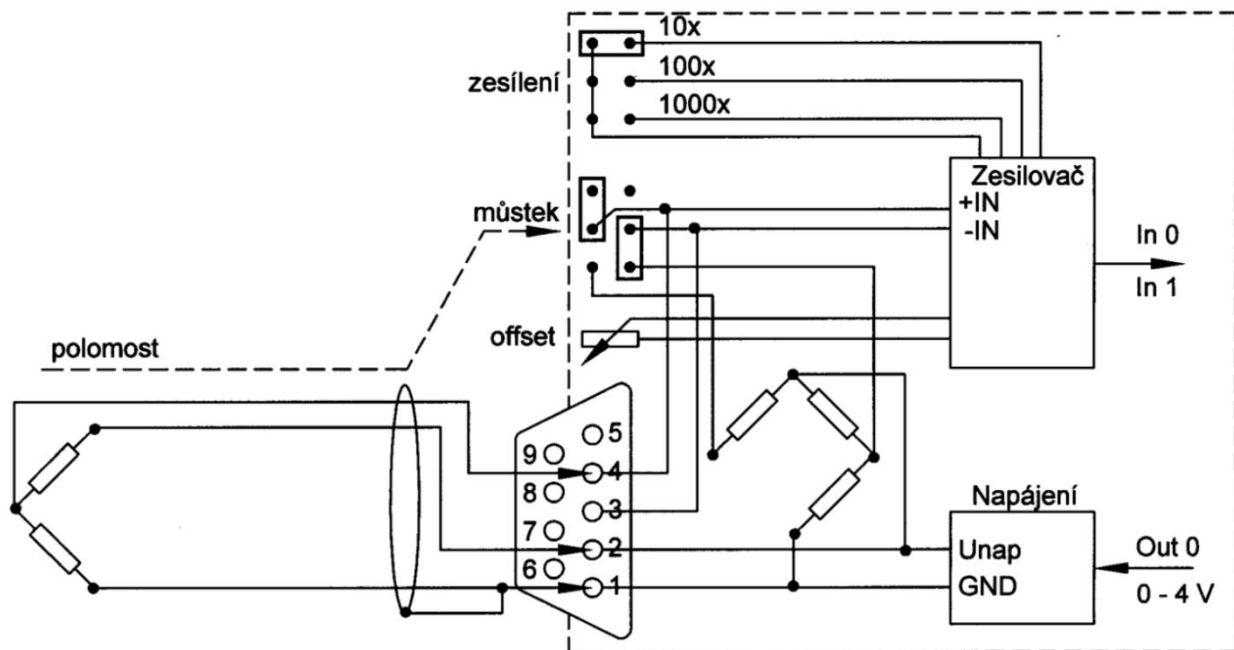
4.1 Popis problému, měření

Laboratorní přípravek-malý ocelový nosník s obdélníkovým průřezem, na který je nalepen odporový tenzometr, je zatížen známou silou. Deformaci vyvolanou prohnutím změříme a vypočteme napětí v místě měření. Toto napětí pak vypočteme analyticky a na závěr numerickou simulací metodou konečných objemů. Výsledky porovnáme a zhodnotíme použití dané numerické metody.

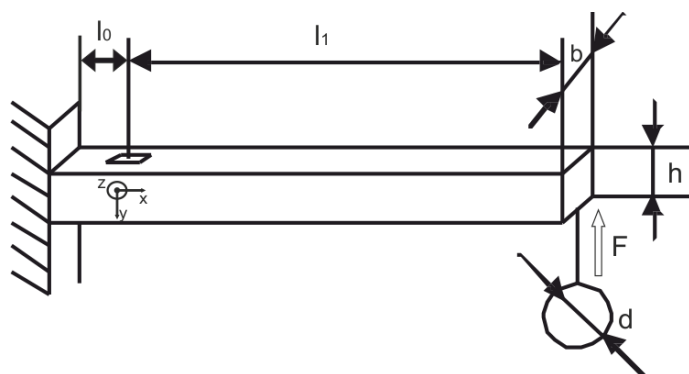


Obr. 12-Nosník s nalepeným tenzometrem

Pro měření namáhání vetknutého nosníku jsou použity dva odporové tenzometry zapojené do polomůstku. Jako zdroj mechanické síly je použit pingpongový míček ponořený do vody. Záznam z tenzometru je proveden pomocí měřicí PC karty.



Obr. 13-Zapojení tenzometrů



Obr. 14-Nosník

$$l_0 = 0,043 \text{ m}$$

$$l_1 = 0,142 \text{ m}$$

$$h = 0,0005 \text{ m}$$

$$b = 0,02 \text{ m}$$

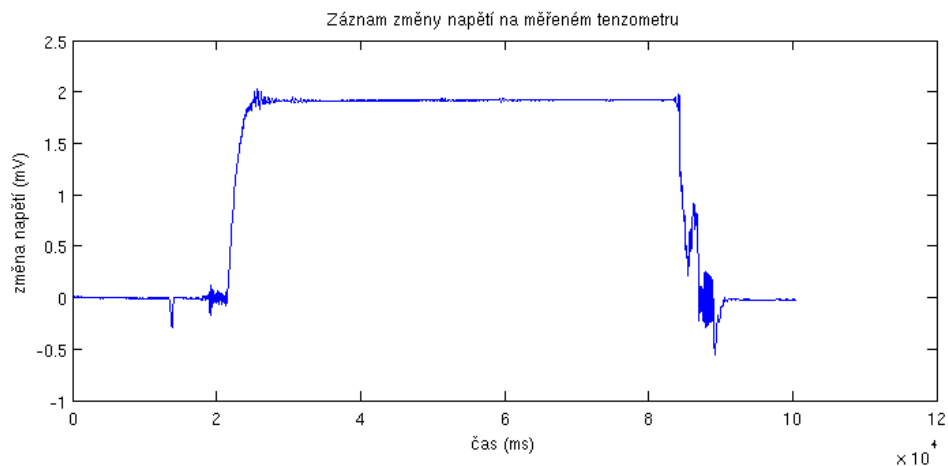
$$d = 2r = 0,04 \text{ m}$$

$$\rho_{\text{voda}} = 1000 \text{ kg} \cdot \text{m}^{-3}$$

$$F = m \cdot g = V \cdot \rho \cdot g = \frac{4}{3} \pi r^3 \rho g = 0,3284 \text{ N}$$

$$E = 210 \text{ GPa}$$

Záznam z tenzometrů:



Obr. 15-Záznam měření tenzometru

Střední hodnota změny napětí po vložení míčku do vody a ustálení pohybu soustavy:

$$\Delta U = 0,001921 \text{ V}$$

Vztah pro výpočet mechanické deformace v závislosti na změně napětí na odporovém tenzometru v polomůstku, kde k je deformační součinitel tenzometru, $k = 2,08$:

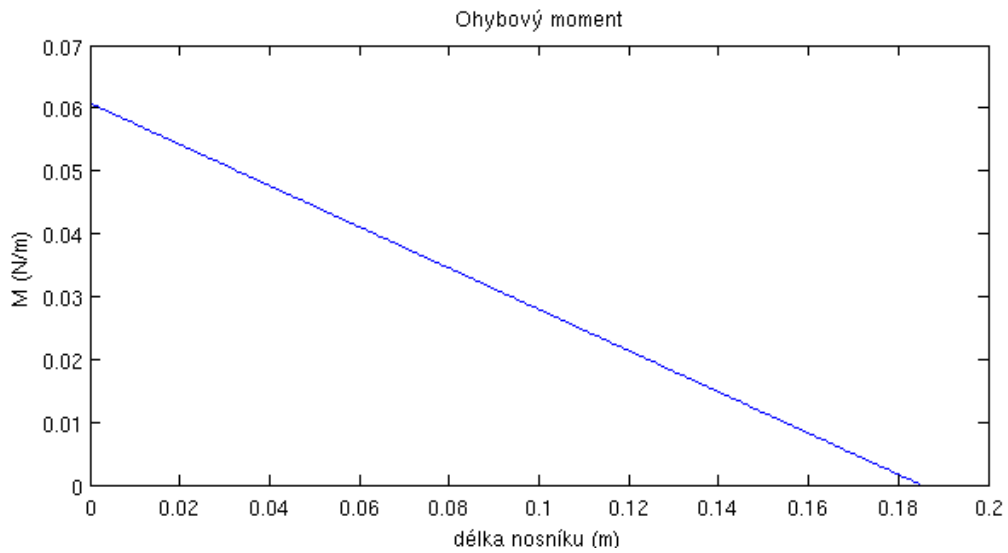
$$\frac{\Delta U}{U} = \frac{k \Delta l}{2 l} = \frac{k \cdot \varepsilon}{2} \quad , \quad \varepsilon = \frac{2\Delta U}{U \cdot k} = 4,61778 \cdot 10^{-4} \text{ strain}$$

$$\sigma_{xx} = \varepsilon \cdot E = 96,97338 \text{ MPa}$$

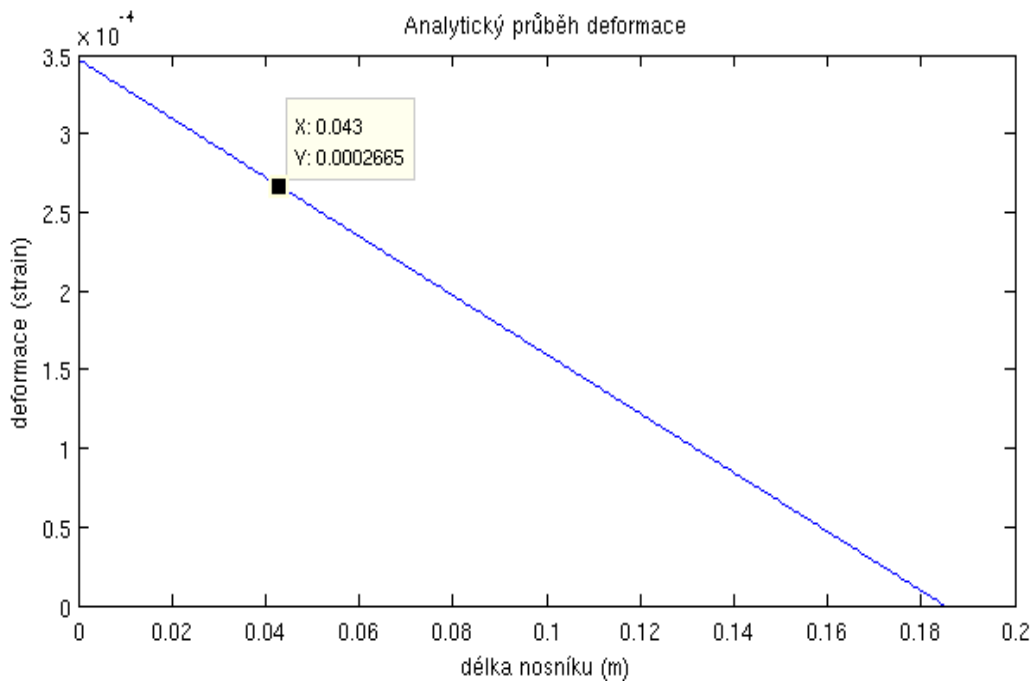
Dále byl změřen průhyb v místě působení síly $\Delta D = 0,026 \text{ m}$

4.2 Analytické řešení

U analytického řešení využijeme funkční vztahy uvedené výše. Vypočteme průběh ohybového momentu dle (1), průběh deformace (4) a napětí (3) na povrchu nosníku a průhybovou křivku (2).

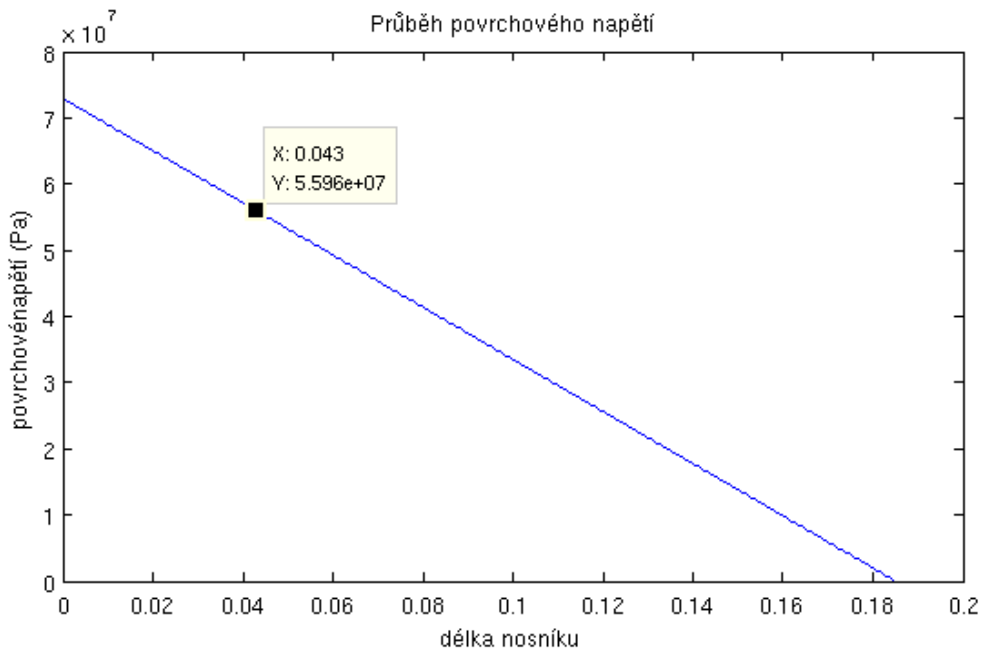


Obr. 16 Analytický průběh zatěžovacího momentu



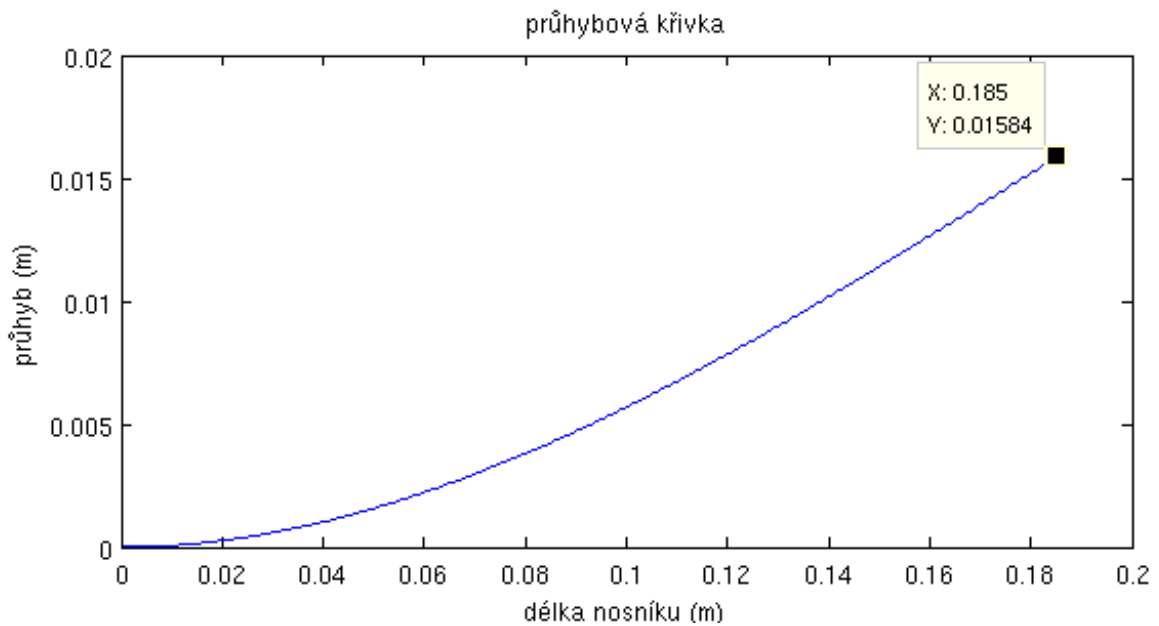
Obr. 17-Analytický průběh deformace

Na obr.17 je znázorněn analytický průběh deformace ε_x na povrchu nosníku $y = \frac{h}{2}$ podél střednice $z = 0$. Střed tenzometru je umístěn v bodě $(0,0043; 0,00025; 0)$



Obr. 18-Analytický průběh napětí

Obr.18 znázorňuje analyticky vypočtený průběh napětí σ_{xx} na povrchu nosníku $y = \frac{h}{2}$ podél střednice $z = 0$.



Obr. 19-Průhybová křivka

Na obr.19 je analyticky vypočtený průběh průhybové křivky s vyznačenou hodnotou průhybu konce nosníku.

Shrnutí analyticky vypočtených hodnot pro bod nalepení tenzometru:

$$\begin{aligned}\varepsilon_x(x, y, z) &= \varepsilon_x(0,0043; 0,00025; 0) = 266,5 \mu\text{strain} \\ \sigma_x(x, y, z) &= \sigma_x(0,0043; 0,00025; 0) = 55,96 \text{ MPa}\end{aligned}$$

Velikost průhybu konce nosníku je pak:

$$w(x = 0,185) = 0,01584m$$

4.3 Numerické řešení

Analytické řešení vypočtené v předchozí kapitole slouží jako soubor referenčních hodnot, se kterými se bude porovnávat výsledek numerické simulace metodou konečných objemů. Tato metoda vyžaduje použití geometrické výpočetní sítě. Tudíž jako první krok je nutné vytvořit v CAD softwaru virtuální model nosníku a pokrýt tento model výpočetní sítí.

4.3.1 Generování výpočetní sítě pomocí Gmsh

Gmsh je open-source trojrozměrný generátor sítě konečných prvků se zabudovaným CAD-modulem a postprocesorem. Jeho vývoj započal v roce 1996. Tehdejší softwarová nabídka byla buďto omezená na velmi drahé komerční balíky nebo na shareware a freeware, který měl omezenou funkčnost nebo jednostranné zaměření. Čili pouze CAD nebo pouze 2D-generátor sítě nebo pouze 3D-generátor sítě nebo pouze postprocessor. Cílem autorů tedy bylo stvořit rychlý, lehký a uživatelsky přívětivý nástroj pro generování výpočetních sítí s parametrickým vstupem a pokročilými vizualizačními schopnostmi. Vývoj vede Christophe Geuzaine a Jean-Francois Remacle, v současné době profesori na univerzitách v Liege a Louvain-la-Neuve v Belgii společně se stovkami uživatelů a vývojářů po celém světě. Kernel a uživatelský interface je napsán v C++, jádro používá knihovny lineární algebry BLAS a LAPACK, grafické rozhraní pak OpenGL a toolkit FLTK.

Výše uvedený software a postup bude použit pro numerickou simulaci mechanického namáhání tenzometrického přípravku, na kterém bude tato simulace poté reálně ověřena. Jedná se o ocelový pásek z jedné strany vetknutý a z druhé zatížený příčnou silou. Takže z hlediska jeho popisu v Gmsh se jedná o obyčejný kvádr. Postupovat budeme tedy tak, že nejdříve nadefinujeme v prostoru kvádr, pak si zvolíme požadovaný typ výpočetní sítě a následně pustíme generátor sítě, který nám danou geometrii proloží danou výpočetní sítí.

System Gmsh ukládá geometrii objektů, které vytváříme do textových, čitelných, souborů s příponou *.geo. Pokud se do nich podíváme, tak zjistíme, že jejich obsah vypadá spíše jako

jakýsi program napsaný v nějakém skriptovacím jazyku. Začínáme od nejjednodušších geometrických útvarů, což jsou body, pomocí nich definujeme vektory-úsečky, pomocí úseček trojúhelníky, čtyřúhelníky nebo obecně uzavřené plochy, z ploch poté tvoříme uzavřené objemy. Jednotlivé body nemusíme spojovat přímkami, lze použít různé matematicky definované křivky, včetně Beziérových křivek. Cílem snažení je samozřejmě přenést tvar trojrozměrného objektu, který modelují do vektorového popisu. Pokud se objekt skládá z opravdu jednoduchých geometrických tvarů, není často vůbec nutné nějaké rýsování v GUI prostředí, lze daný objekt jednoduše „naprogramovat“, což bude i tento případ. Můžeme ale tvořit v GUI pak si objekt uložit, Gmsh nám vytvoří .geo soubor, který si pak můžeme ručně upravit (doplnit např. výkresově přesné rozměry). Při tvorbě geometrií v CADu systému Gmsh můžeme tedy použít tyto dvě metody (rýsování v GUI / programování) a jejich kombinace. Programovací přístup má samozřejmě své výhody - zavedení proměnných, které lze pak globálně měnit a používat je k dalším výpočtům, cykly (For a In {b:c}), využívání knihoven různých matematických funkcí atd. Celkově syntax velmi připomíná hybrid linuxového Bashe a C.

První metodou tvorby geometrie v Gmsh je využití grafického rozhraní vestavěného Cadu [4]. Narýsujeme šest bodů tak, aby tvořily kvádr, pospojujeme úsečkami tak, abychom vytvořili hrany kvádrů. Tyto hrany nám ohraničují plochy stěn kvádrů a tyto stěny nám ohraničují objem kvádrů. Každý bod, každá hrana, plocha nebo objem mají své jednoznačné id uvedené v kulatých závorkách. Program je takto čísluje sám automaticky od 1, ale lze je pak samozřejmě ručně libovolně změnit. Následující kód Gmsh vygeneruje narýsováním kvádrů o rozměrech 0.5 x 0.2 x 1.

```
Point(1) = {0, 0, 0, 0.1};           Definice rohů kvádrů ve tvaru:
Point(2) = {1, 0, 0, 0.1};           Point (id) = (x,y,z, lc)
Point(3) = {1, 0.5, 0, 0.1};         lc není souřadnice, je to
Point(4) = {-0, 0.5, 0, 0.1};        parametr pro síťovací modul
Point(5) = {-0, 0.5, -0.2, 0.1};
Point(6) = {0, 0, -0.2, 1};
Point(7) = {1, 0, -0.2, 1};
Point(8) = {1, 0.5, -0.2, 1};
Line(1) = {5, 8};                     Z bodů id=5,8 je vygenerována hrana s id=1
Line(2) = {8, 7};
Line(3) = {7, 6};
Line(4) = {6, 5};
Line(5) = {3, 4};
Line(6) = {1, 1};
```

```

Line(7) = {4, 1};
Line(8) = {1, 2};
Line(9) = {2, 3};
Line(10) = {3, 8};
Line(11) = {4, 5};
Line(12) = {1, 6};
Line(13) = {2, 7};
Line Loop(15) = {1, 2, 3, 4};      Úsečky id=1,2,3,4 ohraničují rovnou plochu
s id=15
Plane Surface(15) = {15};
Line Loop(17) = {1, -10, 5, 11};
Plane Surface(17) = {17};
Line Loop(19) = {10, 2, -13, 9};
Plane Surface(19) = {19};
Line Loop(21) = {11, -4, -12, -7};
Plane Surface(21) = {21};
Line Loop(23) = {5, 7, 8, 9};
Plane Surface(23) = {23};
Line Loop(25) = {8, 13, 3, -12};
Plane Surface(25) = {25};
SurfaceLoop(27) = {25, 23, 17, 15, 19, 21};      Plochy s id=25,23,17,15,19,21
Volume(27) = {27};                              ohraničují objem kvádru s id=27

```

Parametr *lc* je atribut bodu a udává v jaké vzdálenosti bude nejbližší uzel výpočetní sítě, tedy jak velká budou jednotlivá „oka sítě“ v okolí bodu, čili definují jemnost výpočetní sítě.

Druhou metodou pro vytvoření geometrie, je využít programovatelné vlastnosti jazyka Gmsh. V tomto případě nebudeme explicitně definovat všech osm bodů tvořících rohy kvádru, ale bereme kvádr jako obdélník „protlačený“ (extruded) z plochy kolmo do prostoru. Tedy pokud obdélník o rozměrech $a \times b$ protlačíme kolmo do prostoru o Δc dostaneme kvádr o rozměrech $a \times b \times \Delta c$. S konkrétními hodnotami v jednotkách SI pro daný tenzometrický přípravek bude program tvorby geometrie kvádru vypadat takto:

```

// pripravek pro tenzometricka mereni
// NovakVitezslav
// Velikost elementu meshe
LC= 0.001;
//Rozměry přípravku
A = 0.15;
B = 0.02;

```

```

C = 0.0009;
// Definice pudorysu
Point(1) = {0, 0, 0, LC};
Point(2) = {A, 0, 0, LC};
Point(3) = {A, B, 0, LC};
Point(4) = {0, B, 0, LC};

Line(1) = {1, 2};
Line(2) = {2, 3};
Line(3) = {3, 4};
Line(4) = {4, 1};

Line Loop(6) = {1, 2, 3, 4};
Plane Surface(6) = {6};
// Vyroba z obdelniku kvadr
Extrude {0,0,C} { Surface{6};}

```

Extrude je příkaz, který nám protlačí plochu s id=6 do prostoru ve směru vektoru (0,0,C), čili kolmo k ploše ve směru osy z. Příkaz extrude dokáže poměrně hodně a má mnoho voleb, s pomocí kterých se dá s plochou při extruzi různě manipulovat, hlavně tedy rotovat a extrudovat podél námi definované křivky. Tutoriály po nainstalování Gmsh lze nalézt zde [5].

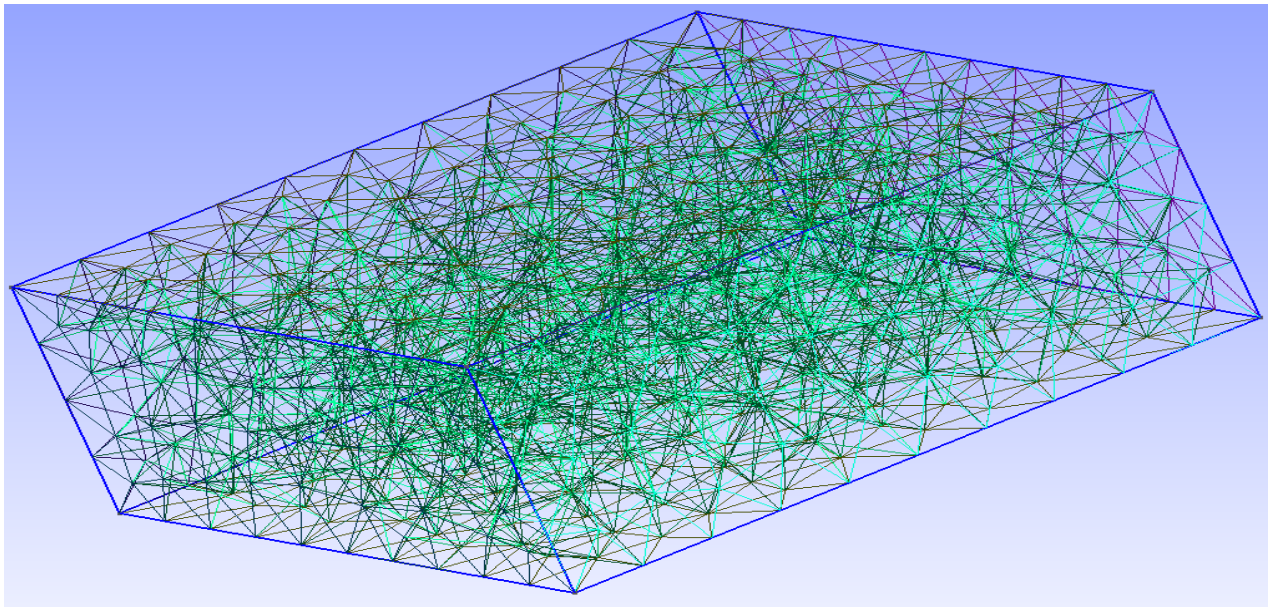
Tímto máme samotnou geometrii přípravku nadefinovanou a připravenou pro pokrytí výpočetní sítí. Výše uvedený kód bude rozšířen o příkazy, které se nepromítají do geometrie tělesa, ale slouží jako řídicí klíčová slova, inicializační parametry pro modul generátoru sítě. Ve chvíli, kdy je připravena geometrie objektu, lze přistoupit k vlastnímu generování výpočetní sítě.

Výpočetní síť nám rozdělí objem tělesa na systém vzájemně sousedících elementů, které dohromady tvoří geometrii tělesa a přes které probíhá výpočet. Cílem je pomocí výpočtů algebraických rovnic mezi těmito malými doménami aproximovat s jistou zvolenou přesností chování celého tělesa, protože odpovídající parciální diferenciální rovnice na složitých geometriích nelze obvykle analyticky řešit. Tvar jednotlivých elementů závisí na zvoleném algoritmu generátoru sítě. Základní rozdělení těchto sítí je na strukturované a nestrukturované. Strukturované si pak Gmsh interně dělí na transfinite a extrudované.

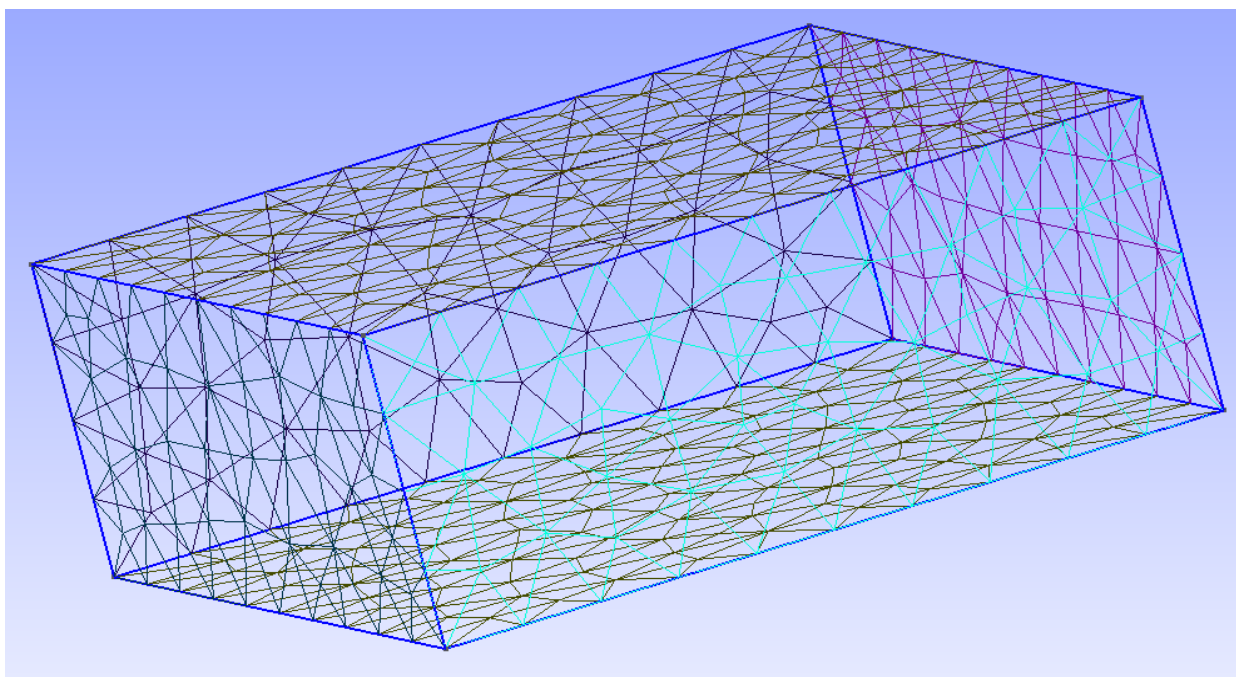
4.3.2 Základní typy výpočetních sítí

Nestrukturovaná síť

Nestrukturovaná síť je v Gmsh základním typem sítě pro zasítování, pokud neřekneme jinak. Je to síť čtyřstěňů nebo šestistěňů, jejichž strany jsou různostranné trojúhelníky nebo čtyřúhelníky. Defaultní nastavení pracuje s trojúhelníky. Převod z trojúhelníků na čtyřúhelníky se děje pomocí klíčového slova `Recombine` s parametrem plochy, kterou chci rekombinovat. V případě použití jakožto součástí příkazu `extrude` pak bez parametru. Vzdálenost nejbližšího uzlu je dána parametrem `LC`. Obr.20 ukazuje příklad nestrukturované sítě tvořené čtyřstěny vygenerované programem Gmsh. Povrchová síť tvořená trojúhelníky je pro tutéž geometrii znázorněna na Obr.21.



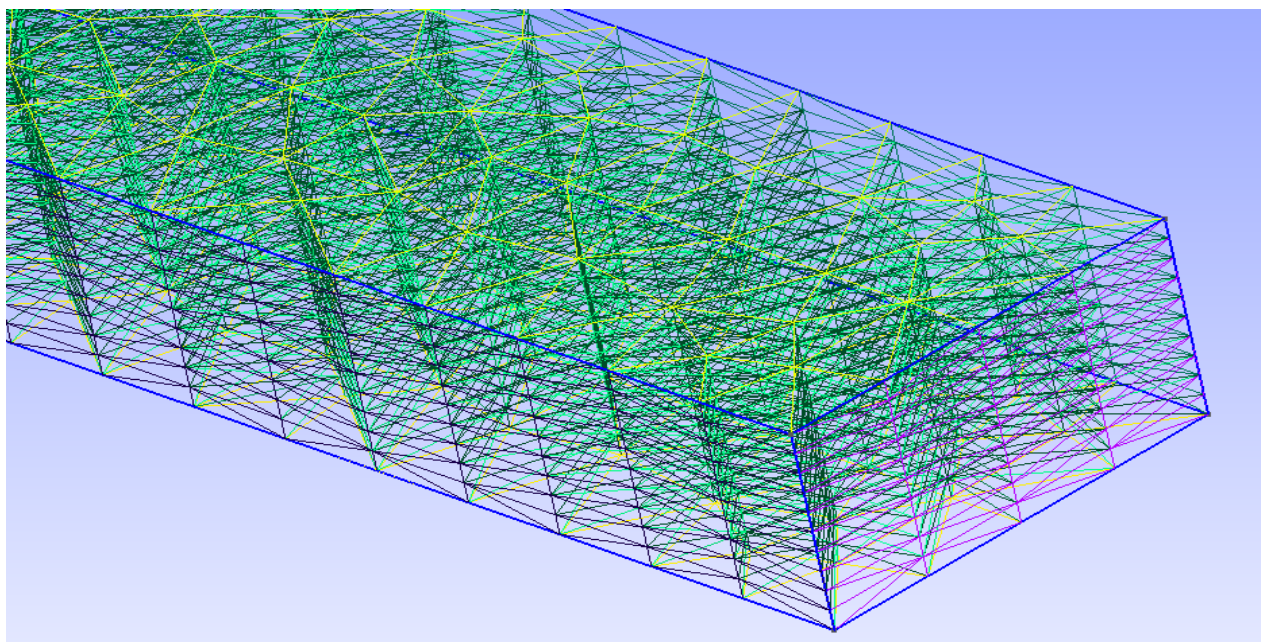
Obr. 20-Příklad nestrukturované tetrahedrální sítě



Obr. 21:- Nestruturovaná síť-pouze povrch

Extrudovaná síť

Extrudovaná síť dodržuje ve vrstvách strukturu sítě plochy, ze které vznikl extruzí daný objem.
(viz.obr. 22)



Obr. 22-Extrudovaná síť z nestruturované sítě

Síť vytvořená extruzí povrchu č. 6 o deseti vrstvách vznikne doplněním následujícího příkazu:

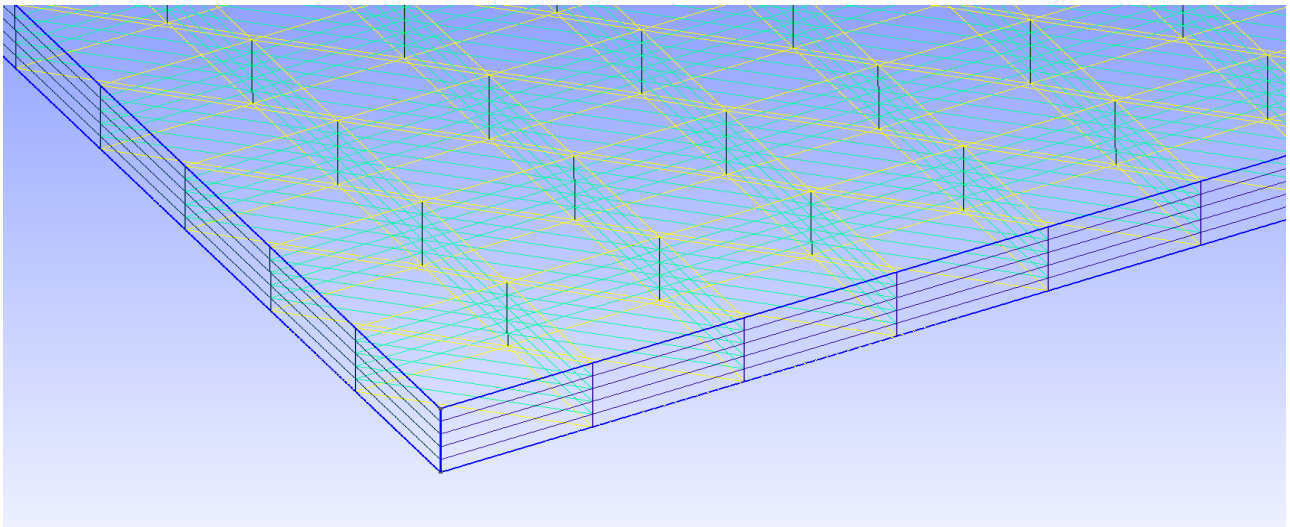
```
Extrude {0,0,C } { Surface{6};Layers{10};}
```

Transfinitní síť

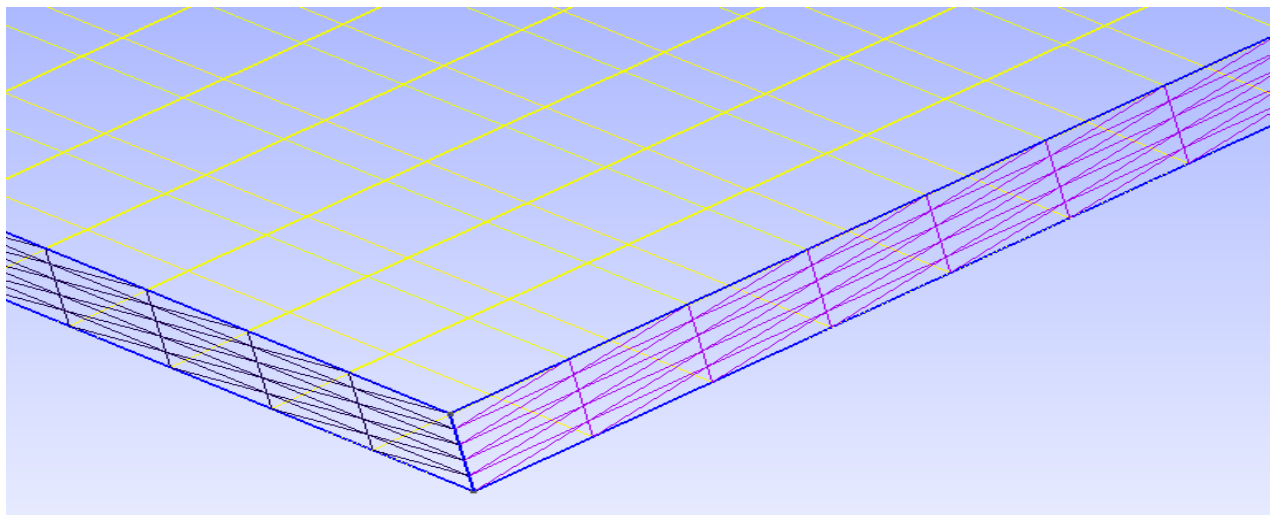
Jako transfinitní síť se v terminologii Gmsh označuje plně strukturovaná síť. Kvádr vznikl extruzí z plochy, kterou jsme definovali jako transfinitní. Vzdálenost k nejbližšímu nodu je dána parametrem LC, pokud však explicitně nedefinujeme hranice plochy jako transfinite lines, jejichž parametry obsahují rozteč dvou nodů a parametr LC je pak ignorován.

Kód:

```
Transfinite Surface {6};  
Recombine Surface{6};  
Extrude {0,0,C} { Surface{6}; Layers{5};}
```



Obr. 23-Transfinitní síť



Obr. 24-Rekombinace do hexahedrální sítě

Finální kód, který vygeneruje geometrii tenzometrického přípravku s přepínači pro generátor sítě a pojmenováním vnějších ploch pro řešič je vypsán v příloze 1.

4.3.3 Výpočet pomocí balíku OpenFoam

OpenFoam CFD Toolbox (OpenFieldOperation and Manipulation) je softwarový balík CFD řešičů a utilit s otevřenými zdrojovými kódy, distribuovaný a spravovaný neziskovým sdružením OpenFoamFoundation a vyvíjený společností OpenCFD Ltd, pod licencí GNU-GPL. Má širokou uživatelskou základnu, jak z akademické, tak průmyslové a komerční sféry. OpenFoam má rozsáhlý soubor nástrojů pro výpočet komplexních úloh proudění stlačitelných i nestlačitelných tekutin, zahrnující chemické reakce, turbulence, hoření a přenosy tepla. Obsahuje nástroje pro řešení úloh dynamiky pevných látek a šíření elektromagnetického pole, vlastní generátor výpočetních sítí a pre- a post-processor. Řešiče a utility balíku OpenFoam jsou naprogramovány v C++ s tím, že při jejich návrhu byl implementován požadavek pro paralelizaci běhu na moderních výpočetních architekturách. V této práci je použita verze 2.2.2, která byla zkompileována ze zdrojových kódů v prostředí operačního systému Slackware14.1-x86. Balík OpenFoam obsahuje přes 80 řešičů, které simulují specifické problémy a přes 170 utilit, které slouží pro generování sítí, vizualizaci a konverze formátů. OpenFoam nemá žádné specifické GUI, jedná se tedy o programy pracující v příkazové řádce. Vstupní a výstupní soubory jsou čitelné textové soubory.

4.3.4 Řešení úlohy pomocí řešiče solidDisplacementFoam

Řešiče (solvery) jsou v OpenFoamu samostatné programy, které provedou vlastní výpočet pole hledaných veličin a parametrů pro každý kontrolní objem nebo uzel výpočetní sítě. Definice geometrie výpočetní sítě, definice okrajových podmínek a materiálových konstant jsou vstupní informace řešiče. Z hlediska postupu výpočtu je lze rozdělit na sdružené a segregované. Segregované, oddělené, řešení znamená, že rovnice pro určitou neznámou proměnnou se vypočte pro všechny kontrolní objemy sítě a pak teprve se pokračuje další neznámou proměnnou. Tedy například nejprve se zvlášť spočítá σ_{xx} pro všechny kontrolní objemy, pak σ_{xy} ..atd. Po vypočtení všech proměnných se vyhodnotí, zda řešení konverguje a pokud ne dojde k dalšímu iteračnímu průchodu. Sdružený řešič vypočte najednou všechny neznámé proměnné pro daný kontrolní objem a pak pokračuje na další. Stejně jako v případě segregovaného řešiče, po projití celé geometrie se zjišťuje konvergence řešení a nastavuje další iterační průchod. Pro potřeby simulace zatížení nosníku příčnou silou použijeme v prostředí OpenFoam řešič solidDisplacementFoam, což je segregovaný řešič lineární elasticity malých deformací pevných těles metodou konečných objemů. solidDisplacementFoam je i název spustitelného binárního souboru. Na rozdíl od utilit, kde za jménem spustitelného souboru můžeme očekávat nějaké parametry, řešič pouštíme bez parametru a veškeré nastavení simulace se děje pomocí adresářové a souborové struktury, která je

specifická pro daný typ úlohy a v jejímž kořeni řešič spustíme. Standartní umístění kořenového adresáře pro simulace je v verzi 2.2.2:

```
/$HOME/OpenFOAM/$USER-2.2.2/run
```

Tutoriály v kterých nalezneme vzorové ukázky inicializačních adresářových struktur jsou součástí instalace a nalezneme je zde:

```
/opt/OpenFOAM/OpenFOAM-2.2.2/tutorials
```

V případě řešiče `solidDisplacementFoam` pak:

```
/opt/OpenFOAM/OpenFOAM-2.2.2/tutorials/stressAnalysis/solidDisplacementFoam/plateHole
```

Počáteční struktura, kterou převezmeme z tutoriálu a umístíme do adresáře

```
/$HOME/OpenFOAM/$USER-2.2.2/run
```

bude vypadat takto:

```
|-- 0
|   |-- D
|   `-- T
|-- constant
|   |-- mechanicalProperties
|   |-- polyMesh
|   |   |-- blockMeshDict
|   |   `-- boundary
|   `-- thermalProperties
`-- system
    |-- controlDict
    |-- fvSchemes
    |-- fvSolution
    |-- sampleDict
    `-- sampleSurfaceDict
```

Jako první krok provedeme import výpočetní sítě z programu Gmsh pomocí OpenFoam utility `gmshToFoam`, která síť zkonvertuje do Foam-formátu a uloží do adresáře `./constant/polyMesh`. Soubor s výpočetní sítí `pripravek.msh` si nakopírujeme do kořene simulace.

```
/$HOME/OpenFOAM/$USER-2.2.2/run/gmshToFoam pripravek.msh
```


Po importu geometrie sítě přípravku se struktura změní následujícím způsobem:

```
|-- 0
|   |-- D
|   `-- T
|-- constant
|   |-- mechanicalProperties
|   |-- polyMesh
|   |   |-- boundary
|   |   |-- cellZones.gz
|   |   |-- faceZones.gz
|   |   |-- faces.gz
|   |   |-- neighbour.gz
|   |   |-- owner.gz
|   |   |-- pointZones.gz
|   |   |-- points.gz
|   |   `-- sets
|   |       `-- deskA.gz
|   `-- thermalProperties
|-- system
|   |-- controlDict
|   |-- fvSchemes
|   |-- fvSolution
|   |-- sampleDict
|   `-- sampleSurfaceDict
```

Pro spuštění simulace na importované geometrii je nutné dále dodefinovat:

okrajové podmínky – adresář ./0/

materiálové konstanty – adresář ./constant/

parametry algoritmu solveru – adresář ./system/

Zadání okrajových podmínek

```
|-- 0
|   |-- D ... Zadání mechanických okrajových podmínek
|   `-- T ... Zadání termálních okrajových podmínek
```

V souboru D musí jména orientovaných ploch souhlasit se jmény, které byly uvedeny v Gmsh při generování výpočetní sítě. Nadefinujeme vetknutí plochy „uchycení“ parametrem fixedValue a protější stranu „namáhání“ budeme virtuálně zatěžovat silou míčku ponořeného do kapaliny ve směru osy z. Sílu je zde potřeba přepočítat na tlak (síla míčku dělená plochou průřezu nosníku). Soubor D v uveden v příloze 2.

Klíč pro použití definice dimenze pomocí základních jednotek SI:

```
dimensions      [a b c d e f g];
```

$$\text{Dimenze} = kg^a \cdot m^b \cdot s^c \cdot K^d \cdot mol^e \cdot A^f \cdot cd^g$$

$$P = \frac{F}{S} = \frac{F}{b \cdot h} = 18244 \text{ Pa}$$

Soubor T není podstatný, veškeré teplotní vlivy jsou v této úloze zanedbatelné a výpočet tepelných procesů v závislosti na mechanickém namáhání bude v této úloze v solveru deaktivován.

Zadání materiálových konstant

```
|-- constant
  |-- mechanicalProperties ... mechanické konstanty
  `-- thermalProperties   ... tepelné konstanty
```

V souboru `mechanicalProperties` jsou uvedeny tabulkové hodnoty hustoty (ρ), Poissonova čísla (ν) a Youngova modulu pružnosti (E) pro běžnou konstrukční ocel. Soubor `thermalProperties` opět needitujeme, tepelné výpočty nejsou aktivované. Soubor `mechanicalProperties` je uveden v příloze 3.

Nastavení parametrů chodu řešiče:

```
:
|-- system
  |-- controlDict
  |-- fvSchemes
  |-- fvSolution
  |-- sampleDict
|-- sampleSurfaceDict
```

Soubor `controlDict`:

Zde je nastaven počet iterací výpočtu. Ukázalo se, že v této úloze, kde je dominantní ohyb, řešič metodou konečných objemů konverguje k řešení velice pomalu. Nastavení v tomto případě je 8000000 iterací. Jedna iterace zde odpovídá jednomu časovému kroku a zapsání mezivýsledků je provedeno každých 40000 časových kroků. Soubor `controlDict` je uveden v příloze 4.

Soubor `fvSolution`

Zde byly upraveny parametry `tolerance 1e-11;a D 1e-10;`
Ostatní soubory v adresáři `/system/` není potřeba editovat.

Spuštění výpočtu:

```
/$HOME/OpenFOAM/$USER-2.2.2/run/solidDisplacementFoam > log.txt
```

Po spuštění se začnou tvořit adresáře s mezivýpočty:

```
.
|-- 0
|   |-- D
|   `-- T
|-- 40000
|   |-- D.gz
|   |-- sigma.gz
|   |-- sigmaEq.gz
|   `-- uniform
|       `-- time.gz
|-- 80000
|   |-- D.gz
|   |-- sigma.gz
|   |-- sigmaEq.gz
|   `-- uniform
|       `-- time.gz
.
.
.
```

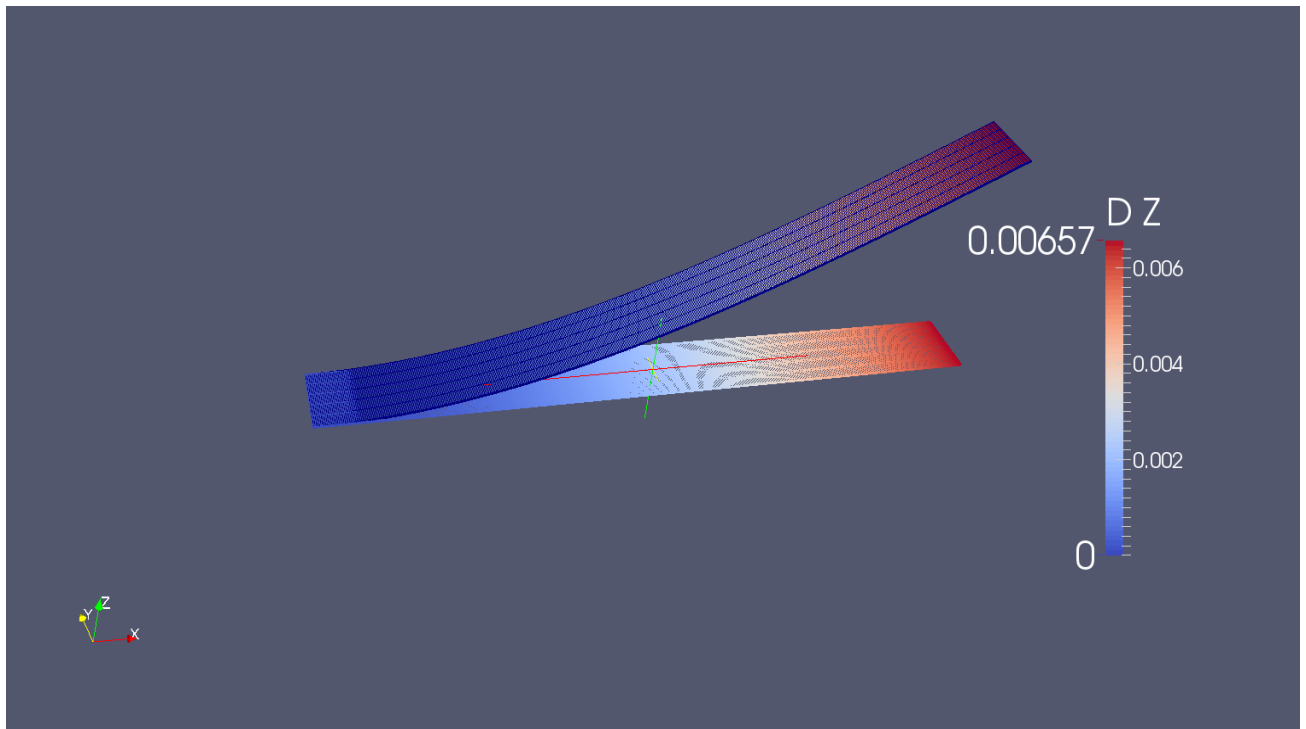
Každý adresář s mezivýpočtem obsahuje soubor vektorů posunutí a soubor tenzorů napětí pro každý element výpočetní sítě.

4.3.5 Vizualizace výsledku výpočtu v Paraview

K prezentaci výsledků použijeme program Paraview, což je opensource interaktivní vizualizační program, podporující nativně výstup z OpenFoamu. Vytvoření nového projektu provedeme založením prázdného souboru s příponou `.foam` v kořenovém adresáři OpenFoam simulace:

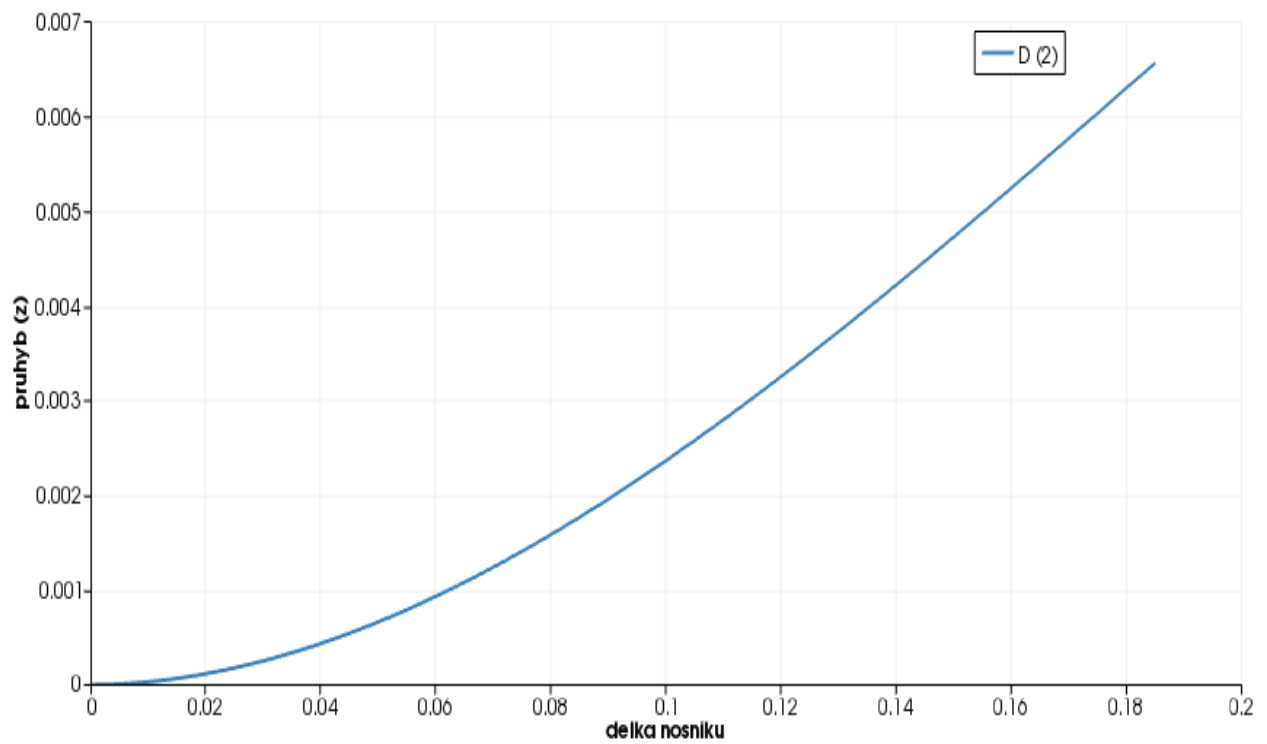
```
$ touch /$HOME/OpenFOAM/$USER-2.2.2/run/nosnik.foam
```

Po otevření souboru `nosnik.foam` v Paraview jsou automaticky načteny výsledky simulace včetně geometrie přípravku. Zobrazení vypočtených hodnot se provádí pomocí filtrů. K zobrazení průhybu nosníku použijeme filtr *Warp by vector*. Pro zobrazení průhybové křivky a povrchového napětí σ_{xx} podél střednice $y=0,01$ filtr *Plot over line*. Použijeme hodnoty z konečného stavu simulace. Pro zobrazení průběhu konvergence výpočtu aplikujeme filtr *Plot selection over time* na polohu koncového bodu nosníku a budeme sledovat změnu jeho polohy v průběhu simulace. Nakonec provedeme řez (*slice*) v místě nalepení tenzometru a zjistíme hodnoty povrchového napětí v daném místě.

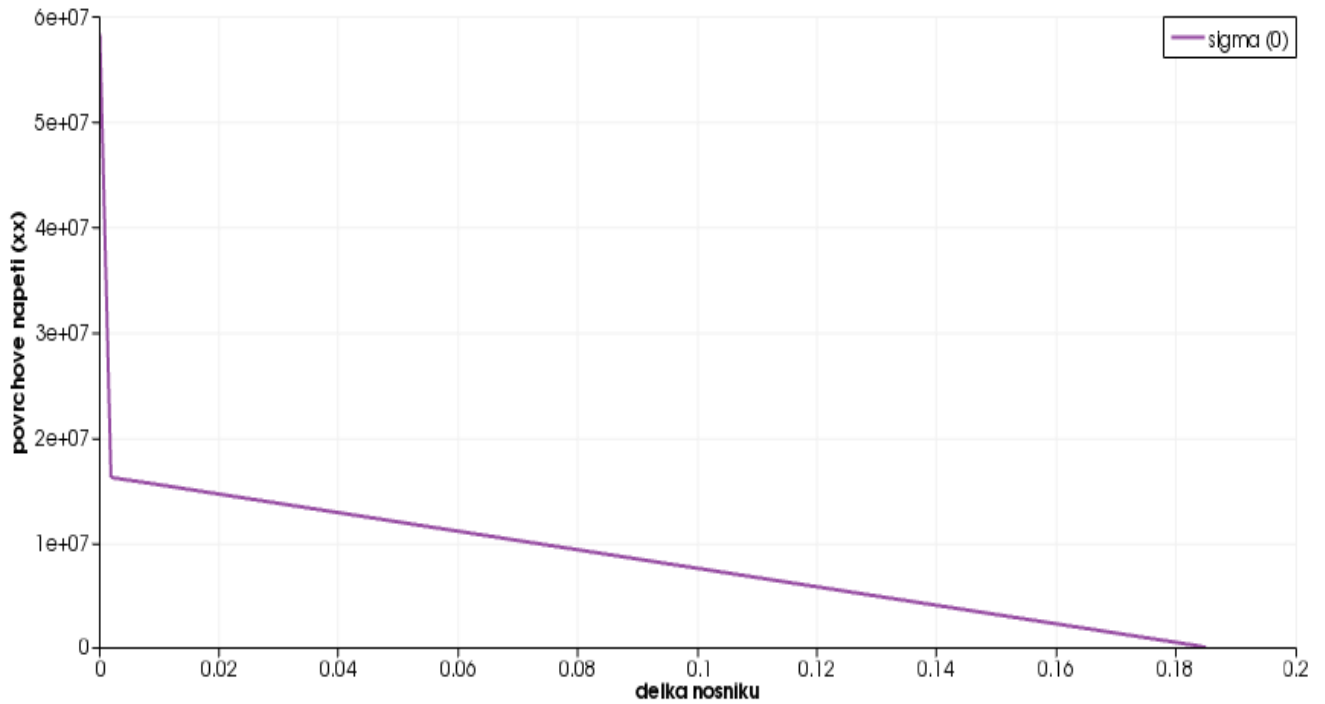


Obr. 25- Průhyb nosníku

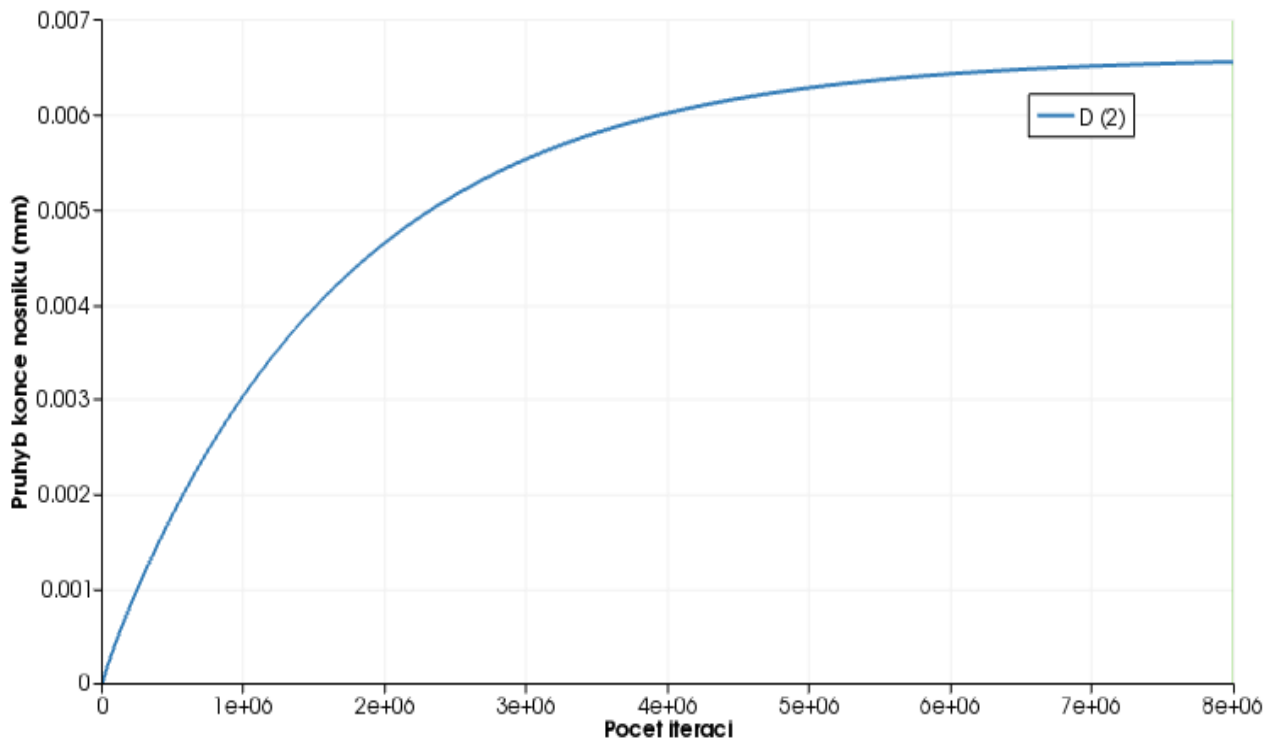
Velikost průhybu konce nosníku $d_z = 6,57mm$



Obr. 26- Průhybová křivka



Obr. 27-Průběh povrchového napětí



Obr. 28-konvergence výpočtu

Hodnota napětí odečtená pomocí filtru Slice z nejbližšího nodu místa nalepení tenzometru (0,043146; 0,12 ;0) je $\sigma_{xx} = 12.572$ MPa.

5 Porovnání výsledků a závěr

V této bakalářské práci jsem řešil numerickou simulaci problému lineární elasticity na jednoduché geometrii v prostředí opensource balíku OpenFoam, pracujícího metodou konečných objemů. Výsledky numerického řešení jsou pak porovnány s analytickým řešením a s výsledkem tenzometrického měření, přičemž analytické řešení považujeme za referenční. Numerické řešení i měření jsou zatíženy chybami.

Tabulka 1: Porovnání analytického výpočtu, experimentální hodnoty a numerického výpočtu pro průhyb nosníku .

Průhyb konce nosníku (m)	
Vypočtený analyticky	0,016
Změřený	0,026
Vypočtený numericky	0,007

Tabulka 2: Porovnání analytického výpočtu, experimentální hodnoty a numerického výpočtu pro velikost napětí na povrchu nosníku.

Povrchové napětí σ_{xx} v místě nalepení tenzometru (MPa)	
Vypočtené analyticky	55,9
Změřené	96,9
Vypočtené numericky	12,5

Jak je vidět, z Tabulek 1 a 2, získané hodnoty se dosti liší, což se děje z následujících důvodů:

Měření se odchyluje od analytického výpočtu z těchto důvodů:

- zatížení mění v čase orientaci (při průhybu konce nosníku nezůstává síla vertikální)
- nosník je v místě umístění tenzometru z důvodu dobrého nalepení zeslaben broušením, což je podstatný faktor.

Numerické řešení se odchyluje od analytického výpočtu z těchto důvodů:

- nesouhlasí okrajová podmínka (síla mění orientaci v čase),
- řešič je navržen pouze pro malé výchyly.

Dále jakožto segregovaný řešič, počítající po kartézských složkách, konverguje k řešení velice pomalu. Zde uvedených 8 milionů iterací počítal Intel i5-3Ghz zhruba 12 hodin. Řešení je navíc silně závislé na typu výpočetní sítě. Nestrukturovaná síť dokonce dokáže způsobit

systémové zhavarování (výjimka v plovoucí čárce). Celkově lze prohlásit, že metoda konečných objemů implementovaná v oficiální distribuci výpočetního balíku OpenFOAM verze 2.2 není pro řešení problémů lineární elasticity příliš vhodná, a to zejména pro situace, kdy převládající směr napětí je orientován kolmo na směr působící síly (to jest kolmo na převládající složku vektoru posunutí), což je právě případ nosníku namáhaného na ohyb. Kvalitativně je numerické řešení průhybu a povrchového napětí v pořádku. Průhybové křivky mají stejný tvar a průběh povrchového napětí je lineární, ačkoliv hodnoty jsou vzhledem k analytickému řešení výrazně nižší. Pokud byla řešena úloha jednoosého namáhání, pak výsledek simulace byl velmi blízko analytického řešení a výpočet konvergoval rychle. Zvláštní pozornost dále zasluhuje obr.27, kde se v okolí místa vetknutí vyskytuje zcela nefyzikální skok na napětí, u kterého se mi nepodařilo najít jeho příčinu. Jedná se o numerickou chybu v místě vetknutí. Při pokusech zjemnit výpočetní síť nebo změnit její typ problém přetrvává.

Použité materiály

- [1] http://www.learneasy.info/MDME/MEMmods/MEM30007A/properties/properties_files/stress_strain_diagram_steel2.GIF
- [2] <http://pmpaspeakingofprecision.files.wordpress.com/2012/03/slip-planes-explain-plasticity-diagram-copy.jpg>
- [3] Christophe Geuzaine and Jean-François Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, [online], URL:<<http://geuz.org/gmsh/>>
- [4] Christophe Geuzaine and Jean-François Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, [online], URL:<<http://geuz.org/gmsh/screencasts/>>
- [5] Christophe Geuzaine and Jean-François Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, [online], <<usr/doc/gmsh/tutorial/>>
- [6] Christophe Geuzaine and Jean-François Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, [online], adresář: <<usr/doc/gmsh/tutorial/t10.geo>>
- [7] Hokr, Milan. Vybrané partie z fyziky, Část „mechanika pružných těles“, studijní text FM 2005.

Přílohy:

Příloha 1: vygenerování geometrie výpočetní sítě modelu vetknutého nosníku

Soubor nosnik.geo

```
// pripravek pro tenzometricka mereni
// Novak Vitezslav M10000289-tul.cz
// Velikost elementu meshe
CL = 0.001;
//Rozměry přípravku
A = 0.185;
B = 0.02;
C = 0.0005;
// Definice pudorysu

Point(1) = {0, 0, 0, CL};
Point(2) = {A, 0, 0, CL};
Point(3) = {A, B, 0, CL};
Point(4) = {0, B, 0, CL};

Line(1) = {1, 2};
Line(2) = {2, 3};
Line(3) = {3, 4};
Line(4) = {4, 1};

Line Loop(6) = {1, 2, 3, 4};
Plane Surface(6) = {6};

// Pokryj plochu transfinite trojúhelníkovou sítí
Transfinite Surface {6};

// Rekombinuj plochu do pravoúhlé čtyřúhelníkové sítě
Recombine Surface {6};

// Vyroba z obdelniku kvadr s pěti interními vrstvami pro mesher
Extrude {0,0,C} { Surface {6}; Layers {2}; Recombine; }

// Pevna definice pravoúhlé transfinite site (LC override)
Transfinite Line {1} = 500 ;
Transfinite Line {2} = 6 ;
Transfinite Line {3} = 500 ;
Transfinite Line {4} = 6 ;
```

```
// Pojmenovani kvádrů a jeho stěn

Physical Surface("top")={6};
Physical Surface("bottom")={28};
Physical Surface("lside")={23};
Physical Surface("rside")={15};
Physical Surface("uchyceni")={27};
Physical Surface("namahani")={19};
Physical Volume("deska")={1};
```

Příloha 2: definice okrajových podmínek geometrie nosníku

Soubor /\$HOME/OpenFOAM/\$USER-2.2.2/run/0/D

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.1.0 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n |
\*-----*/
FoamFile
{
version 2.0;
format ascii;
classvol VectorField;
object D;
}
// * * * * * //

dimensions [1-1 -2 0 0 0 0];

internalField uniform (0 0 0);

boundaryField
{
bottom
{
type tractionDisplacement;
traction uniform (0 0 0);
pressure uniform 0;
value uniform (0 0 0);
}

top
{
type tractionDisplacement;
traction uniform (0 0 0);
pressure uniform 0;
value uniform (0 0 0);
}

rside
{
type tractionDisplacement;
traction uniform (0 0 0);
pressure uniform 0;
value uniform (0 0 0);
}

lside
{
type tractionDisplacement;
traction uniform (0 0 0);
pressure uniform 0;
value uniform (0 0 0);
}

uchyceni
{
type fixedValue;
value uniform (0 0 0);
}
}
```



```
namahani
{
    type      tractionDisplacement;
traction uniform (0 0 18244);
pressure uniform 0;
value uniform (0 0 0);
}
}

// ***** //
```

Příloha 3: nastavení materiálových konstant pro ocel:

Soubor /\$HOME/OpenFOAM/\$USER-2.2.2/run/constant/mechanicalProperties

```
/*-----*- C++ -*-----*\
| =====|
| \\      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O peration  | Version: 2.1.0 |
| \\      / A nd        | Web:      www.OpenFOAM.org |
| \\      / M anipulation |
|-----*\
FoamFile
{
  version      2.0;
  format       ascii;
  classdictionary;
  location     "constant";
  object       mechanicalProperties;
}
// * * * * * //

rho
{
  type        uniform;
  value       7854;
}

nu
{
  type        uniform;
  value       0.3;
}

E
{
  type        uniform;
  value       2.1e+11;
}

planeStress   no;

// * * * * * //
```

Příloha 4: nastavení chodu řešiče.

Soubor /\$HOME/OpenFOAM/\$USER-2.2.2/run/system/controlDict

```
/*-----*- C++ -*-----*\
| ===== |
| \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / O p e r a t i o n | Version: 2.1.0 |
| \ \ / A n d | Web: www.OpenFOAM.org |
| \ \ / M a n i p u l a t i o n | |
| \ \ / | |
\*-----*/
FoamFile
{
  version      2.0;
  format       ascii;
  classdictionary;
  location     "system";
  object       controlDict;
}
// ***** //

Application solidDisplacementFoam;

Start FromstartTime; // latestTime; //startTime; //latestTime;

startTime      0;

stopAtend Time;

endTime        1000e3;

deltaT         1;

writeControl   timeStep;

writeInterval  20e3;

purgeWrite     0;

writeFormat    ascii;

writePrecision 6;

writeCompression on;

timeFormat     general;

timePrecision  6;

graphFormat    raw;

runTimeModifiable true;

// ***** //
```

Příloha 5: nastavení chodu řešiče.

soubor/\$HOME/OpenFOAM/\$USER-2.2.2/run/system/fwSolution

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / F ield | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O peration | Version: 2.1.0 |
| \\ / A nd | Web: www.OpenFOAM.org |
| \\ / M anipulation |
\*-----*/
FoamFile
{
  version 2.0;
  format ascii;
  classdictionary;
  location "system";
  object fvSolution;
}
// * * * * * //

solvers
{
  "(D|T)"
  {
    solver GAMG;
    tolerance 1e-11;
    relTol 0.9;
    smoother GaussSeidel;
    cacheAgglomeration true;
    nCellsInCoarsestLevel 20;
    agglomeratorfaceArea Pair;
    mergeLevels 1;
  }
}

stressAnalysis
{
  compact NormalStress yes;
  nCorrectors 1;
  D 1e-10;
}

// ***** //
```